Exploiting General-Purpose Background Knowledge for Automated Schema Matching

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
der Universität Mannheim

presented by

Jan Philipp Portisch

Mannheim, 2022

Dekan: Dr. Bernd Lübcke, Universität Mannheim
Referent: Professor Dr. Heiko Paulheim, Universität Mannheim
Korreferent: Professor Dr. Catia Pesquita, Universidade de Lisboa

Tag der mündlichen Prüfung: 25. August 2022

Eidesstattliche Versicherung gemäß § 7 Absatz 2 Buchstabe c) der Promotionsordnung der Universität Mannheim zur Erlangung des Doktorgrades der Naturwissenschaften:

i Bei der eingereichten Dissertation zum Thema “Exploiting General-Purpose Background Knowledge for Automated Schema Matching” handelt es sich um mein eigenständig erstelltes eigenes Werk.

ii Ich habe nur die angegebenen Quellen und Hilfsmittel benutzt und mich keiner unzulässigen Hilfe Dritter bedient. Insbesondere habe ich wörtliche Zitate aus anderen Werken als solche kenntlich gemacht.

iii Die Arbeit oder Teile davon habe ich wie bislang nicht an einer Hochschule des In- der Auslands als Bestandteil einer Prüfungs- oder Qualifikationsleistung vorgelegt.

iv Die Richtigkeit der vorstehenden Erklärung bestätige ich.

v Die Bedeutung der eidesstattlichen Versicherung und die strafrechtlichen Folgen einer unrichtigen oder unvollständigen eidesstattlichen Versicherung sind mir bekannt. Ich versichere an Eides statt, dass ich nach bestem Wissen die reine Wahrheit erklärt und nichts verschwiegen habe.

Abstract

The schema matching task is an integral part of the data integration process. It is usually the first step in integrating data. Schema matching is typically very complex and time-consuming. It is, therefore, to the largest part, carried out by humans. One reason for the low amount of automation is the fact that schemas are often defined with deep background knowledge that is not itself present within the schemas. Overcoming the problem of missing background knowledge is a core challenge in automating the data integration process.

In this dissertation, the task of matching semantic models, so-called ontologies, with the help of external background knowledge is investigated in-depth in Part I. Throughout this thesis, the focus lies on large, general-purpose resources since domain-specific resources are rarely available for most domains. Besides new knowledge resources, this thesis also explores new strategies to exploit such resources.

A technical base for the development and comparison of matching systems is presented in Part II. The framework introduced here allows for simple and modularized matcher development (with background knowledge sources) and for extensive evaluations of matching systems.

One of the largest structured sources for general-purpose background knowledge are knowledge graphs which have grown significantly in size in recent years. However, exploiting such graphs is not trivial. In Part III, knowledge graph embeddings are explored, analyzed, and compared. Multiple improvements to existing approaches are presented.

In Part IV, numerous concrete matching systems which exploit general-purpose background knowledge are presented. Furthermore, exploitation strategies and resources are analyzed and compared. This dissertation closes with a perspective on real-world applications.

Zusammenfassung

Schema Matching ist ein wesentlicher Bestandteil des Datenintegrationsprozesses. Es stellt typischerweise den ersten Schritt der Datenintegration dar. Schema Matching ist sehr komplex und zeitaufwändig. Es wird – zu großen Teilen – noch immer von Menschen ausgeführt. Ein Grund für den niedrigen Grad der Automation hierbei ist die Tatsache, dass Schemata sehr oft mit Kontextwissen modelliert werden, welches letztendlich jedoch nicht Teil des Schemas wird.

In der vorliegenden Dissertation wird das Matching semantischer Modelle, sogenannter *Ontologien*, unter Zuhilfenahme externen Kontextwissens grundlegend erforscht; dies geschieht in Teil I dieser Arbeit. Ein Fokus liegt hierbei auf großen, allgemein gefassten Wissensressourcen, da fachspezifische Ressourcen für die meisten Domänen nur selten verfügbar sind. Neben der Untersuchung neuer Wissensressourcen werden in dieser Dissertation auch Methoden betrachtet, um solche Ressourcen sinnvoll zu nutzen.

Eine technische Grundlage für die Entwicklung und den Vergleich von Matchingsystemen wird in Teil II vorgestellt. Das hier eingeführte Framework erlaubt einfaches, gegebenenfalls kontextwissenbasiertes, sowie modulbasiertes Entwickeln von Softwareartefakten. Ferner bietet das vorgestelle Framework umfassende Möglichkeiten der Evaluation von Matchingsystemen.

Eine der größten strukturierten Ressourcen für allgemein gefasste Wissensressourcen sind Wissensgraphen (sogenannte *knowledge graphs*), welche in den letzten Jahren wesentlich gewachsen sind. Nichtsdestotrotz ist die Nutzung solcher Wissensgraphen nicht trivial. Teil III dieser Arbeit untersucht, analysiert und vergleicht sogenannte *knowledge graph embeddings*. Mehrere Verbesserungen bereits existierender Verfahren werden vorgestellt.

In Teil IV werden zahlreiche konkrete Matchingsysteme, welche allgemein gefasste Wissensressourcen nutzen, vorgestellt. Zudem werden Nutzungsstrategien und Ressourcen analysiert und verglichen. Diese Dissertation wird mit einem Blick auf praxisorientierte Anwendungsfälle abgerundet.

Contents

List of Publications

Parts of the work presented in this thesis were previously published in international journals and proceedings of international conferences. These publications are listed below. In cases in which the first authorship is shared, a spade (♠) footnote denotes the first authors.

International Journals

- Portisch, Jan; Heist, Nicolas; Paulheim, Heiko. Knowledge Graph Embedding for Data Mining vs. Knowledge Graph Embedding for Link Prediction - Two Sides of the Same Coin?. In: Semantic Web Journal (SWJ). 13(3). Pp. 399–422. 2022.
- Portisch, Jan; Hladik, Michael; Paulheim, Heiko. Background Knowledge in Ontology Matching: A Survey. Semantic Web Journal (SWJ). 2022.
- Portisch, Jan; Paulheim, Heiko. RDF2vec Variants and DL Classes. Semantic Web Journal. 2022. [to be submitted]
- Breit, Anna; Waltersdorfer, Laura; Ekaputra, Fajar J.; Sabou, Marta; Ekelhart, Andreas; Iana, Andreea; Paulheim, Heiko; Portisch, Jan; Revenko, Artem; ten Teije, Annette; van Harmelen, Frank. Combining Machine Learning and Semantic Web: A Systematic Mapping Study. ACM Computing Surveys. 2022. [under review]

International Conferences

- Portisch, Jan; Hladik, Michael; Paulheim, Heiko. Background Knowledge in Schema Matching: Strategy vs. Data. In: Proceedings of the International Semantic Web Conference (ISWC 2021). 2021.
- Portisch, Jan; Hladik, Michael; Paulheim, Heiko. KGvec2go – Knowledge Graph Embeddings as a Service. In: Language Resources and Evaluation Conference (LREC). 2020.

- Portisch, Jan; Fallatah, Omaima; Neumaier, Sebastian; Jaradeh, Mohamad Yaser; Polleres, Axel. Challenges of Linking Organizational Information in Open Government Data to Knowledge Graphs. In: Proceedings of the 22nd International Conference on Knowledge Engineering and Knowledge Management (EKAW 2020). 2020.

- Hertling, Sven; Portisch, Jan; Paulheim, Heiko. MELT - Matching Evaluation Toolkit. In: Lecture Notes in Computer Science Semantic Systems - The Power of AI and Knowledge Graphs. 15th International Conference, SEMANTiCS 2019. Karlsruhe, Germany. September 9–12, 2019.

- Portisch, Jan; Paulheim, Heiko. The DLCC Node Classification Benchmark for Analyzing Knowledge Graph Embeddings. International Semantic Web Conference (ISWC 2022). 2022. [to appear]

- Biswas, Russa[♠]; Portisch, Jan[♠]; Alam, Mehwish; Sack, Harald; Paulheim, Heiko. Entity Type Prediction Leveraging Graph Walks and Entity Descriptions. International Semantic Web Conference (ISWC 2022). 2022. [to appear]

International Workshops (ex OAEI)

- Hertling, Sven[♠]; Portisch, Jan[♠]; Paulheim, Heiko. KERMIT - A Transformer-Based Approach for Knowledge Graph Matching. Deep Learning meets Ontologies and Natural Language Processing (DeepOntoNLP2022) in conjunction with the ESWC 2022. 2022. [to appear]

- Hertling, Sven[♠]; Portisch, Jan[♠]; Paulheim, Heiko. Matching with Transformers in MELT. In: Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), OM@ISWC 2021. 2022.

- Portisch, Jan; Hladik, Michael; Paulheim, Heiko. FinMatcher at FinSim-2: Hypernymy Detection in the Financial Services Domain using Knowledge Graphs. In: Workshop on Financial Technology on the Web (FinWeb) in conjunction with The Web Conference. 2021.

- Hertling, Sven[♠]; Portisch, Jan[♠]; Paulheim, Heiko. Supervised Ontology and Instance Matching with MELT. In: The Fifteenth International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020). 2020.

Posters, Demos, and Short Papers

- Portisch, Jan; Costa, Guilherme; Stefani, Karolin; Kreplin, Katharina; Hladik, Michael; Paulheim, Heiko. Ontology Matching Through Absolute Orientation of Embedding Spaces. In: The Semantic Web: ESWC 2022 Satellite Events. 2022. [to appear]

- Portisch, Jan; Paulheim, Heiko. Walk this Way! Entity Walks and Property Walks for RDF2vec. In: The Semantic Web: ESWC 2022 Satellite Events. 2022. [to appear]

- Portisch, Jan; Paulheim, Heiko. Putting RDF2vec in Order. In: Proceedings of the International Semantic Web Conference - Posters and Demos, ISWC 2021. 2021.

- Portisch, Jan; Hladik, Michael; Paulheim, Heiko. RDF2Vec Light - A Lightweight Approach for Knowledge Graph Embeddings. International Semantic Web Conference (ISWC) 2020, Posters and Demonstrations Track. 2020.

- Portisch, Jan[♠]; Hertling, Sven[♠]; Paulheim, Heiko. Visual Analysis of Ontology Matching Results with the MELT Dashboard. In: The Semantic Web: ESWC 2020 Satellite Events. 2020.

- Portisch, Jan Philipp. Towards Matching of Domain-Specific Schemas Using General-Purpose External Background Knowledge. In: The Semantic Web: ESWC 2020 Satellite Events. 2020.

- Portisch, Jan; Hladik, Michael; Paulheim, Heiko. Evaluating Ontology Matchers on Real-World Financial Services Data Models. SEMANTiCS 2019. Karlsruhe, Germany. 2019.

- Hofmann, Alexandra; Perchani, Samresh; Portisch, Jan; Hertling, Sven; Paulheim, Heiko. DBkWik: Towards Knowledge Graph Creation From Thousands of Wikis. In: Proceedings of the ISWC 2017 Posters & Demonstrations and Industry Tracks co-located with the 16th International Semantic Web Conference (ISWC 2017). Vienna, Austria. 2017.

OAEI Systems and Results Publications

- Pour, Mina Abd Nikooie; Algergawy, Alsayed; Amardeilh, Florence; Amini, Reihaneh; Fallatah, Omaima; Faria, Daniel; Fundulaki, Irini; Harrow, Ian;

Hertling, Sven; Hitzler, Pascal; Huschka, Martin; Ibanescu, Liliana; Jiménez-Ruiz, Ernesto; Karam, Naouel; Laadhar, Amir; Lambrix, Patrick; Li, Huanyu; Li, Ying; Michel, Franck; Nasr, Engy; Paulheim, Heiko; Pesquita, Catia; Portisch, Jan; Roussey, Catherine; Tzania, Saveta; Splendiani, Andrea; Trojahn, Cássia; Vatascinová, Jana; Yaman, Beyza; Zamazal, Ondřej; Zhou, Lu. Results of the Ontology Alignment Evaluation Initiative 2021. In: Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), OM-@ISWC 2021. 2022.

- Portisch, Jan; Paulheim, Heiko. ALOD2Vec Matcher Results for OAEI 2021. In: Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), OM@ISWC 2021. 2022.

- Portisch, Jan; Paulheim, Heiko. Wiktionary Matcher Results for OAEI 2021. In: Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), OM@ISWC 2021. Virtual Space. 2022.

- Knorr, Leon; Portisch, Jan. Fine-TOM Matcher Results for OAEI 2021. In: Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), OM@ISWC 2021. 2022.

- Kossack, Daniel; Borg, Niklas; Knorr, Leon; Portisch, Jan. TOM Matcher Results for OAEI 2021. In: Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), OM@ISWC 2021. 2022.

- Portisch, Jan; Hladik, Michael; Paulheim, Heiko. ALOD2Vec Matcher Results for OAEI 2020. In: The Fifteenth International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020), OM@ISWC 2020. 2020.

- Portisch, Jan; Paulheim, Heiko. Wiktionary Matcher Results for OAEI 2020. In: The Fifteenth International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020), OM-@ISWC 2020. Virtual Space. 2020.

- Portisch, Jan; Hladik, Michael; Paulheim, Heiko. Wiktionary Matcher. CEUR Workshop Proceedings OM 2019 - Proceedings of the 14th International

Workshop on Ontology Matching co-located with the 18th International Semantic Web Conference (ISWC 2019), OM@ISWC 2019. Auckland, New Zealand. 2019.

- Monych, Michael; Portisch, Jan; Hladik, Michael; Paulheim, Heiko. DESK-Matcher. In: The Fifteenth International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020), OM@ISWC 2020. 2020.

- Portisch, Jan; Paulheim, Heiko. ALOD2Vec Matcher. In: CEUR Workshop Proceedings OM 2018 - Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference (ISWC 2018). Monterey, CA, USA. 2018.

Reports

- Portisch, Jan; Emonet, Vincent; Jaradeh, Mohamad Yaser; Fallatah, Omaima; Koteich, Bilal; Espinoza-Arias, Paola; Polleres, Axel. Tracking the Evolution of Public Datasets and Their Governance Bodies by Linking Open Data. In: Knowledge Graphs Evolution and Preservation - A Technical Report from ISWS 2019. 2020.

Theses

- Portisch, Jan Philipp. Automatic Schema Matching Utilizing Hypernymy Relations Extracted From the Web. 2018.

List of Patents

The author of this thesis has been employed at SAP SE during the creation of this work. Under §5 of the German *Arbeitnehmererfindungsgesetz* (ArbnErfG), German employees are legally obliged to notify their employer of inventions made within the scope of their employment. During the time of the employment at SAP SE, the author of this thesis reported multiple inventions to his employer. The company instructed law firms to file the inventions listed below as patents. It is important to note the subsequent enumerations are not complete since not all applications are yet publicly available due to the 18 months rule of the *United States Patent and Trademark Office* (USPTO). As of June 2022, six patents have been granted and an additional eight applications have been published.

Granted Patents
The year refers to the year of the patent grant.

- Bracholdt, Sandra; Saggau, Volker; Portisch, Jan. Schema Alignment and Structural Data Mapping of Database Objects. Patent Number: US 11,263,187 B2. 2022.

- Groß, Joachim; Portisch, Jan. Automatic Source Code Refactoring. Patent Number: US 11,288,062 B2. 2022.

- Portisch, Jan; Gross, Joachim; Bracholdt, Sandra. Logical, recursive definition of data transformations. Patent Number: US 11,106,861 B2. 2021.

- Portisch, Jan; Bracholdt, Sandra; Saggau, Volker. Dynamic generation of join statements for database operations. Patent Number: US 11,144,549 B2. 2021

- Portisch, Jan; Bracholdt, Sandra; Saggau, Volker. Dynamic Automatic Generation of Database Views. Patent Number: US 10,901,987 B2. 2021.

- Bracholdt, Sandra; Saggau, Volker; Portisch, Jan. Path Generation and Selection Tool for Database Objects. Patent Number: US 10,726,052 B2. 2020.

Patent Applications
The year refers to the year of the publication of the application.

- Portisch, Jan; Bracholdt, Sandra; Hladik, Michael; Huth, Oliver. Systems and Methods for Intelligent Labeling of Instance Data Clusters Based on Knowledge Graph. Publication Number: US 2022-0101151 A1. Application Number: 17/033,357. 2022.
- Portisch, Jan; Saggua, Volker; Bracholdt, Sandra. Generation of Realistic Mock Data. Publication Number: US 2022-0004532 A1. Application Number: 16/920,260. 2022.
- Portisch, Jan; Pantel, Bjoern; Merkle, Benjamin; Schub, Stephan; Bracholdt, Sandra; Kolata, Roland; Hladik, Michael; Meinel, Christoph. Automatic Conversion of Data Models Using Data Model Annotations. Publication Number: US 2021-0240675 A1. Application Number 16/780,481. 2021.
- Portisch, Jan; Boehle, Roland; Bracholdt, Sandra; Saggau, Volker. Transformation Rule Generation and Validation. Publication Number: US 2021-0232591 A1. Application Number: 16/776,407. 2021.
- Portisch, Jan; Boehle, Ronald; Saggau, Volker; Bracholdt, Sandra. Rule Mining for Rule and Logic Statement Development. Publication Number: US 2021-0073655 A1. Application Number: 16/567,470. 2021.
- Schub, Stephan; Portisch, Jan; Monych, Michael. Semantic, Single-Column Identifiers for Data Entries. Publication Number: US 2021-0073196 A1. Application Number: 16/564,365. 2021.
- Bracholdt, Sandra; Gross, Joachim; Saggau, Volker; Portisch, Jan. Matching Metastructure for Data Modeling. Publication Number: US 2020-0349130 A1. Application Number: 16/399,533. 2020.
- Portisch, Jan; Bracholdt, Sandra; Pantel, Björn; Saggau, Volker. Clustering Within Database Data Models. Publication Number: US 2020-0349128 A1. Application Number: 16/399,363. 2020.

List of Algorithms

Listings

List of Figures

List of Tables

Acronyms

ALCOMO Applying Logical Constraints on Matching Ontologies.

AML AgreementMakerLight.

ANN artificial neural network.

API application programming interface.

ArbnErfG Arbeitnehmererfindungsgesetz.

AUC Area under the ROC Curve.

BERT Bidirectional Encoder Representations from Transformers.

BK external knowledge resource.

CBOW continuous bag-of-words.

CNN convolutional neural network.

CPU central processing unit.

CSV comma-separated values.

DAEOM Deep Attentional Embedded Ontology Matching.

DI data integration.

DL description logic.

DL deep learning.

DLCC Description Logic Class Constructors.

DOI digital object identifier.

DOID Human Disease Ontology.

DOLCE descriptive ontology for linguistic and cognitive engineering.

EC exclusion criteria.

EDM Enterprise Data Management.

EDOAL Expressive and Declarative Ontology Alignment Language.

ER diagram entity-relationship diagram.

ETL extract, transform, load.

FAIR Findable, Accessible, Interoperable, Reusable.

FIBO Financial Industry Business Ontology.

FMA Foundational Model of Anatomy.

FOAF friend of a friend.

GNN graph neural network.

GPU graphics processing unit.

GUI graphical user interface.

HDT Header, Dictionary, Triples.

HOBBIT Holistic Benchmarking of Big Linked Data.

HTML HyperText Markup Language.

HTTP Hypertext Transfer Protocol.

IC inclusion criteria.

ICD International Classification of Diseases.

ID identifier.

IEEE Institute of Electrical and Electronics Engineers.

JSON JavaScript Object Notation.

JVM Java virtual machine.

KERMIT Knowledge Graph Matching with Transformers.

KG knowledge graph.

KGE knowledge graph embedding.

KNN k-nearest neighbors.

KPI key performance indicator.

LDA Latent Dirichlet Allocation.

LEMON Lexicon Model for Ontologies.

LOD Linked Open Data.

MAD mean absolute deviation.

MELT Matching EvaLuation Toolkit.

MeSH Medical Subject Headings.

MG Mapping Gain.

MIT Massachusetts Institute of Technology.

ML machine learning.

MLP multilayer perceptron network.

MRR mean reciprocal rank.

MWB Max Weight Bipartite Filtering.

NCBO National Center for Biomedical Ontology.

NCI National Cancer Institute.

NIST National Institute of Standards and Technology.

NLM United States National Library of Medicine.

NLP natural language processing.

NN neural network.

OA order-aware.

OAEI Ontology Alignment Evaluation Initiative.

OLA OWL-Lite Alignment.

OM ontology matching.

OPEC Organization of the Petroleum Exporting Countries.

OWL Web Ontology Language.

PBT population-based training.

PCA principal component analysis.

POS part of speech.

RAM random-access memory.

RDF Resource Description Framework.

RDFS RDF Schema.

REST representational state transfer.

RMSE root mean squared error.

RNN recurrent neural network.

RQ research question.

SBERT Sentence-BERT.

SEALS Semantic Evaluation At Large Scale.

SG skip-gram.

SKOS Simple Knowledge Organization System.

SMOTE synthetic minority oversampling technique.

SNOMED Systematized Nomenclature of Medicine.

SNOMED-CT Systematized Nomenclature of Medicine Clinical Terms.

SPARQL SPARQL Protocol and RDF Query Language.

SQL Structured Query Language.

SUMO suggested upper merge ontology.

SVM support vector machine.

SW Semantic Web.

SWRL Semantic Web Rule Language.

TF-IDF term frequency–inverse document frequency.

UBERON Uber-anatomy ontology.

UCITS Undertakings for Collective Investment in Transferable Securities.

UI user interface.

UMLS Unified Medical Language System.

URI Uniform Resource Identifier.

URL Uniform Resource Locator.

USPTO United States Patent and Trademark Office.

W3C World Wide Web Consortium.

WSD Word Sense Disambiguation.

XML eXtensible Markup Language.

YAAA Yet Another Alignment API.

Acknowledgment

Several people played a vital role throughout the years I worked on my dissertation. I am extraordinarily thankful to have received constant support in pursuing my research.

Heiko Paulheim encouraged me early on to follow my interests and to seek a doctoral degree. He agreed to supervise this dissertation, never stopped to provide valuable and creative input, and was always available to discuss detailed aspects in-depth. He was very encouraging in every phase of the dissertation and created a pleasant working atmosphere with a fine sense of humor. I met Catia Pesquita multiple times in the context of the Ontology Matching Workshop and I am very happy that she agreed to be the secondary supervisor of this dissertation.

During my dissertation, I always valued the plentiful conversations with other researchers at the University of Mannheim. In particular, long discussions with Sven Hertling helped to sharpen and to improve my research; during the time of this dissertation, he was always there to discuss complicated conceptual issues as well as challenging code parts and I enjoyed very much working with him on multiple projects. I further appreciated working together with Andreea Iana and Nicolas Heist on various occasions. Another influence and help in this dissertation was the whole Data and Web Science Group at the University of Mannheim: Although hard to pin down to individual people, I found the weekly colloquia of multiple chairs inspiring and highly appreciate the feedback I got for my presentations and posters.

Despite far too few in-person meetings due to the coronavirus pandemic, I valued the (digital) encounters with the research community very much. Due to my research focus, I particularly enjoyed the Ontology Matching Workshop. At the Ontology Alignment Evaluation Initiative, Ernesto Jiménez-Ruiz put a lot of faith in Sven and me by helping us to roll out the Matching EvaLuation Toolkit to the OAEI community. Rudi Studer was among the first external researchers to review my research plan and to give me feedback. I also enjoyed the short time of working together with Marta Sabou, Frank van Harmelen, and their teams on a very broad survey. After meeting Axel Polleres at a summer school, he helped Omaima Fallatah, Mohamad Yaser Jaradeh, and myself to publish our initial idea.

I also received a lot of backing at SAP. Michael Hladik helped me early on to focus on research and in shaping the topic; he agreed to supervise this dissertation from SAP's side, supported my research throughout the dissertation, and was always available to discuss open questions and new ideas. Sandra Bracholdt always stepped in in stressful situations and continues to amaze me with her ability to find simple solutions for complicated problems. Ralf Sabiwalsky, whom I, unfortunately, got to know only for a very short time, always took me seriously, recommended me to pursue a PhD, and was in many ways a role model. I always valued the openness at SAP Financial Services towards new ideas. Here, Michael Monych, Stephan Schub, Volker Saggau, Björn Pantel, Stefan Schonger, Christoph Meinel, and Roland Kolata were always available to discuss and challenge new ideas. I also received support from the SAP One Domain Model team – particularly Thomas Ziegert, Eckehard Schmidt, and Daniela Wünsch. I am particularly grateful to have had the chance to participate in building a software pilot at SAP Financial Services and SAP Domain Services. Of enormous help in realizing this prototype were Markus Bucher, Michael Monych, Guilherme Costa, and Radu Purdel.

In every situation, my best friend Sophia was available for a phone call and always cheered me up. My partner, Isabella, supported me in life, endured unusual working hours, and also proofread many publications. I am also grateful for my family who never stops believing in me.

– Thank you, Jan

Part I

Motivation and Foundation

Chapter 1

Introduction

The ubiquity of data influences the decision of individuals and businesses. Information systems empower a traveler to find the cheapest price of a flight and help a warehouse manager ensure that a warehouse is never short on goods. The performance of such systems is driven by their ability to process information. A core feature of information processing is the capacity to consume multiple, heterogeneous sources of data. Therefore, data needs to be integrated. Data integration describes the effort to allow for unified access across multiple autonomous and heterogeneous sources of data [107]. Data integration generally increases the value of an information system. The traveler in the introductory example, for instance, will obtain the highest value for his information intent if *all* flights of *all* airlines are considered. Data integration is, therefore, a vital process in improving existing information systems. For businesses in the 21^{st} century, processing data – ergo integrating data – is a competitive advantage. The process of integrating data can be divided into multiple steps [561, 301]: (1) Schema Matching, (2) Schema Translation, (3) Record Linkage, and (4) Data Fusion. This dissertation focuses on the first step of matching schemas, the task of finding semantically related elements in two schemas. The most important relation here is equivalence which is also the focus of this dissertation. Schema matching is mainly necessary because schemas are heterogeneous. Particularly semantic heterogeneity is a challenging factor.

The schema matching task is typically very complex and, to the largest part, carried out by humans. One reason for the low amount of automation is the fact that schemas are often defined with deep background knowledge that is not itself present within the schemas. Overcoming the problem of missing background knowledge is a core challenge in automating the data integration process.

While background knowledge is missing in the data integration process, there is an increasing supply of publicly available, large data sources such as open dictionaries, knowledge graphs, or deep learning models that contain latent background knowledge.

In addition to newly available background knowledge sources, new techniques for using knowledge graphs emerged and yielded promising results in other settings such as traditional machine learning problems. Combining knowledge graphs and embedding techniques for the task of schema matching is, therefore, particularly interesting.

This thesis explores and compares multiple general-purpose background knowledge sources and exploitation strategies for schema matching. Hereby, a focus is also put on novel latent exploitation techniques in the area of knowledge graph embeddings.

1.1 Research Questions

The main goal of this thesis is to answer the overarching question *How can general-purpose background knowledge be exploited in ontology matching?* In order to answer this question, several subordinate research questions have been enunciated. These sub-questions are listed below[1] together with the parts and chapters which address them:

- **RQ1** *How can matching systems be developed and evaluated in a re-usable way?* Developing matching systems is not easy. In order to develop, analyze, and compare matching systems, a comprehensive framework is required. This question is addressed mainly in Part II. Additionally, throughout this thesis, individual matching components are highlighted and integrated into an overall framework.

- **RQ2** *How can very large background knowledge sources be exploited as background knowledge?* Large knowledge sources are challenging in many ways. Cases in point are knowledge graph embedding approaches: Most embedding approaches do not scale to very large graphs such as Wikidata. Furthermore, knowledge graph embedding vectors can easily require multiple gigabytes of free disk storage together with further hardware requirements to process the vectors. In many instances, this is im-

[1] The provided enumeration contains still a high level of abstraction. Individual chapters may further refine these questions and cover sub-aspects. In the enumeration, each *research question* (RQ) is assigned to a number to form an identifier; this identifier is used consistently throughout this dissertation.

practical; a case in point is submitting a matching system to an ontology evaluation campaign. This research question is addressed in Part III Chapters 8 and 9 of this thesis.

- **RQ3** *What are knowledge graph embeddings really learning and how can this be influenced?* Embeddings are a powerful way of exploiting knowledge graphs. However, in many instances, it is not clear what is actually learned by the approaches. This dissertation analyzes, compares, and presents new approaches to knowledge graph embeddings. This research question is addressed throughout Part III of this dissertation.

- **RQ4** *How do changes in the background knowledge source and the exploitation strategy affect automated matching?* When using general-purpose external background knowledge for the matching operation, multiple resources are available to choose from. In addition, numerous strategies exist to exploit a knowledge source. Individual matching approaches are presented and evaluated throughout Part IV. In Chapter 18, a systematic evaluation of strategies and resources is presented to explore this question.

- **RQ5** *What are applications of background-knowledge-based matching systems?* Applications are found throughout this dissertation but are explicitly addressed in Part IV where multiple individual matching systems are presented and evaluated; moreover, multiple real-world applications are discussed.

1.2 Contributions

This thesis contains numerous diverse contributions in the area of ontology matching and knowledge graph matching, knowledge graph embeddings, and the exploitation of general-purpose background sources in schema matching. More specifically, the following contributions are made:

- **Review of Background Knowledge in Ontology Matching** The first contribution of this thesis is an in-depth analysis of background knowledge usage in ontology matching. This contribution is addressed in Chapter 3.

- **Modern Open-Source Matching Framework** The second contribution is a mature matching framework for matcher development, evaluation, fine-tuning, and packaging. It is used for all matcher development and evaluation tasks carried out in this dissertation. Therefore, it goes beyond simple evaluation capabilities but contains state of the art matching components. The matching framework has significant third-party usage in the ontology matching community. The contribution is primarily addressed in Part II of this dissertation.

This contribution is joint work with Sven Hertling.

- **Systematic Comparison of Knowledge Graph Embedding Approaches** Another contribution is the systematic comparison of knowledge graph embedding approaches for data mining and knowledge graph embedding approaches for link prediction. The contribution is primarily addressed in Part III Chapter 7.

  This contribution is joint work with Nicolas Heist.

- **Embedding Accessibility** Knowledge graph embeddings are computationally expensive to calculate and also to consume. This dissertation contributes to improved accessibility for knowledge graph embeddings. This aspect is covered in Chapters 8 and 9.

- **Improvement to Existing Knowledge Graph Embedding Algorithms** In this dissertation, the RDF2vec algorithm is extended and evaluated, leading to significant performance improvements on many tasks. This contribution is mainly addressed in Chapters 10 and 11.

- **Provisioning of a new Embedding Gold Standard** Most if not all embedding approaches have in common that it is not definitely clear *what* is learned. In this dissertation, a novel gold standard based on *description logic* (DL) is presented in Chapter 12 to analyze embeddings in depth.

- **Systematic Review of RDF2vec Approaches** RDF2vec (and other knowledge graph embedding approaches) come in many flavors. This dissertation analyzes them in Chapter 13 and derives recommendations for various tasks.

- **Matching Systems Exploiting Background Knowledge** This dissertation contributes multiple novel matching systems exploiting various background knowledge resources using explicit and latent strategies. This contribution is primarily discussed in Chapters 15, 16, and 17.

- **Extensive Analysis of Datasets and Strategies** Another contribution is the extensive analysis of datasets, exploitation strategies, and their interrelations made in Chapter 18.

- **Presentation of Practical Applications** Lastly, Chapter 19 presents the application of matching systems in real-world applications.

In addition to the contributions mentioned above, multiple software and dataset contributions were made. Except for parts in Chapter 19, all implementations are publicly available. The most notable contributions with 3^{rd} party usage are the *Matching EvaLuation Toolkit* (MELT)[2] and jRDF2vec[3]. A compilation can be found in Table 1.1.

1.3 Thesis Outline

The structure of this dissertation is visualized in Figure 1.1. It is divided into five parts (dark gray boxes in the figure). Each part consists of one or more chapters (white boxes with a chapter indicator in square brackets in the figure). White boxes without a chapter indicator symbolize topic areas within a chapter in Figure 1.1. Some chapters focus specifically on a matching system which exploits general-purpose background knowledge. In the figure, those chapters are additionally annotated with respect to the exploitation method being (A) latent or (B) explicit.[4] In the following, each part and chapter are shortly summarized.

Part I: Motivation and Foundation This part introduces the reader to the topic. It is comprised of the following chapters:

> **Chapter 1: Introduction** This chapter motivates the dissertation at hand and introduces the underlying research questions. It further summarizes the contributions of this PhD project and provides an overview of the structure of this dissertation.

> **Chapter 2: Fundamentals** This chapter introduces basic concepts of data integration, semantics, the Semantic Web, ontology matching, and knowledge graphs.

[2] `https://github.com/dwslab/melt/`

[3] `https://github.com/dwslab/jRDF2Vec`

[4] We limit this annotation to chapters dedicated to concrete matching systems for improved clarity. Latent and explicit matching methods are also explored in other chapters, such as Chapter 6, where a latent method is also evaluated but not the core contribution.

Name	Description	Chapters	URL
Absolute Orientation	Implementation and evaluation concerning rotations of embedding spaces for ontology matching.	14	[a]
ALOD2vec Matcher	The implementation of the ALOD2vec matching system.	17	[b]
DL-Evaluation-Framework	A framework to evaluate knowledge graph embeddings on description logics test cases.	12	[c]
DL-TC-Generator	A generation framework for a gold standard for knowledge graph embeddings. The repository contains also a dataset.	12, 13	[d]
jRDF2vec	A high-performance implementation for RDF2vec embeddings and their derivations.	9, 10	[e]
KBC Evaluation	An evaluation framework to evaluate knowledge base completion predictions.	7	[f]
KBC Predictions	A gensim extension to automatically produce knowledge base completion predictions.	7	[g]
KGE Models	All knowledge graph embedding models are made publicly available via KGvec2go.	8	[h]
KGvec2go	Server code for KGvec2go and its API.	8	[i]
MELT	The Matching EvaLuation Toolkit is a large knowledge graph matching and evaluation framework.	4, 5,6, 16	[j]
ODP GS	A gold standard which links open data publishers to Wikidata and DBpedia URIs. The gold standard can be used to evaluate and/or train entity linking systems. For more information, see [398].	–	[k]
Wiktionary Matcher	The implementation of the Wiktionary Matcher ontology matching system.	15	[l]

[a] `https://github.com/guilhermesfc/ontology-matching-absolute-orientation`
[b] `https://github.com/janothan/ALOD2VecMatcher`
[c] `https://github.com/janothan/dl-evaluation-framework`
[d] `https://github.com/janothan/DL-TC-Generator`
[e] `https://github.com/dwslab/jRDF2Vec`
[f] `https://github.com/janothan/kbc_evaluation/`
[g] `https://github.com/janothan/kbc_rdf2vec`
[h] `http://kgvec2go.org/download.html`
[i] `https://github.com/janothan/kgvec2go-server`
[j] `https://github.com/dwslab/melt/`
[k] `https://github.com/YaserJaradeh/LinkingODPublishers/blob/master/GoldStandard.csv`
[l] `https://github.com/janothan/WiktionaryMatcher`

Table 1.1: Software and Dataset Contributions Made in this Dissertation. Excluded (but yet publicly available) are evaluation scripts and implementations that do not provide any value besides experiment reproducibility.

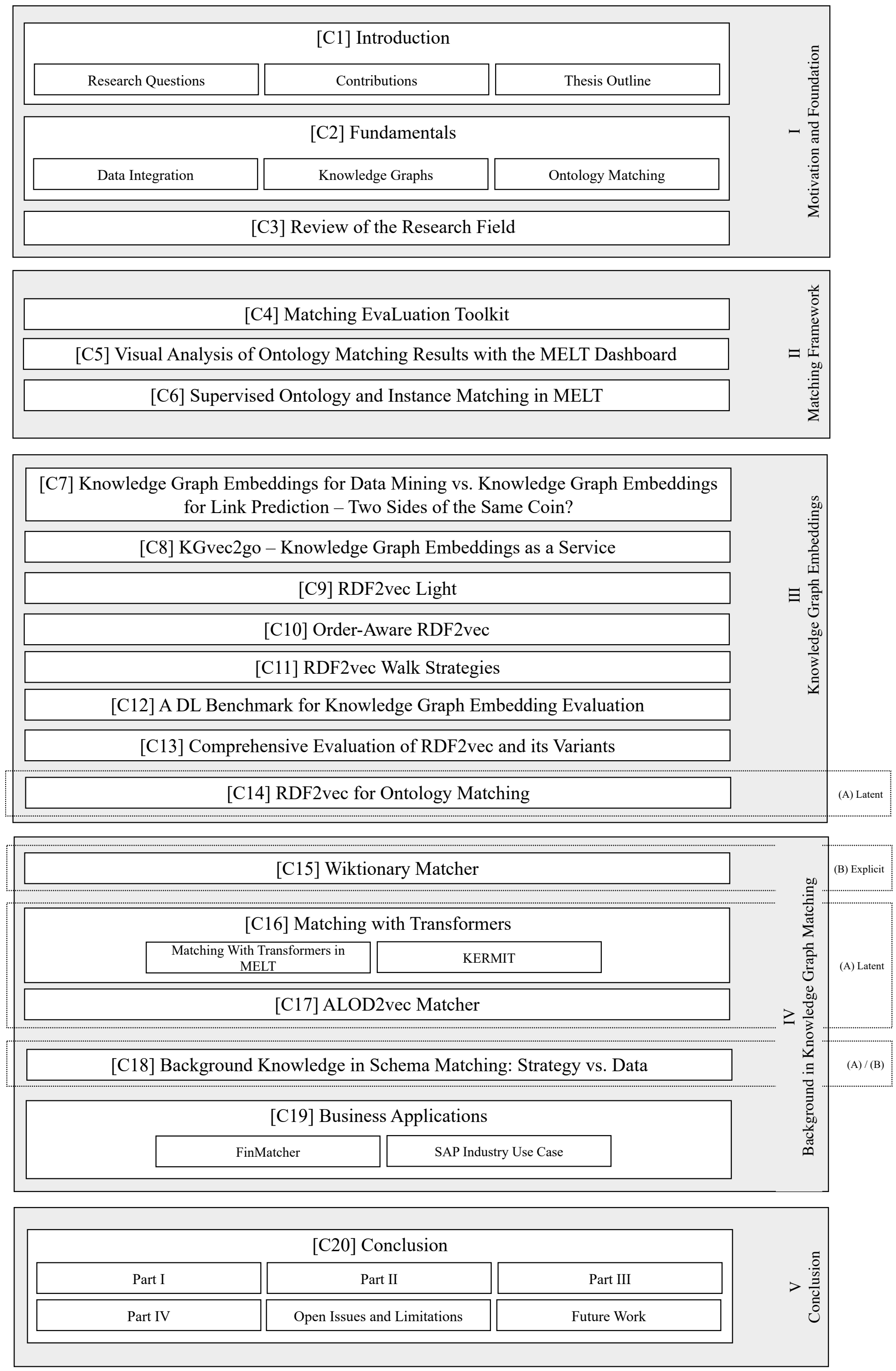


Figure 1.1: Structure of this Dissertation

Chapter 3: Review of the Research Field After the core concepts are introduced, this chapter provides an in-depth analysis of the usage of background knowledge for the task of ontology alignment. More precisely, the background knowledge sources and the approaches applied to use external knowledge are reviewed in-depth using a systematic literature review methodology.

Part II: Matching Framework In this part, the underlying framework for matcher development and evaluation, named *Matching EvaLuation Toolkit* (MELT), is presented. MELT is a software framework to facilitate ontology matcher development, configuration, evaluation, and packaging. It was developed in the course of the dissertation, particularly to expedite matcher evaluation and reuse. Part II is comprised of the following chapters:

Chapter 4: Matching EvaLuation Toolkit Overview This chapter provides an overview of the MELT framework. The core architectural concepts are presented together with an exemplary analysis of two *Ontology Alignment Evaluation Initiative* (OAEI) tracks.

Chapter 5: Visual Analysis of Ontology Matching Results with the MELT Dashboard After the core framework was introduced in the previous chapter, an interactive dashboard extension of MELT is introduced. The dashboard allows for interactive self-service analyses such as a drill down into the matcher performance for data type properties or into the performance of matchers within a certain confidence threshold. In addition, the dashboard offers detailed group evaluation capabilities that allow for the application in broad evaluation campaigns. The MELT Dashboard is actively used in the research community.

Chapter 6: Supervised Ontology and Instance Matching in MELT In this chapter, a machine learning extension to the Matching EvaLuation Toolkit is presented, which facilitates the application of supervised learning for ontology and instance matching. The extension is used to evaluate two supervised machine learning matchers: (1) A latent, RDF2vec-based matching approach and (2) a multi-feature approach for knowledge graphs.

Part III: Knowledge Graph Embeddings Knowledge graph embeddings are projections of entities and relations to lower-dimensional spaces. These representations are useful for a broad range of tasks. In this part, multiple contributions to the field of knowledge graph embeddings are presented.

Chapter 7: Knowledge Graph Embedding for Data Mining vs. Knowledge Graph Embedding for Link Prediction – Two Sides of the Same Coin? In this chapter, the reader is introduced to the topic of knowledge graph embeddings. They have been proposed mainly for two purposes: (1) providing an encoding for data mining tasks, and (2) predicting links in a knowledge graph. Both lines of research have been pursued rather in isolation from each other with their own benchmarks and evaluation methodologies. In this chapter, it is evaluated in how far both tasks are actually related. It is shown in two sets of experiments that both approaches can be used for both tasks. The differences in the similarity functions evoked by the different embedding approaches are discussed.

Chapter 8: KGvec2go – Knowledge Graph Embeddings as a Service In this chapter, KGvec2go is presented. KGvec2go is a Web API for accessing and consuming graph embeddings in a lightweight fashion in downstream applications. Pre-trained embeddings for four knowledge graphs are provided. The service and its usage are introduced, and it is further shown that the trained models have semantic value by evaluating them on multiple semantic benchmarks. The evaluation also reveals that the combination of multiple models can lead to a better outcome than the best individual model.

Chapter 9: RDF2vec Light In this chapter, a new, lightweight, RDF2vec-based approach for knowledge graph embeddings is presented. It is evaluated on three machine learning and retrieval tasks, and the performance is compared with the classic RDF2vec approach. It is shown that the new approach requires only a fraction of the computing power compared to the original approach while maintaining a similar performance. Moreover, it is shown that RDF2vec Light does not lose performance when reducing the dimensionality of the vector space.

Chapter 10: Order-Aware RDF2vec In this chapter, a small but very effective adaption of the classic RDF2vec algorithm is proposed and evaluated: While the classic approach cannot distinguish the position of the elements in the randomly generated walks, the adaption presented in this chapter can. Both approaches are evaluated and compared.

Chapter 11: RDF2vec Walk Strategies In this chapter, we introduce two new flavors of walk extraction coined e-walks and p-walks, which put an emphasis on the structure or the neighborhood of an entity respectively and thereby allow for creating embeddings that focus on similarity or relatedness.

Chapter 12: A DL Benchmark for Knowledge Graph Embedding Evaluation Most knowledge graph embedding approaches are evaluated on a single task or a single group of tasks to determine their overall performance. The evaluation is then assessed in terms of how well the embedding approach performs on the task at hand, but it is hardly evaluated (and often not even deeply understood) what information the embedding approaches are *actually* learning to represent. The chapter at hand presents a new gold standard, named *Description Logic Class Constructors* (DLCC). In addition, a first evaluation is presented.

Chapter 13: Comprehensive Evaluation of RDF2vec and its Variants In earlier chapters, multiple extensions to RDF2vec are introduced; of particular interest here are ordered RDF2vec (Chapter 10) and RDF2vec walk strategies (Chapter 11). In addition, a description logic-based gold standard is introduced in Chapter 12. An interesting perspective is, hence, an extensive evaluation of the RDF2vec variants presented using, among others, the newly presented gold standard. This chapter provides an in-depth evaluation of 12 RDF2vec variants together with seven benchmark models. Hypotheses based on logic constructors are developed, verified, and discussed.

Chapter 14: RDF2vec for Ontology Matching After having introduced and analyzed knowledge graph embeddings together with the presentation of multiple novel extensions, Part III closes with a presentation of two exploitation options for RDF2vec for the task of ontology matching: A structural and a background-knowledge-based approach. A matching system following the structural approach is presented and evaluated. A system which uses embedding-based methods on background knowledge is presented in the subsequent part of this dissertation.

Part IV: Background Knowledge in Knowledge Graph Matching In this part, multiple external-knowledge-based matchers are presented and evaluated. Different background knowledge sources, as well as the exploitation strategies, are explored. A comparison of multiple sources and strategies is provided together with an impact analysis. This part closes with a look at real-world applications.

Chapter 15: Wiktionary Matcher In this chapter, a knowledge-based matching system is presented, which uses a large, community-built dictionary as a resource for matching. An explicit exploitation strategy is applied. The system participated in the OAEI multiple times and was continuously updated and improved.

Chapter 16: Matching with Transformers With the rise of transformer-based language models, text comparison based on meaning (rather than lexical features) is possible. In this chapter, we model the ontology matching task as a classification problem and present approaches based on transformer models. We provide an easy-to-use implementation in the MELT framework, which is suited for ontology and knowledge graph matching. We show that a transformer-based filter helps to choose the correct correspondences given a high-recall alignment and already achieves a good result with simple alignment post-processing methods. As a second contribution, we present *Knowledge Graph Matching with Transformers* (KERMIT), a matching tool that combines bi- and cross-encoders. We show that bi-encoders are suitable for blocking and that – despite the supervised matching setting – a reference sample is not necessarily required.

Chapter 17: ALOD2vec Matcher In this chapter, an external-knowledge-based matching system is presented, which uses a very large, automatically built knowledge graph. The general-purpose graph is embedded using RDF2vec, and the embeddings are subsequently used within the matching operation. The system participated in the OAEI multiple times and was continuously updated and improved.

Chapter 18: Background Knowledge in Schema Matching: Strategy vs. Data In this chapter, six general-purpose knowledge graphs are exploited as sources of background knowledge for the matching task. The background sources are evaluated by applying three different exploitation strategies. We find that explicit strategies still outperform latent ones and that the choice of the strategy has a greater impact on the final alignment than the actual background dataset on which the strategy is applied. While we could not identify a universally superior resource, BabelNet achieved consistently good results. The best matcher configuration with BabelNet performs very competitively when compared to other matching systems even though no dataset-specific optimizations were made.

Chapter 19: Business Applications In this chapter, two concrete, exemplary business applications are presented which exploit techniques presented in this dissertation: (1) A financial matching system for financial instruments and (2) a prototype for business schema matching developed at SAP SE.

Part V: Outlook and Conclusion This chapter summarizes the previous parts of this thesis. The contributions are outlined, and open issues are addressed together with future work.

Parts of this dissertation have already been published (see *List of Publications* on page xiii). Tables 1.2 and 1.3 show the relation of individual chapters of this dissertation to published works. Table 1.4 lists published work without a dedicated chapter in this dissertation.

At the beginning of each chapter or section, bold print indicates whether the complete chapter/section or parts of the chapter/section have been published before together with the full reference of the publication. Even in the case of a completely published chapter, changes may have been applied which are not explicitly highlighted. Examples for such changes may be but are not limited to: Fixes of spelling errors, unification of writings[5], stylistic optimizations to figures and tables, or additional paragraphs/footnotes for further clarification.

[5] An example for such a unification is the writing of “RDF2vec”. Some publications refer to the approach as “RDF2Vec”. More recently, the former variant can be observed more often and is also used on `http://rdf2vec.org/`. In order to ensure a consistent reading experience, the former writing variant is used consistently throughout this dissertation, even if the other variant was originally used in published works.

Chapter	Reference	Publication
2	[417]	Portisch, Jan Philipp. Automatic Schema Matching Utilizing Hypernymy Relations Extracted From the Web. 2018.
3	[408]	Portisch, Jan; Hladik, Michael; Paulheim, Heiko. Background Knowledge in Ontology Matching: A Survey. Semantic Web Journal (SWJ). 2022.
4	[203]	Hertling, Sven; Portisch, Jan; Paulheim, Heiko. MELT - Matching Evaluation Toolkit. In: Lecture Notes in Computer Science Semantic Systems - The Power of AI and Knowledge Graphs. 15th International Conference, SEMANTiCS 2019. Karlsruhe, Germany. September 9–12, 2019.
5	[400]	Portisch, Jan♠; Hertling, Sven♠; Paulheim, Heiko. Visual Analysis of Ontology Matching Results with the MELT Dashboard. In: The Semantic Web: ESWC 2020 Satellite Events. 2020.
6	[204]	Hertling, Sven♠; Portisch, Jan♠; Paulheim, Heiko. Supervised Ontology and Instance Matching with MELT. In: The Fifteenth International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020). 2020.
7	[399]	Portisch, Jan; Heist, Nicolas; Paulheim, Heiko. Knowledge Graph Embedding for Data Mining vs. Knowledge Graph Embedding for Link Prediction - Two Sides of the Same Coin?. In: Semantic Web Journal (SWJ). 13(3). Pp. 399–422. 2022.
8	[404]	Portisch, Jan; Hladik, Michael; Paulheim, Heiko. KGvec2go – Knowledge Graph Embeddings as a Service. In: Language Resources and Evaluation Conference (LREC). 2020.
9	[405]	Portisch, Jan; Hladik, Michael; Paulheim, Heiko. RDF2Vec Light - A Lightweight Approach for Knowledge Graph Embeddings. International Semantic Web Conference (ISWC) 2020, Posters and Demonstrations Track. 2020.
10	[412]	Portisch, Jan; Paulheim, Heiko. Putting RDF2vec in Order. In: Proceedings of the International Semantic Web Conference - Posters and Demos, ISWC 2021. 2021.
11	[416]	Portisch, Jan; Paulheim, Heiko. Walk this Way! Entity Walks and Property Walks for RDF2vec. In: The Semantic Web: ESWC 2022 Satellite Events. 2022. [to appear]
12	[414]	Portisch, Jan; Paulheim, Heiko. The DLCC Node Classification Benchmark for Analyzing Knowledge Graph Embeddings. International Semantic Web Conference (ISWC 2022). 2022. [to appear]
13	[415]	Portisch, Jan; Paulheim, Heiko. RDF2vec Variants and DL Classes. Semantic Web Journal. 2022. [to be submitted]

Table 1.2: Assignment of Publications to Chapters (Part 1 of 2)

Chapter	Reference	Publication
14	[397]	Portisch, Jan; Costa, Guilherme; Stefani, Karolin; Kreplin, Katharina; Hladik, Michael; Paulheim, Heiko. Ontology Matching Through Absolute Orientation of Embedding Spaces. In: The Semantic Web: ESWC 2022 Satellite Events. 2022. [to appear]
15	[402]	Portisch, Jan; Hladik, Michael; Paulheim, Heiko. Wiktionary Matcher. CEUR Workshop Proceedings OM 2019 - Proceedings of the 14th International Workshop on Ontology Matching co-located with the 18th International Semantic Web Conference (ISWC 2019), OM-@ISWC 2019. Auckland, New Zealand. 2019.
	[410]	Portisch, Jan; Paulheim, Heiko. Wiktionary Matcher Results for OAEI 2020. In: The Fifteenth International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020), OM@ISWC 2020. Virtual Space. 2020.
	[413]	Portisch, Jan; Paulheim, Heiko. Wiktionary Matcher Results for OAEI 2021. In: Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), OM@ISWC 2021. Virtual Space. 2022.
16	[205]	Hertling, Sven[♠]; Portisch, Jan[♠]; Paulheim, Heiko. Matching with Transformers in MELT. In: Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), OM@ISWC 2021. 2022.
	[206]	Hertling, Sven[♠]; Portisch, Jan[♠]; Paulheim, Heiko. KERMIT - A Transformer-Based Approach for Knowledge Graph Matching. Deep Learning meets Ontologies and Natural Language Processing (DeepOntoNLP2022) in conjunction with the ESWC 2022. 2022. [to appear]
18	[406]	Portisch, Jan; Hladik, Michael; Paulheim, Heiko. Background Knowledge in Schema Matching: Strategy vs. Data. In: Proceedings of the International Semantic Web Conference, ISWC 2021. 2021.
19	[407]	Portisch, Jan; Hladik, Michael; Paulheim, Heiko. FinMatcher at FinSim-2: Hypernymy Detection in the Financial Services Domain using Knowledge Graphs. In: Workshop on Financial Technology on the Web (FinWeb) in conjunction with The Web Conference. 2021.

Table 1.3: Assignment of Publications to Chapters (Part 2 of 2)

Reference	Publication
[216]	Hofmann, Alexandra; Perchani, Samresh; Portisch, Jan; Hertling, Sven; Paulheim, Heiko. DBkWik: Towards Knowledge Graph Creation From Thousands of Wikis. In: Proceedings of the ISWC 2017 Posters & Demonstrations and Industry Tracks co-located with the 16th International Semantic Web Conference (ISWC 2017). Vienna, Austria. 2017.
[401]	Portisch, Jan; Hladik, Michael; Paulheim, Heiko. Evaluating Ontology Matchers on Real-World Financial Services Data Models. SEMANTiCS 2019. Karlsruhe, Germany. 2019.
[1]	Portisch, Jan; Emonet, Vincent; Jaradeh, Mohamad Yaser; Fallatah, Omaima; Koteich, Bilal; Espinoza-Arias, Paola; Polleres, Axel. Tracking the Evolution of Public Datasets and Their Governance Bodies by Linking Open Data. In: Knowledge Graphs Evolution and Preservation - A Technical Report from ISWS 2019. 2020.
[396]	Portisch, Jan Philipp. Towards Matching of Domain-Specific Schemas Using General-Purpose External Background Knowledge. In: The Semantic Web: ESWC 2020 Satellite Events. 2020.
[351]	Monych, Michael; Portisch, Jan; Hladik, Michael; Paulheim, Heiko. DESK-Matcher. In: The Fifteenth International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020), OM-@ISWC 2020. 2020.
[398]	Portisch, Jan; Fallatah, Omaima; Neumaier, Sebastian; Jaradeh, Mohamad Yaser; Polleres, Axel. Challenges of Linking Organizational Information in Open Government Data to Knowledge Graphs. In: Proceedings of the 22nd International Conference on Knowledge Engineering and Knowledge Management (EKAW 2020). 2020.
[419]	Pour, Mina Abd Nikooie; Algergawy, Alsayed; Amardeilh, Florence; Amini, Reihaneh; Fallatah, Omaima; Faria, Daniel; Fundulaki, Irini; Harrow, Ian; Hertling, Sven; Hitzler, Pascal; Huschka, Martin; Ibanescu, Liliana; Jiménez-Ruiz, Ernesto; Karam, Naouel; Laadhar, Amir; Lambrix, Patrick; Li, Huanyu; Li, Ying; Michel, Franck; Nasr, Engy; Paulheim, Heiko; Pesquita, Catia; Portisch, Jan; Roussey, Catherine; Tzania, Saveta; Splendiani, Andrea; Trojahn, Cássia; Vataščinová, Jana; Yaman, Beyza; Zamazal, Ondřej; Zhou, Lu. Results of the Ontology Alignment Evaluation Initiative 2021. In: Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), OM@ISWC 2021. 2022.
[281]	Kossack, Daniel; Borg, Niklas; Knorr, Leon; Portisch, Jan. TOM Matcher Results for OAEI 2021. In: Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), OM@ISWC 2021. 2022.
[278]	Knorr, Leon; Portisch, Jan. Fine-TOM Matcher Results for OAEI 2021. In: Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), OM@ISWC 2021. 2022.
[39]	Biswas, Russa[♠]; Portisch, Jan[♠]; Alam, Mehwish; Sack, Harald; Paulheim, Heiko. Entity Type Prediction Leveraging Graph Walks and Entity Descriptions. International Semantic Web Conference (ISWC 2022). 2022. [to appear]
[47]	Breit, Anna; Waltersdorfer, Laura; Ekaputra, Fajar J.; Sabou, Marta; Ekelhart, Andreas; Iana, Andreea; Paulheim, Heiko; Portisch, Jan; Revenko, Artem; ten Teije, Annette; van Harmelen, Frank. Combining Machine Learning and Semantic Web: A Systematic Mapping Study. ACM Computing Surveys. 2022. [under review]

Table 1.4: Publications Without a Dedicated Chapter

Chapter 2

Fundamentals

This chapter introduces the basics of data integration, semantics and semantic relations, the Semantic Web, and the ontology matching problem.

Parts of the work presented in this chapter have been published before as: Portisch, Jan Philipp. Automatic Schema Matching Utilizing Hypernymy Relations Extracted From the Web. 2018. [417]

2.1 Data Integration

Data integration (DI) describes the effort to allow for unified access across multiple autonomous and heterogeneous sources of data [107]. Data integration is not restricted to a technology stack (such as relational or graph databases) but, instead, comprises all technological means to store and access data. Data integration can be understood as a multistep process. It can be divided in four main parts [561] as depicted in Figure 2.1: (i) *Schema Matching*, (ii) *Schema Translation*, (iii) *Record Linkage*, and (iv) *Data Fusion*.

Schema Matching Schema matching is an important and time-consuming part of the data integration process. Out of the actions to carry out in order to integrate two given schemas (depicted in Figure 2.1), schema matching is the first

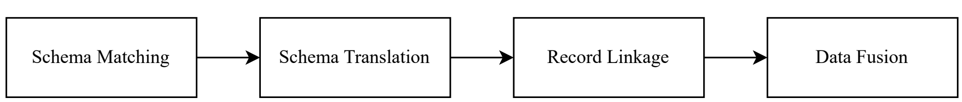


Figure 2.1: Process for integrating two schemas, compiled from [561].

step. Schema matching describes the process of finding the relations that hold between the elements of the schemas that are to be matched. The most important relation here is the equivalence relation. In this step, structural as well as semantic heterogeneity between the two schemas are bridged.

Schema Translation Schema translation describes the process of deriving the translation function from one schema to the other schema.

Record Linkage Record linkage describes the process of linking the records of instances of two schemas, i.e. finding equivalent records in disparate datasets.

Data Fusion Data fusion describes the process of resolving conflicting information concerning individual instances.

In this dissertation, the focus is on the first step: schema matching. More precisely, the task of ontology matching (see Section 2.6) is addressed.

2.2 Semantics

In this section, a general introduction to semantics is given and aspects relevant for this thesis are explained: First, the difference between *syntax* and *semantics* is pointed out. Afterward, important relations between concepts in the semantic space are introduced.

2.2.1 General Concepts

Syntax Syntax, from Latin *syntaxis* derived from Greek σύν ("with") and τάξις ("placing"), refers – on a general level – to a set of rules that define how to structure characters and strings [573, 342, 209]. In linguistics, it refers to the analysis of the arrangements of words, phrases, and clauses together with their grammatical relations [49].

Semantics Semantics is derived from ancient Greek ςεμαντικός ("significant") and refers to the "the study of meaning" [435, 341, 572].[1] The meaning of a word can also be referred to as *concept* [331]. As the field of semantics is too broad

[1]The meaning of *meaning*, i.e., the question of what meaning actually is, is itself an interesting research area which is – due to the focus of this thesis – not covered at this point. For details, one can refer to Riemer who dedicates a full 40 pages long chapter of his textbook *Introducing Semantics* to this topic [436].

to be presented in the scope of this thesis, the focus in the following lies on a subset, i.e., semantic relations among concepts.[2]

2.2.2 Semantic Relations

Every linguistic sign (i.e., word or lexeme[3]) itself is a relation between the signifier (also *sound-image*, French: *signifiant*) and the signified (the concept, French: *signifié*) [458], as depicted in Figure 2.2.

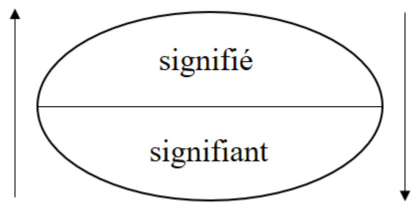


Figure 2.2: Two Sides of a Linguistic Sign According to Saussure [458]

Besides the relation between signifiant and signifié, there are also relations between signs: *syntagmatic relations* and *paradigmatic relations* (also *associative relations*).

Syntagmatic Relations

Syntagmatic relations are those between signs in a chain of signs; in the English language, for instance, it is grammatically correct to say “he sleeps at night” but not “he sleep at night” because the verb and the subject have to agree in person. [62, 91]

Paradigmatic Relations

Paradigmatic relations are associations of concepts that exist in the mind of humans but are not necessarily existent in the chain of signs. When reading “to sleep”, for instance, there is an implicit association with “sleeping”, “bed”, “night”, and so on. [459, 460, 62]
Busch and Stenschke count more than ten possible paradigmatic relations (see

[2] A concise introduction to semantics for non-linguists can be found in Busse’s book *Semantik* [61].

[3] A lexeme is “a unit of lexical meaning, which exists regardless of any inflectional endings[...]” [89]. It is also known as “*lexical item*” [90]. The “headwords in a dictionary are [...] lexemes” [89].

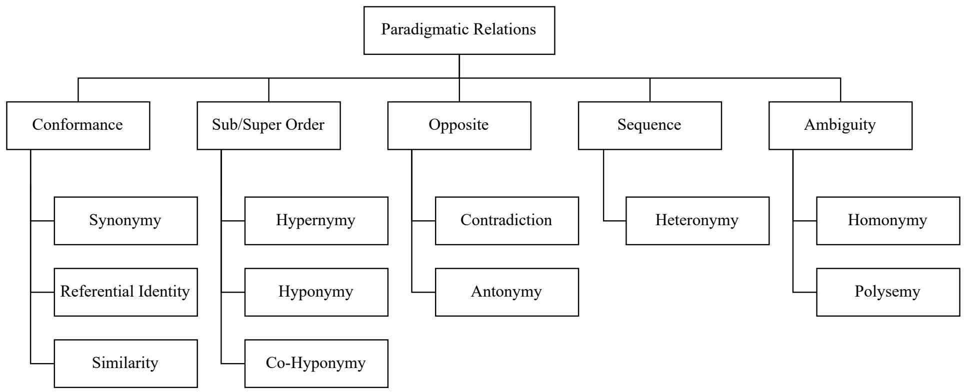


Figure 2.3: Paradigmatic Relations According to Busch and Stenschke [60]

Figure 2.3 for a complete overview) [60]. In the following, only the paradigmatic relations relevant for this thesis are further explained: Hypernymy and hyponymy, monosemy and polysemy, synonymy and antonymy, homonymy as well as similarity and relatedness.

2.3 Paradigmatic Relations

Hypernymy and Hyponymy A hypernym (also hyperonym) is a concept that is superordinate to other concepts, i.e., it defines a category to which other concepts belong. Those subordinate concepts are called hyponyms. [60, 62] The concept of a *financial contract*, for instance, subsumes the concept of a *loan*; therefore, the *financial contract* is a hypernym of *loan* whereas the latter one is a hyponym of the first one.

Monosemy and Polysemy Polysemy describes the property of a lexeme to carry more than one meaning [62]. The concept of *apple*, for example, can refer to (i) the fruit, (ii) the tree, or (iii) the Californian technology company; the concept is, therefore, polysemous. A monosemous lexeme, in contrast, carries only one meaning.

Synonymy and Antonymy Synonymy describes the property of two words to be used interchangeably. Within this definition, there are various forms which mainly focus on whether synonyms have to share one sense, i.e., are interchangeable in one particular context or whether they have to share all senses, i.e., are interchangeable in (almost) all contexts. A strong-form definition of synonymy

requires the two words to be interchangeable in any situation. Strong-form synonymous words are seldom. [62] An example of weak-form synonymy would be *student* and *pupil* regarding the sense of *somebody being taught by a teacher* but not regarding the sense *being the center of the eye* [437]. The words *doorknob* and *doorhandle*, on the other hand, have only one and the same meaning and could be used as an example of strong-form synonymy [420].[4]
Antonyms, on the other hand, are incompatible with each other like *hot* and *cold* [521, 60]. If antonyms divide a domain into exactly two parts and are logically incompatible at the same time, like *dead* and *alive*, they are considered to be a *contradiction* [62].

Homonymy Words with the same writing and pronunciation but different meanings are called *homonyms* [395]. An example of a homonym would be *bear* which – depending on the context – can refer to the animal (*Winnie-the-Pooh is a bear.*) or to the verb (*I cannot bear it any longer.*).

Similarity and Relatedness Similarity describes how far two concepts are similar to each other "by virtue of their similarity" [55]. Similarity and relatedness are often not clearly separated from each other (for instance in [154]). Nevertheless, there are significant differences. Dissimilar entities can even be semantically related by antonymy relationships [55]. Hill et al. distinguish the two relations by giving examples: While the concepts *coffee* and *cup* are certainly related, they are not similar; however, a mug and a cup can – in language as in the real world – almost be used interchangeably and are, therefore, similar [208].

2.4 The Semantic Web

In this section, a general introduction to the Semantic Web is given. First, general concepts of the Semantic Web are introduced. Then, linked data is explained and, lastly, the dataset used in this thesis is presented.

2.4.1 General Concepts

Semantic Web While information is broadly available on the Web and consumable by humans, computers cannot consume this information due to data

[4]Note that when having a very close look, there are still subtle differences; even though they carry the same sense, *doorhandle*, for example, is more common in Great Britain whereas *doorknob* is mostly used in the United States [420]. This goes even as far as some linguists believing "that there is no such thing as true synonymy" [395].

heterogeneity and lack of implicit knowledge. One solution would be to have an artificial intelligence that actually can interpret all the information as it is. However, up to now, there is no such artificial entity that can reliably accomplish this task. The idea of the *Semantic Web*, on the other hand, is to give information right away in a format that can be interpreted by machines and to provide the required toolset to do so.[5] The Semantic Web provides standards to ensure interoperability and to allow reasoning according to logic. [209] The Semantic Web technology is sometimes also referred to as *Web 3.0* [193].[6]

Semantic Web Language Stack In Figure 2.4, the Semantic Web language stack is depicted. The technical foundations are Unicode and *Uniform Resource Identifiers* (URIs). Together, they allow to uniquely identify concepts on the Web in the desired language. The *eXtensible Markup Language* (XML) is a language that allows exchanging structured data in a machine- and human-readable way [551]. The *Resource Description Framework* (RDF) allows to express simple statements on the Web [556]; it is further explained in the following paragraph.

RDF Schema (RDFS) and the *Web Ontology Language* (OWL) are used to give meaning to the vocabulary used in RDF statements. Rules can additionally be used to express semantics on a deeper level. OWL and RDFS are explained later in this section in more detail. The *SPARQL Protocol and RDF Query Language* (SPARQL) allows to query RDF data [553].

By combining RDF data and the corresponding semantics, logical inference is possible. This process is referred to as *reasoning*. [115] Because "anybody can say anything about anything" (*AAA Principle*) [16], there might be multiple views on the truth. Thus, it is valuable to evaluate the credibility of sources and to build trust.

In this thesis, the focus is on the middle layer of the stack, mainly RDF, SPARQL, and OWL. Therefore, selected concepts are explained in more detail in the following.[7]

Resource Description Framework (RDF) To represent information about resources in a structrued form, the *World Wide Web Consortium* (W3C) developed

[5] Although information can be provided directly so that it is consumable by computers, it is also possible that extractors derive structured information from websites. An example of such a process would be the implementation of *DBpedia* [300] or *DBkWik* [216].

[6] Unfortunately, the term *Web 3.0* is often used for marketing purposes due to the success of the term *Web 2.0*. Therefore, there is no real definition for *Web 3.0*, and it is used to refer to different things ranging from virtual worlds [361] to decentralized services such as cryptocurrencies [604].

[7] There is more to the Semantic Web than the content described in this section. A comprehensive introduction is given in Hitzler et al.'s textbook *Semantic Web* [210].

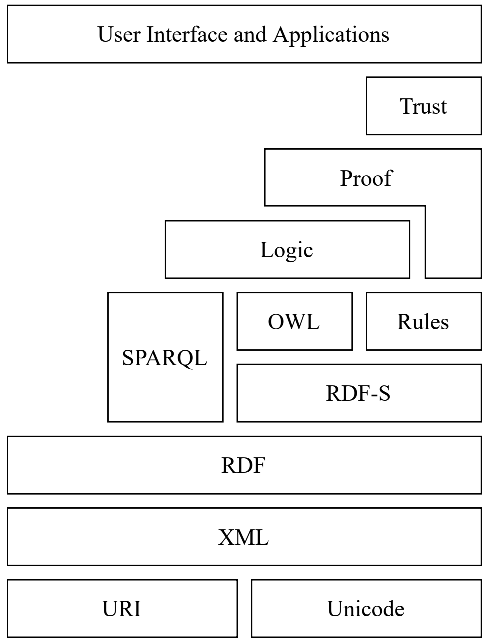


Figure 2.4: Semantic Web Language Stack According to Tim Berners-Lee [36] (adapted)

the *Resource Description Framework* (RDF). The data model behind this standard is relatively simple; statements are given in triples: `<subject> <predicate> <object>`. Resources are uniquely identified by URIs. When regarding subjects and objects as nodes and predicates as edges, multiple triples can form a connected graph. This structure allows to interlink knowledge on the Web. [211]

In some cases, it is necessary to model more complex relations that would require *helper nodes*. An example would be a network of friends where it shall be expressed when people met for the first time. In such cases, blank nodes are used. They are addressed by using a node ID but cannot be addressed by a URI (which would be semantically questionable). An example is given in Figure 2.5. [211]

For RDF serialization, different formats are available such as Turtle [557] or JSON-LD [554]. There are also formats to serialize multiple graphs in one file, such as N-Quads [555].

Ontologies *Ontology*, from Latin *ontologia* derived from Greek οντος ('being') and λογος ('study of'), is originally a part of philosophy that focuses on the ques-

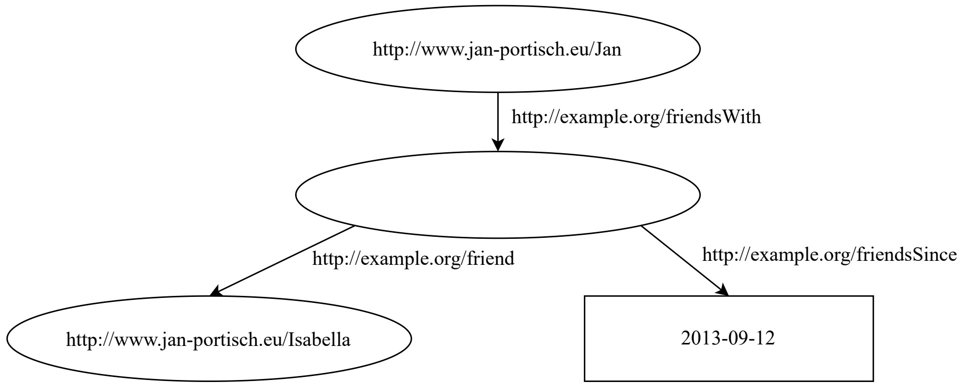


Figure 2.5: RDF Blank Node Example

tion of *being*, i.e., the nature of the world [58].[8] In philosophy, the terms *ontology* and *metaphysics* are often used interchangeably [58].
In information technology, the term *ontology* is used to refer to a specific formalization of concepts: Gruber defines a conceptualization as an “abstract, simplified view of the world” and an ontology as “an explicit specification of a conceptualization” [174]. In the context of the Semantic Web, an ontology models a domain and defines a vocabulary to be used by an application [124]. Two important concepts of ontologies are classes and properties: Classes define the type of a resource, whereas properties are the predicates of a statement. Classes and properties can be hierarchically structured, i.e., it is also possible to define subclasses as well as sub-properties. [211]
An example of an ontology would be the *friend of a friend* (FOAF) ontology[9] which can be used to describe social networks, for instance [48]. Ontologies are also already used directly in the business world, for example, in the form of industry-specific ontologies such as the *Financial Industry Business Ontology* (FIBO)[10] by the *Enterprise Data Management* (EDM) Council. Oberle et al. [375] also describe the usage of concrete enterprise applications.
An ontology consisting of “meta, generic, abstract and philosophical” [508] concepts is also referred to as *upper ontology* by the IEEE[11] Upper Ontology Working

[8] Bunge and Mahner [59] give an excellent (and understandable) introduction into the philosophical dimension of *ontology*.
[9] see `http://www.foaf-project.org/`
[10] see `https://www.edmcouncil.org/financialbusiness`
[11] The *Institute of Electrical and Electronics Engineers* (IEEE) is a large association of technical professionals. For more details, see `https://www.ieee.org/`

Group[12]; however, the term is likewise used to refer to ontologies with general concepts (for instance in [334]). An example for an ontology according to a strict definition would be *DOLCE* [161] whereas *SUMO* [372] or *OpenCyc* [302] are illustrations for more general-purpose upper ontologies containing also domain-specific concepts [334].[13]

Ontology Languages There are multiple languages and ways to represent ontologies.[14] A lightweight format to do so is *RDF Schema* (RDFS, RDF-S). [211] A more expressive and powerful format is the *Web Ontology Language* (OWL) which is structured in three sublanguages listed here in descending expressibility: OWL Full, OWL DL, and OWL Lite. OWL Lite is a subset of OWL DL, and OWL DL is a subset of OWL Full. [212] OWL is recommended by the W3C [552] and also the language of choice to represent ontologies within the scope of this thesis.

SPARQL Similar to the *Structured Query Language* (SQL) for databases, the *SPARQL Protocol and RDF Query Language* (SPARQL) allows to query RDF data. Queries are formulated as patterns that are matched against a *knowledge graph* (KG). In addition, more complex structures such as filters, aggregations, or optional patterns are also available. [213] An example of a simple query is given in Listing 2.1. Originally designed as a pure query language, version 1.1 offers functions to update data [553].

---

[12] The Upper Ontology Working Group has resolved by now. Nevertheless, their definitions are still available using web.archive.org, see `http://web.archive.org/web/20140512225349/http://suo.ieee.org/`.

[13] *DOLCE* is an acronym for "descriptive ontology for linguistic and cognitive engineering"; *SUMO* is an acronym for "suggested upper merge ontology" and *OpenCyc* is derived from *Open Encyclopedia*. All three ontologies are rather addressed using their abbreviated forms.

[14] Staab and Studer dedicate more than 100 pages to this topic in their *Handbook on Ontologies* [507].

```
PREFIX : <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?population WHERE {
        :Mannheim dbo:populationMetro ?population .
}
```

Listing 2.1: SPARQL sample query that will return the population of Mannheim. The query can be run on the public DBpedia endpoint.[15]

2.4.2 Linked Data

Tim Berners-Lee defined four principles for linked data which are given word-by-word in the following enumeration [35]:

1. Use `URI`s as names for things[.]
2. Use `HTTP URI`s so that people can look up those names.
3. When someone looks up a `URI`, provide useful information, using the standards (RDF*, SPARQL)[.]
4. Include links to other `URI`s so that they can discover things.

He further defines *Linked Open Data* (LOD) in 2010 as "Linked Data which is released under an open license, which does not impede its reuse for free" [35].

2.5 Knowledge Graph Embeddings

A knowledge graph $\mathcal{G}$ is a labeled directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{R} \times \mathcal{V}$ for a set of relations $\mathcal{R}$. Vertices are subsequently also referred to as *entities* and edges as *predicates*. Such a graph is also referred to as *directed heterogeneous graph* in opposition to homogeneous graphs where all nodes and edges belong to a single type [63, 584].

A *knowledge graph embedding* (KGE) is a projection Π for all vertices $v \in \mathcal{V}$ and optionally $r \in \mathcal{R}$ into a multi-dimensional space of dimension Δ. Hence

[15] see `https://dbpedia.org/snorql/?query=PREFIX+%3A+%3Chttp%3A%2F%2Fdbpedia.org%2Fresource%2F%3E%0D%0APREFIX+dbo%3A+%3Chttp%3A%2F%2Fdbpedia.org%2Fontology%2F%3E%0D%0ASELECT+%3Fpopulation+WHERE+%7B%0D%0A%09%3AMannheim+dbo%3ApopulationMetro+%3Fpopulation+.%0D%0A%7D`

$\Pi = \{e_i \in \mathbb{R}^{\Delta}\}$ where $i = 1, 2, ...|\mathcal{V}|$ or $i = 1, 2, ...|\mathcal{V}| + |\mathcal{R}|$.[16] Numerous approaches for knowledge graph embeddings were presented in the past, and multiple surveys on knowledge graph embeddings were published [559, 63, 93, 584]. A detailed introduction and presentation of related work with regards to knowledge graph embeddings can also be found in Chapter 7 of this dissertation. Different knowledge graph embeddings are compared with each other in Chapters 7 and 13.

In this introduction, we will follow the classification system established by Cai et al. [63] which is depicted in Figure 2.6. The authors introduce two taxonomies: (1) A taxonomy based on the inputs and outputs of an embedding approach (depicted on the left in Figure 2.6) and (2) a taxonomy based on the embedding techniques (depicted on the right in Figure 2.6).

Graph Embedding Problem Settings The four embedding input types are (1) homogeneous and (2) heterogeneous graphs, (3) graphs with auxiliary information (namely labels, attributes, etc.), and (4) constructed graphs (e.g., from images where pixels are interpreted as nodes). The four embedding output types are (1) node embeddings, (2) edge embeddings, (3) hybrid embeddings, and (4) whole-graph embeddings, where the output is a single vector representing the complete graph.

Since for KGEs, the input and output types are rather static (directed heterogeneous graphs or graphs with auxiliary information, respectively, as input and mainly node embeddings as output), we will focus on the second taxonomy introduced in the subsequent paragraph.

Graph Embedding Techniques and Well-Known Embedding Approaches Matrix factorization embedding approaches transform the graphs into tensors and apply a factorization method. A well-known matrix factorization approach based on node proximity matrix factorization is RESCAL [370]. The approach models a graph as a three-way tensor and subsequently applies tensor decomposition. DistMult [538] is a scalability improvement over RESCAL at the cost that relationships are assumed to be symmetric. ComplEx [538] extends DistMult by using complex vector spaces rather than real ones.[17]

Some approaches apply *deep learning* (DL) for embedding graphs. These approaches use either the complete graph as input (without random walk) or apply sampling for element proximity (with random walk). RDF2vec [442] (and

[16] In this dissertation, the focus lies on deterministic point vector embedding approaches. The notation assumes a real vector space; this is not the case for ComplEx [538] and RotatE [516].

[17] Hence, for ComplEx: $\Pi = \{e_i \in \mathbb{C}^{\Delta}\}$ where $i = 1, 2, ...|\mathcal{V}| + |\mathcal{R}|$

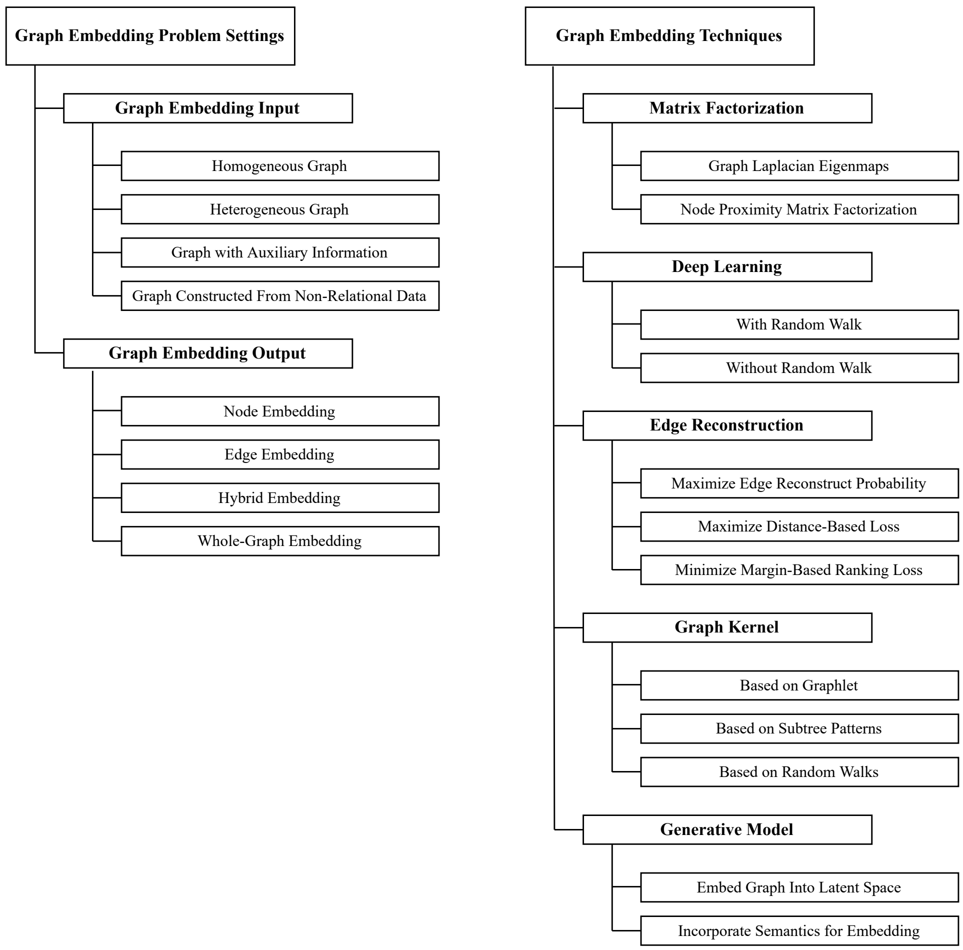


Figure 2.6: Graph Embedding Taxonomies According to Cai et al. [63]

all its variants [412, 416]) fall into the latter category: Multiple walks are performed within a graph, typically for each node, and the set of walks is then interpreted as sentences by the word2vec language embedding algorithm [345, 344]. Conceptually, RDF2vec is similar to DeepWalk [391] with the difference that the latter approach was presented in the context of homogeneous graphs, i.e., graphs with merely one edge type.

Edge reconstruction methods follow the notion that edges in the embedding space should be as similar as possible to the edges in the input graph. TransE [44] is a well-known edge-reconstruction approach which minimizes the margin-based ranking loss. Given a triple in the form *(**h**ead, **l**abel, **t**ail)*, TransE trains embeddings $\mathbf{h}$, $\mathbf{l}$, $\mathbf{t}$, such that $\mathbf{h} + \mathbf{l} \approx \mathbf{t}$. As an extension, TransR [317] learns two embedding spaces, one for entities and one for relations, so that it better captures compositional rules and non-one-to-one cardinalities of relationships. RotatE [516] regards relations as rotations of vertices in complex space.[18]

Since graph kernels are designed for embedding a whole graph, this category is not relevant for this dissertation. An example for generative models would be the *Latent Dirichlet Allocation* (LDA) applied on graphs. Embedding approaches from this category, however, are not commonly used for knowledge graph embedding applications and are not further discussed in this chapter.

2.6 The Ontology Matching Problem

This section covers the very core problem of this thesis: Ontology matching. First, general concepts are introduced. Afterward, different levels of ontology heterogeneity are analyzed. In order to link to the world of data integration, schema matching and how it relates to ontology matching is explained subsequently. Thereafter, different techniques for ontology matching are presented, and it is shown where the matcher of this thesis fits in. Lastly, areas of ontology evaluation and challenges within the process are covered.

2.6.1 General Concepts

Ontology The concept of ontologies has been introduced in Subsection 2.4.1. In the following, ontologies refer to their meaning in the context of the Semantic Web.

Correspondence A correspondence is a relation that holds between entities e_1 and e_2, which are from different ontologies. An entity can be a class or a property

[18] Hence, for RotatE: $\Pi = \{e_i \in \mathbb{C}^{\Delta}\}$ where $i = 1, 2, \ldots |\mathcal{V}| + |\mathcal{R}|$

of an ontology. [124] In its minimal form, a correspondence is a triple of the form (e_1, e_2, r) where r is the relation that holds between the entities. The relation is a set-theoretic one like *equivalence (=)*, *disjointness* ($\perp$), or *less general* ($\leq$). Additionally, a matcher might assign an *identifier* (ID) and a confidence value to a triple. [124] In this thesis, the focus is on correspondences with equivalence relations.
Correspondences can be of different complexity: Given two ontologies (prefixed *onto1* and *onto2*), a correspondence consists in its simplest form of the triple notation explained above, for instance: *(onto1:Author, onto2:Writer, =)*. Such simple relations can be insufficient and not expressive enough as there might be additional conditions such as restrictions or conversions. An example for three complex correspondences is given in Figure 2.7 together with their translation in first-order logic in Listing 2.2.

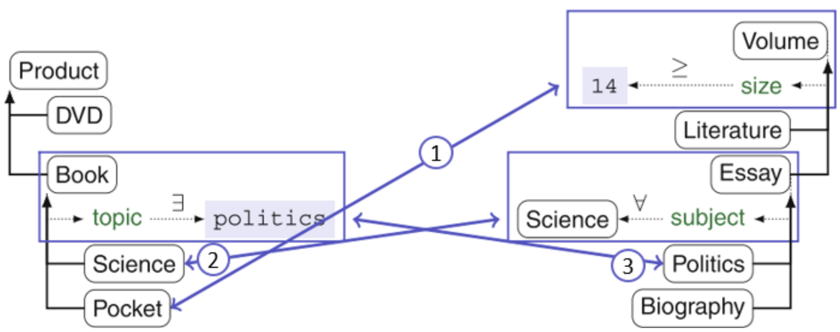


Figure 2.7: Complex Correspondences Example. This figure is taken from [125] and adapted. First-order logic translations for the numbered correspondences can be found in Listing 2.2.

```
∀x, Pocket(X) ≡ Volume(x) ∧ size(x,y) ∧ y ≤ 14
∀x, Science(x) ≡ Essay(x) ∧ (∀y, subject(x,y) ⇒ Science(y))
∀x, Book(x) ∧ topic(x,politics) ≡ Politics(x)
```

Listing 2.2: Complex Correspondences in First Order Logic
The translations are given for the example in Figure 2.7.

Those complex mappings require an elaborate format. Examples for such a format would be the *Semantic Web Rule Language* (SWRL) [220] or the *Expressive and Declarative Ontology Alignment Language* (EDOAL) which was originally known as *SEKT Mapping Language* [52] and *OMWG Ontology Mapping Language* [123]. [125] This thesis concentrates on non-complex correspondences.

Alignment The set of correspondences between ontologies is called *alignment*. An alignment is not restricted to a one-to-one (1:1) cardinality but can instead be of different cardinalities: One-to-one (1:1), one-to-many (1:m), many-to-one (m:1), or many-to-many (n:m) [480]. Those are explained in more detail in the next paragraph. The goal of ontology alignment is, ultimately, to automatically obtain correct alignments between any given ontologies [124].

Matching Restrictions The matching process can be subject to restrictions. There are multiple possible *arity restrictions* when ontology A is matched to ontology B:

1. One-to-One (1:1)
   This restriction specifies that one element $e_1 \in A$ is matched to zero or one element $e_2 \in B$. Each element $e_2 \in B$ is matched to zero or one element $e_1 \in A$. When there are multiple options for correspondences, and each correspondence has a confidence score, this problem is equivalent to the *maximum weighted bipartite graph matching problem* in mathematics.

2. One-to-Many (1:m) / Many-to-One (m:1)
   This restriction specifies that one element $e_1 \in A$ is matched to zero or more elements $e_j \in B$. Each element $e_j \in B$ can, therefore, be matched to zero or more elements $e_1 \in A$.

3. Many-to-Many (n:m)
   This restriction specifies that each element $e_i \in A$ is matched to zero or more elements $e_j \in B$.

From an implementation viewpoint, there is not one exclusive option but multiple ways in implementing arity restrictions. [82]
Another alignment restriction is concerned with what can be matched: In a *homogeneous alignment*, only resources of the same type are matched; for example, ontology classes can only be matched to other classes but not to data or object properties. In *heterogenous alignments*, on the other hand, any resource type can be matched to any other resource type. [167]

Ontology Matching The goal of the *ontology matching process* is to obtain an alignment A for a pair of ontologies o_1 and o_2. This process is also known as *ontology alignment* or *ontology mediation* [51]. This is achieved through a $matcher$

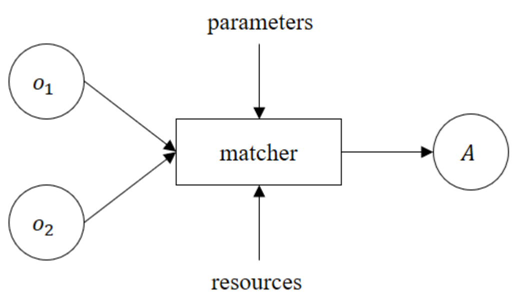


Figure 2.8: Matching Process According to Euzenat and Shvaiko [124].

which may use resources r (such as thesauri[19] or common knowledge) and which can be configured by setting parameters p (such as weights or thresholds). The matcher can be viewed as a function $f(o_1, o_2, p, r) = A$. This *matching process* is also depicted in Figure 2.8.[20] [124, 479]
It is also possible to apply a filter operation to the matcher output: Matchers can assign a confidence value to each correspondence which is usually in the $[0, 1]$ range. A threshold $t \in [0, 1]$ can then be defined to only add correspondences with a confidence $\geq t$ to the final alignment. [6]
Concerning the methodology of ontology matching, no distinct superior methodology has emerged over the years – not even in the older field of data model schema matching [126].

2.6.2 Ontology Heterogeneity

Differences in ontologies require a reconciliation process if interoperability is a desired property. The differences can occur at several levels. Important distinctions are differences in the structure (syntax) and differences in the semantics. This observation is older than the Semantic Web itself and has already been made in the area of multidatabase systems [476, 273].
Several classification systems exist to bring these observations into the broader context of ontologies, for example, by Klein [277] or Hameed et al. [180]. Euzenat and Shvaiko consolidate different views on heterogeneity into four main

[19] A thesaurus groups lexemes by meaning. As opposed to a dictionary, where the user tries to find the meaning or use of a lexeme, a thesaurus is used to find lexemes for a certain meaning. [91] A well-known English thesaurus is *WordNet* [346]; an example for a German thesaurus would be *GermaNet* [181, 194].

[20] Euzenat and Shvaiko include in their formal definition also an input alignment A'. As techniques utilizing A' are not discussed in this thesis, a slightly simplified version is presented here.

types following Bouquet et al. [45][21]. Figure 2.9 displays a general overview of the different types of heterogeneity.

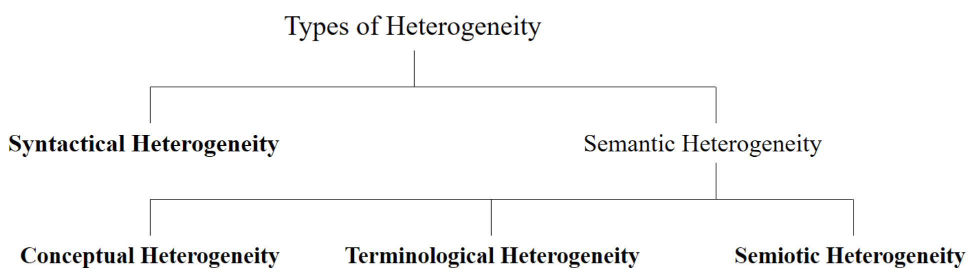


Figure 2.9: Ontology Heterogeneity. The grouping according to syntax and semantics is taken from Klein [277], boldly printed types are from Euzenat and Shvaiko [124]

Syntactic Heterogeneity Syntactic heterogeneity is used to refer to the difference in formalization of ontologies, i.e., when different ontology languages are used. In such cases, a transformation is required if interoperability is desired. [124]

Terminological Heterogeneity Terminological heterogeneity encompasses the situation where two identical concepts are described in distinct ontologies with different terms. This may be due to synonyms (*business partner* and *customer*) or due to different languages (*Finanzinstrument* and *Financial Instrument*), for instance. [124]

Conceptual Heterogeneity Conceptual heterogeneity in ontologies is due to differences in modeling. Concrete reasons are:

1. *Coverage*, originally called *partiality* [34], i.e., two ontologies describe different domains with the same level of detail. There may be an overlap between the two domains. [124]

2. *Granularity*, originally called *approximation* [34], i.e., two ontologies describe the same domain but at different levels of granularity [124].

3. *Perspective*, i.e., two domains describe the same domain but take different unique perspectives [34, 124].

[21] Note that Euzenat is also co-author of this paper.

Semiotic Heterogeneity Semiotic heterogeneity characterizes the situation in which concepts are described identically but are interpreted differently by users. This is due to the fact that interpretations may differ depending on the context in which they are made. [124]

2.6.3 Schema Matching

Doan et al. refer to *schema matching* and *schema mapping* as synonymous [108]. A *semantic mapping* "relates a schema S with a schema T" [108] and "[a] *semantic match* relates a set of elements in schema S to a set of elements in schema T" [108].

2.6.4 Data Model Schema Matching and Ontology Matching

Even though there are differences in data modeling and ontology engineering (Spyns et al. [506] mainly mention higher expressiveness, higher abstraction, and higher application independence of pure ontology models as opposed to database schemas), there are also commonalities: According to the definitions of an ontology provided above, a conceptual data model and even a database schema can be regarded as an ontology. Techniques presented for ontology matching in this thesis (and very often also elsewhere[22]) can also be applied to schema matching of data models or databases. Straightforward approaches exist which allow converting a database schema or *entity-relationship diagram* (ER diagram) into an ontology using OWL by applying a set of rules, for example, as outlined by Fahad [133].

2.6.5 Techniques to Ontology Matching

There is not one superior matching technique or approach in matching ontologies.[23] Rather, there are different types and families of algorithms and approaches used. In this subsection, a categorization of techniques will be pre-

[22] Euzenat and Shvaiko already write in the preface of their book *Ontology Matching* that "though we use the word ontology, the work and the techniques considered in this book can equally be applied to database schema matching [...] and other related problems" [127]. Similarly, in Hepp et al.'s textbook *Ontology Management* [195], Euzenat, Mocan, and Scharffe write: "When we talk about ontologies, we include database schemas and other extensional descriptions of data [...]" [122]. There is also literature where ontology matching is viewed as a form of schema matching, for example, in *Schema Matching and Mapping* by Bellahse, Bonifati, and Rahm [31]. In the latter book, schemas and ontologies are both viewed as metadata models between which mappings can exist [32].

[23] This can easily be seen when looking at the different algorithms applied at campaigns by the OAEI.

sented to better outline the differences and similarities of algorithms and also to classify the matcher developed in this thesis. In 2005, Shvaiko and Euzenat presented two classifications for matching approaches [478] which were revised in 2013 [128].
The first classification approach, called *Granularity/Input Interpretation*, differentiates matchers according to the granularity, which can be either element-level (analyze entities/instances in isolation) or structure-level (analyze the ontology structure), and then according to whether syntactic or semantic techniques are used (*Input Interpretation*). Syntactic techniques use a structured algorithm, whereas semantic techniques apply formal semantics (see Section 2.2).
The second classification approach, called *Origin/Kind of Input*, first differentiates according to whether context (i.e., external resources) or content (i.e., internal resources like the structure or instances) is used (*Origin*) and then further distinguishes different characteristics of the origin (*Kind of Input*). [129] Both classification approaches are depicted in Figure 2.10.

Formal Resource-Based Techniques Formal resource-based techniques make use of external ontologies (which can also be domain-specific). It is also possible to use linked data. [129]

Informal Resource-Based Techniques Informal resource-based techniques utilize informal resources such as pictures or encyclopedia pages. Ontology entities can be related to such resources. [130]

String-Based Techniques A very old class of techniques is represented by string-based techniques, which use annotations – such as names, labels, and descriptions – to calculate similarities between resources. The underlying intuition is that similar words are used to describe similar concepts. [129]

Language-Based Techniques[24] String-based techniques presented above do not require that the language is known in order to be applied. Language-based techniques, on the other hand, consider text encoded in the specified language. Linguistic techniques, like lemmatization or tokenization, can be used here, for example. Phonetic methods, such as *Soundex* [217] or *Kölner Phonetik* [418],

[24] In an earlier version [478], external linguistic resources were explicitly differentiated from plain language-based techniques. The latest version [129] counts everything concerned with the actual language into this category and does not explicitly make this differentiation.

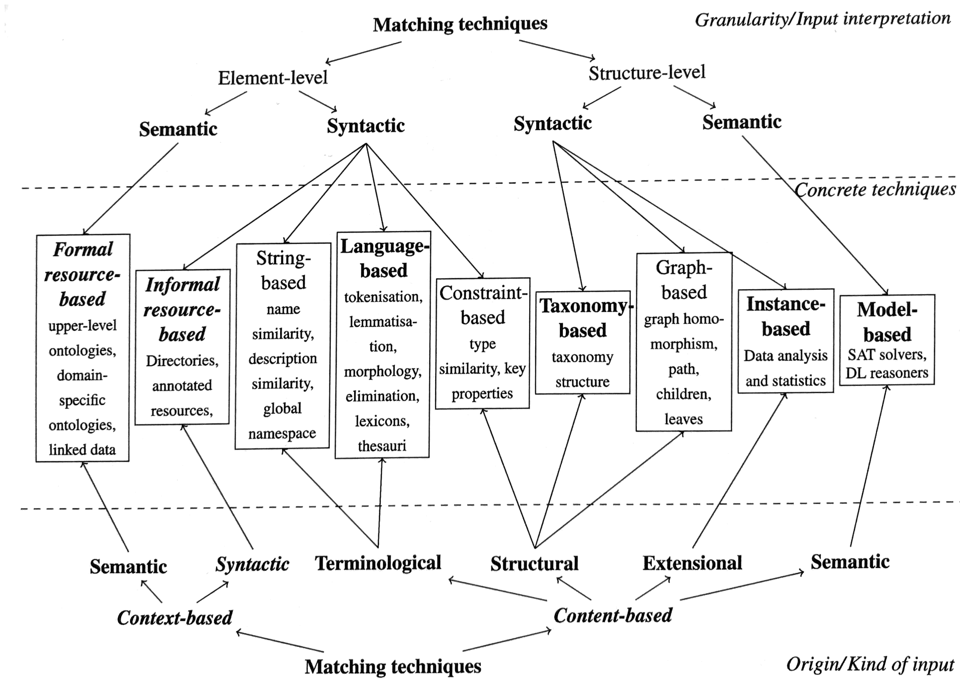


Figure 2.10: Ontology Matching Classification Approaches [129]. Boldly printed concepts are newly introduced compared to [426], and italic-boldly printed concepts were added in the 2013 version [128]. Note that the original figure [478] did not have formal and informal resource-based techniques; those were introduced in the newest version.

also fall into this category. It is, furthermore, possible to exploit external, language-based resources, such as thesauri or lexicons. [129]

Constraint-Based Techniques Constraint-based techniques check internal constraints which apply to entities such as cardinality or data types [129].

Taxonomy-Based Techniques Taxonomy-based techniques apply graph algorithms to the inheritance structure of the resources. The underlying intuition is that concepts that are connected by inheritance are similar. [129]

Graph-Based Techniques Graph-based techniques also view the ontology as a graph. Compared to taxonomy-based techniques, they consider all kinds of information within the graph. Pattern matching methods count as graph-based techniques, for instance. [129]

Instance-Based Techniques Depending on the use case, instances of the ontologies to be matched might be available. Comparing concrete instances can help to calculate distances of resources in the ontologies. Such approaches are referred to as *instance-based techniques*. [129]

Model-Based Techniques Lastly, model-based techniques exploit reasoning and propositional satisfiability in order to match two ontologies. [129]

2.6.6 Evaluation of Ontology Alignments

Measures Ontology alignments are commonly evaluated on the basis of *reference alignments*, i.e., an annotated gold standard of correspondences. In terms of evaluation metrics, the most often used performance measures are *precision*, *recall*, *recall+*/*residual recall*, and *f-score*. These are computed from correctly predicted correspondences (true positives, TP), non-predicted but correct correspondences (false negatives, FN), and incorrectly predicted correspondences (false positives, FP). True negatives, i.e., the correct acknowledgment of a non-existing correspondence, are plentiful in the matching domain and are not relevant for the evaluation metrics. The metrics are quickly introduced in the following:

Precision is the share of correctly found correspondences out of all correspondences proposed by the system:

$$precision = \frac{|TP|}{|TP \cup FP|} \tag{2.1}$$

Recall is the share of correct correspondences that have been found by the matching system:

$$recall = \frac{|TP|}{|TP \cup FN|} \tag{2.2}$$

Residual recall or recall+ refers to the share of correctly found correspondences that are not trivial, where triviality is defined by a baseline reference alignment B [7].

$$recall+ = \frac{|TP \setminus B|}{|(TP \cup FN) \setminus B|} \tag{2.3}$$

The f-measure is a mean of precision of recall – most often, the harmonic mean is used:

$$F_1 = \frac{2 * precision * recall}{precision + recall} \tag{2.4}$$

When evaluating multiple datasets D at once, there are two options for stating one overall performance number: Macro average and micro average. Macro average simply averages scores regardless of the individual datasets' size. The formula is given in an exemplary way for F_1 in Equation 2.5:

$$\Sigma_{d=1}^{|D|} \frac{F_1(d)}{|D|} \tag{2.5}$$

where $F_1(d)$ is the obtained F_1 score on dataset $d \in D$ and $|D|$ is the total number of datasets.
In order to calculate the micro average, one contingency table is built for all datasets by adding all true positives, true negatives, false positives, and false negatives of each individual dataset. Then, precision, recall, and F_1 can be calculated by using this table. [356]

OAEI In order to compare various matchers in a fair setting, common reference alignments are required. The *Ontology Alignment Evaluation Initiative* (OAEI)[25] tackles this problem by providing several reference alignments and carrying out campaigns every year since 2004. Participants can evaluate their matchers in several tracks. [131] One major goal of the OAEI is to create transparency and "to allow anyone to draw conclusions about the best matching strategies" [479].
For the alignment evaluation, the *Semantic Evaluation At Large Scale* (SEALS)[26] platform was originally used. Starting in 2017, the OAEI was beginning to use the *Holistic Benchmarking of Big Linked Data* (HOBBIT)[27] platform where users are able to upload and evaluate matching systems [448] – nonetheless, HOBBIT never gained much traction at the OAEI. Since 2020, the OAEI changed the evaluation platform to MELT, which is compatible with both SEALS and HOBBIT and is one of the contributions of this dissertation. More details on MELT are provided in Part II of this dissertation. More details and statistics on the OAEI are provided in Section 3.3.3 of the subsequent chapter.

[25] see `http://oaei.ontologymatching.org`
[26] see `http://seals-project.eu/`
[27] see `https://project-hobbit.eu/`

Chapter 3

Review of the Research Field

The previous chapter presented the fundamental concepts of this dissertation. In the survey covered in this chapter, we broaden the scope and analyze the state of the art in detail. We review the background knowledge sources as well as the approaches applied to make use of external knowledge. Our survey covers all ontology matching systems that have been presented within the years 2004 – 2021 at a well-known ontology matching competition together with systematically selected publications in the research field. We present a classification system for external background knowledge, concept linking strategies, as well as for background knowledge exploitation approaches. We provide extensive examples and classify all ontology matching systems under review in a resource/strategy matrix obtained by coalescing the two classification systems. Lastly, we outline interesting and yet underexplored research directions for applying external knowledge within the ontology matching process.

It is important to emphasize that this survey already includes several contributions to this dissertation, namely [203, 400, 204, 410, 413, 278, 281, 404, 403, 402, 401, 396].

This chapter constitutes the main part of the related work of this dissertation. In cases of subsequent chapters which focus on a very specific topic that is not covered in this chapter, a short section on related work can be additionally found there.

The work presented in this chapter has been published before as: Portisch, Jan; Hladik, Michael; Paulheim, Heiko. Background Knowledge in Ontology Matching: A Survey. Semantic Web Journal (SWJ). 2022.

3.1 Introduction

Ontology matching is the non-trivial task of finding correspondences between entities of two or more given ontologies or schemas. It is an integral part to ensure semantic interoperability. The matching can be performed manually or through the use of an automated matching system. Ontology matching is a problem for Open Data (e.g., matching publicly available domain ontologies or interlinking concepts in the *Linked Open Data Cloud*[1]) as well as for private companies which need to integrate disparate data stores for transactional or analytical purposes.

A major challenge for matching ontologies is the fact that they are typically designed within a given context and deep background knowledge that is not explicitly expressed in the schema definition [130]. In order to automatize the ontology matching process, external background knowledge is therefore required so that the automated matching system can interpret, for example, textual labels and descriptions of the elements within the schemas that are to be matched.

Current surveys in the ontology matching [378, 20, 25, 353] and schema matching [518, 18] domain classify matching systems according to their matching technique (strongly influenced by Euzenat and Shvaiko [478, 129] as well as Rahm and Bernstein [426]) with minor or no emphasis at all on the background knowledge used.

In the area of context-based matching, i.e., matching with intermediate resources, Locoro et al. [326] present an abstract seven-step process for context-based matching together with an experimental evaluation of different parameter configurations. The proposed framework is flexible but experimentally focused on ontologies as background knowledge and a path- and logic-based exploitation approach. The survey at hand takes a broader look at the types of background sources and different exploitation strategies used in research, including, for instance, unstructured data and statistical or neural approaches.

A recent survey by Trojahn et al. [537] provides a detailed perspective into foundational ontologies in ontology matching, which includes, among other use cases, the exploitation of those for the task of matching domain ontologies. The survey presented here is broader in the sense that foundational ontologies are considered only as *one* kind of external background knowledge; it is narrower in the sense that it focuses purely on the use case of finding equivalence relations between schemas with additional background knowledge automatically.

Thiéblin et al. [528] review complex matching systems, i.e., systems that are capable of generating correspondences involving multiple entities, transforma-

[1] see `https://lod-cloud.net/`

tion functions, and logical constructors. The matching systems covered in their survey use different knowledge representation models (including table-based or document-based schemas, for instance). The systems are characterized based on the correspondence output and the underlying process type, which generated the complex alignment. Background knowledge is not discussed and does not play a major role in the current implementations of complex matching systems. The survey at hand is complementary in the sense that it focuses on systems producing simple equivalence correspondences through the use of background knowledge.

This comprehensive survey reviews an extensive set of ontology matching and integration systems published in the last two decades in terms of the background knowledge used and in terms of the strategy that is applied to exploit the external background knowledge. It further covers the approaches used to link schema concepts to background knowledge. Based on the extensive collection of reviewed systems, we provide a comprehensive overview of background knowledge sources and strategies used in the past. Furthermore, this survey reveals a number of blind spots that have not yet been thoroughly explored.

In the following, the selection method for publications used in this survey is presented (Section 3.2.1). Afterward, the core theoretic concepts are introduced in Section 3.3, namely schema matching and *ontology matching* (OM). In Section 3.4, background knowledge is defined, its usage in ontology matching system is analyzed, and the most used resources are presented. Thereupon, classification systems for background knowledge sources (Section 3.5), concept linking approaches (Section 3.6), and exploitation approaches (Section 3.7) are presented together with examples. In Section 3.8, we outline interesting directions for future work in the research field.

3.2 About this Survey

3.2.1 Selection of Publications

Search Parameters For this survey, we defined three search parameters: *(Q1) "ontology matching"*, *(Q2) "ontology alignment"*, and *(Q3) "ontology mapping"*. We queried publications via the *dblp computer science bibliography* (DBLP)[2] without further filters. The search criteria have been intentionally chosen to be very broad since the usage of background knowledge is very often not indicated in the title or abstract of a paper.
We further manually added *all* matching systems that participated in the schema

[2] see `https://dblp.org/`

Q1 “ontology matching” on DBLP	589
Q2 “ontology alignment” on DBLP	514
Q3 “ontology mapping” on DBLP	570
OAEI system papers	242
De-duplicated papers	1,814
Included papers	341

Table 3.1: Search parameters and the associated number of papers.

matching tracks of the Ontology Alignment Evaluation Initiative (OAEI, see Section 3.3.3) from its inception in 2004[3] until 2021 [517, 26, 495, 481, 482, 484, 483, 485, 493, 496, 494, 486, 489, 487, 488, 490, 491, 492].

The number of retrieved papers for each search parameter can be found in Table 3.1. The BibTeX files can be found in the GitHub repository of this survey.[4]

De-Duplication The BibTeX files of all publications were gathered and loaded via the *Zotero*[5] bibliographic management tool. The latter was used to detect duplicate publications based on the metadata of the papers. All scientific artifacts were exported as a *comma-separated values* (CSV) file, including the metadata (title, authors, publication venue, date, etc.) for manual de-duplication.

The resulting set of papers constitutes the final set of publications used for identifying relevant works for this survey. In total, 1,814 papers were considered in this study.

Selection Process In order to identify papers that are relevant for this survey, *inclusion criteria* (IC) and *exclusion criteria* (EC) were defined. The set of all papers was manually scanned in order to filter out publications not relevant to this survey. The complete list of inclusion and exclusion criteria is shown in Table 3.2. Every paper that is considered in this survey has to match all inclusion criteria.

Papers considered in this survey had to be written in English language (C1), had to be accessible through the infrastructure of a large German research university (C2), and had not to be a duplicate of another paper (C3). It is important to note that multiple publications on the same topic (such as a matching system)

[3] Back then the competition was actually referred to as *EON Ontology Alignment Contest*.

[4] see `https://github.com/janothan/bk-in-matching-survey/`

[5] see `https://www.zotero.org/`

Criteria		Inclusion Criteria (IC)	Exclusion Criteria (EC)
C1	Language	The paper is written in English.	The paper is not written in English; the paper is written in English but heavily ungrammatical.
C2	Accessibility	The paper can be accessed through the infrastructure of the University of Mannheim without additional payment.	The paper cannot be accessed through the infrastructure of the University of Mannheim without additional payment.
C3	Duplication	Included are papers whose content is unique. This explicitly includes papers on the same matching system; for example, all OAEI LogMap papers are included in this survey rather than only the latest publication in order to carry out a thorough time analysis.	Excluded are papers with identical content, such as preprints that are identical in content with their peer-reviewed publications or identical papers published in multiple venues.
C4	Ontology Matching System	The paper presents a matching system, i.e., a system which accepts two ontologies and returns an alignment. The matching system must be able to match ontologies (T-box). Papers that align schema *and* instances are also included.	The paper does not present a matching system that is able to match ontologies such as pure entity-linking or pure instance matching approaches.
C5	Simple Correspondences	The matching system produces simple correspondences.	The paper presents a matching system for complex matching.
C6	Background Knowledge	The matching system exploits *some* form of external knowledge.	The matching system presented does not use any external knowledge.
C7	Application/Evaluation	The paper presents a matching system that is evaluated on the task of ontology matching.	The paper merely describes a framework or a theoretical idea but lacks a concrete implementation regarding ontology matching.
C8	Level of Detail	The paper describes the use of background knowledge with an appropriate level of detail.	The usage of background knowledge is mentioned, but it is unclear which knowledge source is used or how it is used.

Table 3.2: Inclusion and exclusion criteria for the papers in this survey.

do not qualify as duplicates despite their potentially large content overlap. This is rooted in the observation that there are often multiple versions and papers of a single matching system, which evolves over time (for example Agreement-MakerLight *(AML)* [147] or *LogMap* [241]); in such cases, we always refer to the specific matching paper we mean in order to be precise rather than referencing the most current or most extensive paper published for the system in question.

We explicitly exclude works limited solely to instance matching or entity linking (C4). We further focus on matching systems that produce simple correspondences rather than complex ones (C5). Lastly, we only cover papers that present an actual system, i.e., a background knowledge-based (C6) ontology matching system implementation (C7) for which an evaluation is presented. The usage of the background knowledge must be appropriately documented (C8). In total, 341 papers fulfilled the inclusion criteria of this survey.

All matching systems were systematically evaluated in terms of (i) the background knowledge sources used, (ii) the strategy deployed to link ontology concepts to the background knowledge source, and (iii) the strategies the matching systems apply to exploit the background knowledge sources.

3.2.2 Figures and Data

All data points and code used for the quantitative analysis of this survey are available online.[6] This includes statistical figures, which are also available online in a higher resolution; they can further be re-generated with the provided Python code.

3.3 Schema Matching and Ontology Matching

3.3.1 Schemas and Ontologies

The focus of this survey is a special case of the first step of the DI process (see Section 2.1), schema matching. It is important to note that a schema is not bound to a technology stack. It is, for example, possible that the same schema is implemented on different technology stacks, such as different database types. Many formalization notations for schemas have evolved over time – for example, in the area of (conceptual) entity-relationship models *Barker's notation* [28], *IDEF1X* [50] by the *National Institute of Standards and Technology* (NIST), or *MERISE* [525]. In semantic data modeling, data representation paradigms, such as controlled vocabularies, taxonomies, or knowledge graphs, among others, are

[6] see `https://github.com/janothan/bk-in-matching-survey/`

used [13], all of which have been subsumed under the umbrella term of ontologies in different publications [294, 540, 382, 429, 116]. Hence, we conclude that most of the methods described for ontology matching can be more broadly understood as methods for matching semantic models in general [127].

3.3.2 The Ontology Matching Problem

Ontology Matching Given two ontologies O_1 and O_2, the matching problem describes the task of finding an alignment A between O_1 and O_2. An alignment is a set of correspondences whereby a correspondence is a triple in the form $\langle e_1, e_2, r\rangle$ with $e_1 \in O_1$ and $e_2 \in O_2$ being elements of the ontologies to be matched and r being the relation that holds between the two elements. Examples for the relation are equivalence ($\equiv$) or inclusion ($\sqsubseteq$). A correspondence may optionally have an explanation e and a confidence value c assigned to it and is, therefore, sometimes also described as a quintuple in the form $\langle e_1, e_2, r, c, e\rangle$. Two types of correspondences are distinguished: Simple ones that link one element from O_1 to one element from O_2 and complex ones, i.e., correspondences that contain logical constructors or transformation functions [529].

A matching system can be seen as a function $f(O_1, O_2, A', p, b) = A$. Variable A' refers to an existing alignment (which may be empty), p specifies additional parameters for the matching process, and b[7] represents external background knowledge sources used in the matching process. [124] For this survey, it is of particular interest how b is used in f.

For more details concerning ontologies and the ontology matching problem, we direct the reader to Sections 2.4 and 2.6 of Chapter 2.

Ontology Integration Multiple interpretations exist to the terms *ontology integration* and *ontology merging*. We follow the proposal from Osman et al. [377] in this survey and regard ontology merging as a special case of ontology integration:

Ontology integration (also referred to as ontology enrichment, ontology inclusion, or ontology extension) describes the process of extending a given target ontology O_T with another (source) ontology O_S given an alignment A_{S-T} between O_S and O_T: $Integrate(O_S, O_T, A_{S-T}) = O_T$. A special case is ontology merging where given two ontologies O_1 and O_2, a third ontology O_3 is derived given an alignment A_{1-2} between O_1 and O_2: $Merge(O_1, O_2, A_{1-2}) = O_3$. According to Osman et al. [377], the ontology integration process can be generally seen as a four step process:

[7] Originally called r but renamed for better clarity here.

1. Pre-processing Phase
2. Matching Phase
3. Merging Phase
4. Post-processing Phase

Pre-processing describes preparing the ontology files that are to be matched, e.g., by converting them into the same uniform representation. The *Matching Phase* describes the ontology matching process as outlined in the previous paragraph. The *Merging Phase* describes the execution of the $Integrate/Merge$ operator, and the *Post-processing Phase* summarizes various amendments to the resulting ontology to improve its quality, such as resolving cycles or coherence and conservatory violations. For details, we refer the reader to the comprehensive survey by Osman et al. [377].

In this chapter, we also cover papers and systems which address the ontology integration problem where background knowledge plays a significant role in the matching phase. In figures and tables, those systems are notated with a subscript I such as MoA_I.

3.3.3 The Ontology Evaluation Initiative since 2004

About the OAEI Schema matching can be performed manually, through an automated matching system, or in a hybrid environment. For systematically evaluating the latter two cases, the *Ontology Alignment Evaluation Initiative* (OAEI)[8] has been running campaigns every year since 2004. Unlike other evaluation campaigns where researchers submit datasets as solutions to report their results (such as *Kaggle*[9]), the OAEI requires participants to submit a matching system, i.e., an implemented and packaged matching system, which is then executed on-site.[10] In order to do so, multiple frameworks and platforms for standardized matcher development, packaging, and evaluation have been developed and are used by OAEI participants, namely the *Alignment API* [96] format and framework, the *SEALS* [162, 577] and *HOBBIT* [368] packaging and evaluation platforms as well as *MELT* [203, 400, 204], a framework for matcher development, packaging, and evaluation, which also integrates with the aforemen-

[8] see `http://oaei.ontologymatching.org/`

[9] see `https://www.kaggle.com/`

[10] Prior to 2010, participants submitted resulting alignments directly. The submission of packaged tools (at first in the form of URLs of Web services running on the participants' site) instead of results was started in 2010. Since 2012, the submission of packaged tools has been the standard evaluation procedure at the OAEI.

tioned frameworks. After the evaluation, the results are publicly reported. The individual matching tasks are referred to as *test cases* which are bundled in *tracks*. Originally, the OAEI started with plain ontology matching tracks focused on simple alignments with an equality relation, i.e., a correspondence that contains only one entity from the source ontology and one ontology from the target ontology and where $r = equivalence$. More recently, new tracks have been introduced, such as the *Knowledge Graph Track* [216, 202] which combines schema and instance matching tasks. The most transparent way of presenting and benchmarking a new matching system is the participation in an OAEI campaign – however, most datasets are also available for download[11] and can be used outside of OAEI campaigns to evaluate matching systems.

OAEI Tracks Figure 3.1 summarizes all OAEI schema matching tracks since the inception of the initiative. As visible in the figure, some older tracks have been discontinued[12] while new tracks have also been introduced. All current schema matching tracks that were evaluated in the OAEI 2020 and 2021 are listed in Table 3.3 together with a quick description and the best performing system of the corresponding year.

OAEI Matching Systems Since 2004, many matching systems have been submitted and evaluated. Figures 3.4 and 3.5 list all matching systems that have been evaluated in OAEI schema matching campaigns[13] since its inception on the y-axis; the x-axis represents a timeline, and the black bars represent the time frame in which the systems have participated in the campaigns. As visible in the figures, many systems have been evaluated in multiple campaigns. For this survey, all of the listed matching systems that are used for schema matching have been examined in terms of what background knowledge source is used, if any, how a connection between the ontologies and the background knowledge source is established, and how the background knowledge source is exploited.

[11] see `https://dwslab.github.io/melt/track-repository`

[12] The discontinuation of tracks is often due to missing track organizers. Reasons may be the high effort connected to evaluating other researchers' matching systems and writing summarizing reports or a change in the research focus. However, most track data is still available for download and for further usage.

[13] The tracks which were considered are listed in Figure 3.1. Figures 3.4 and 3.5 do not include other evaluation tracks such as team participations in the SemTab [247] track. Due to very high similarity, the following matching systems have been merged in the figure: *NLM* [609] and *AOAS* [610], *Agreement Maker* and *AMExt* (both described in [87]), as well as *GeRoMe* [423, 424] and *GeRoMe SMB* [422].

Track	**Track Description**	**Best Performing System in the OAEI 2020**	**Best Performing System in the OAEI 2021**
Anatomy [42]	An alignment between the Adult Mouse Anatomy and a part of the NCI Thesaurus is to be found.	AML [312] (Uberon, DOID, MeSh, WordNet, Microsoft Translator, OBO logical definitions)	AML [140] (Uberon, DOID, MeSh, WordNet, Microsoft Translator, OBO logical definitions)
Conference [77]	16 ontologies from the conference domain have to be matched.	VeeAlign [227] (Google Universal Sentence Encoder)	AML [140] (see above)
Multifarm [337]	7 conference ontologies translated into 8 languages (+ English) have to be matched.	AML [312] (see above)	AML [140] (see above)
LargeBio	An alignment between 3 large bio ontologies is to be found.	AML [312] (see above)	AML [140] (see above)
Phenotype [185]	An alignment between two disease and two phenotype ontologies is to be found.	LogMapBio [239] (Bioportal)	LogMap [240] (SPECIALIST, Microsoft Translator) LogMapBio [240] (Bioportal) AML [140] (see above)
Biodiversity and Ecology [260]	4 matching tasks from the biodiversity and ecology domains.	AML [312] (see above)	AML [140] (see above)
Knowledge Graph [199]	5 matching tasks consisting of knowledge graphs extracted from fandom.com.	Wiktionary Matcher [410] (Wiktionary/DBnary)	Wiktionary Matcher [413] (Wiktionary/DBnary)
Common Knowledge Graph [136]	An alignment between the classes of two large, automatically constructed knowledge graphs is to be found.	–	KGMatcher [137] (BERT, Google language model)

Table 3.3: Depicted are all schema matching tasks of the OAEI 2020 and 2021 together with the best-performing systems in terms of F_1. For the conference track, the rar2-M3 results have been used to determine the best system. For tracks with multiple tasks that do not name a best-performing system (LargeBio, phenotype), the average position in all tasks was chosen as criterion to determine the best-performing system here. The Common Knowledge Graph track was first evaluated in 2021.

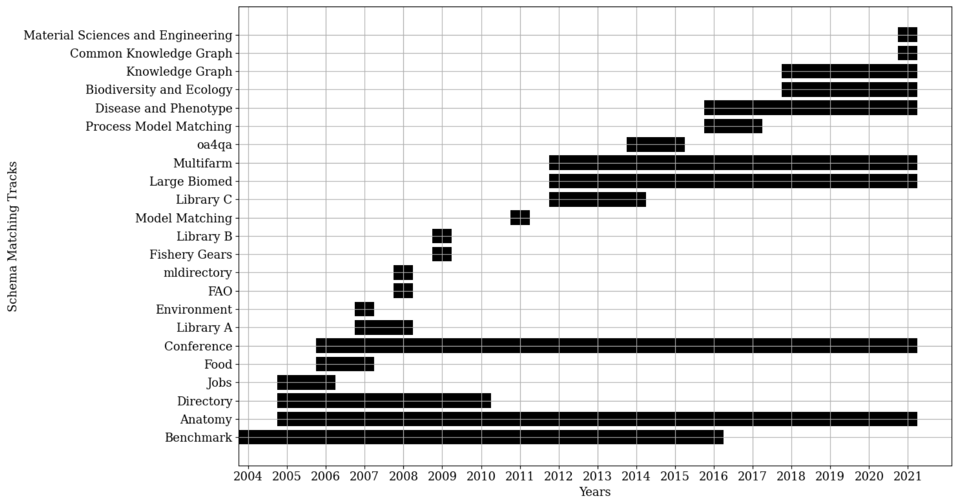


Figure 3.1: OAEI schema matching tracks since the inception of the initiative. Explicitly excluded are complex matching tracks and instance matching tracks. The knowledge graph track is not a pure schema matching task but a combined one where schemas and instances have to be matched simultaneously. The library track has been organized multiple times with completely different datasets and by different researchers using the same track name. Therefore, the track streams have been divided into three groups (A, B, C).

Figure 3.2 reveals that over the years, the number of participating schema matching systems to date has slightly dropped from the peak in the year 2012, albeit the current participation total is still comparatively high compared to the early days of the initiative.[14]

[14] Figure 3.2 has been compiled from Figures 3.4 and 3.5, hence the concrete number of schema matching systems is counted each year excluding pure instance matching systems. The OAEI does not calculate this statistic. In addition, we found that over the years, the OAEI counted inconsistently with regards to participation (for example, counting participating teams in 2012 but matching systems in 2013 on their results Web page).

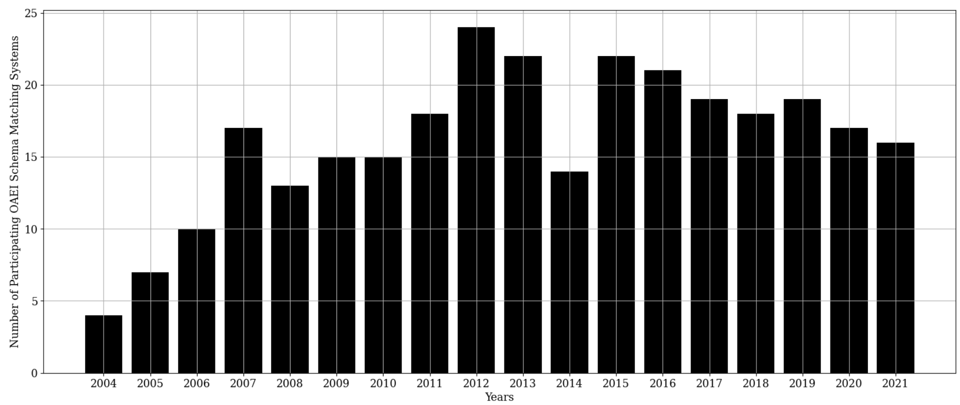


Figure 3.2: The number of ontology matching systems participating in the OAEI from inception to date.

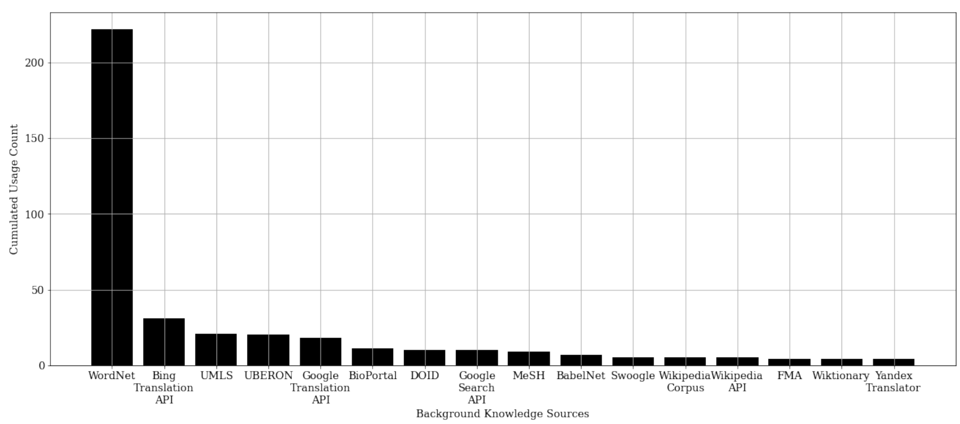


Figure 3.3: Cumulative usage of a particular knowledge source of all systems in this survey within the years 2000 to 2021.

Table 3.3 lists all schema matching tracks from 2020 and 2021 together with the best performing system and the background knowledge sources used by those. As visible in the table, all those systems make use of external knowledge datasets. AML, which scores as the best-performing system in multiple tracks, exploits multiple external knowledge sources.

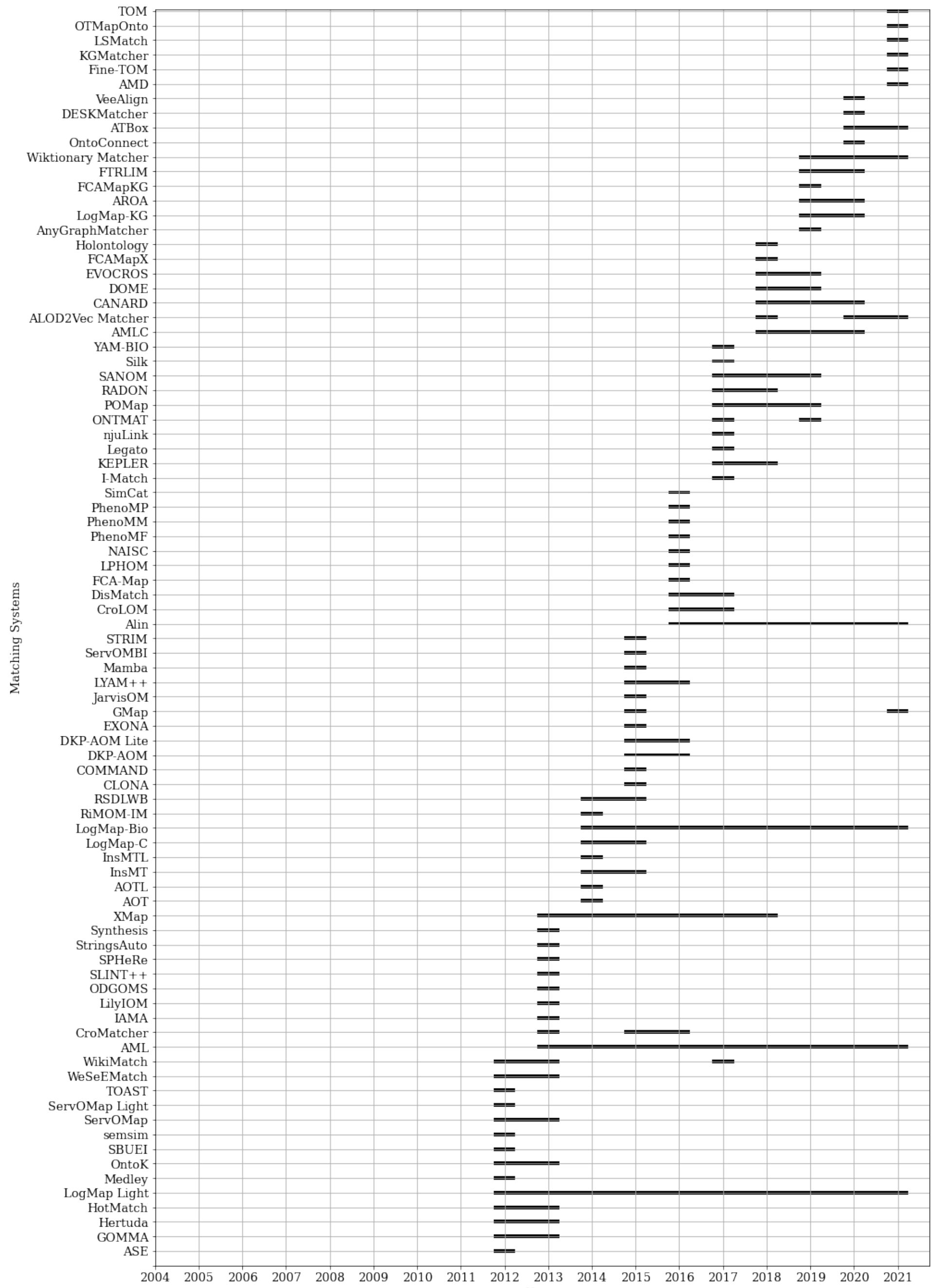


Figure 3.4: All OAEI schema matching systems (which participated in the tracks listed in Figure 3.1) and their evaluation time frame since the inception of the OAEI; Part 1 of 2 from 2012 - 2021.

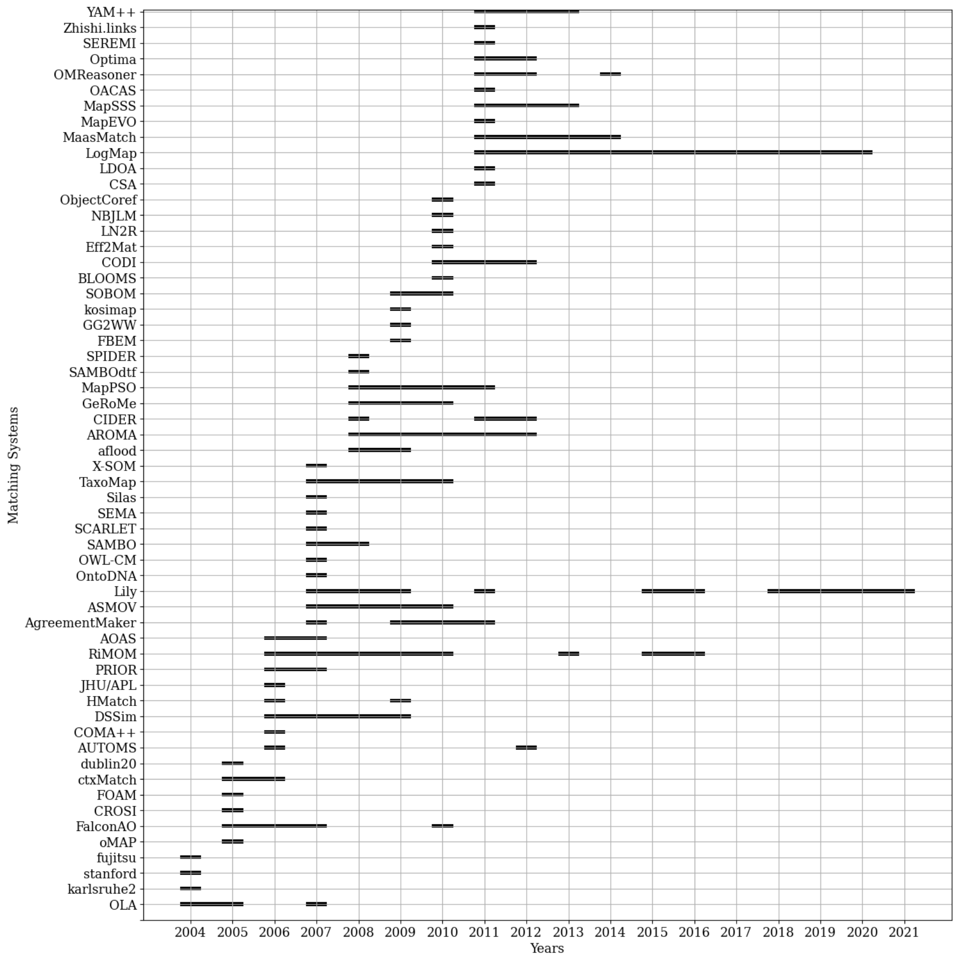


Figure 3.5: All OAEI schema matching systems (which participated in the tracks listed in Figure 3.1) and their evaluation time frame since the inception of the OAEI; Part 2 of 2 from 2004 - 2021.

3.4 Background Knowledge in Ontology Matching

3.4.1 Background Knowledge

We define background knowledge in matching as any knowledge source that is external to the matching process and is used to obtain the final alignment. Hence, within the matching process, external knowledge can be used in the form of an existing alignment (A') or in the form of a resource that is independent of the matching task. The resource used is technology-independent and may also be represented as an API, for example.

Background knowledge can significantly improve the performance of ontology matching systems. This is clearly visible by analyzing different OAEI systems: When comparing LogMap and LogMapBio [240] in the OAEI 2021 campaign, for instance, it can be seen that the latter system scores a significantly higher recall on the OAEI Anatomy dataset. Other examples can be found through a comparison of AML [145] and Gomma[15] in the 2013 campaign: Both systems participated in two configurations – with and without background knowledge. On the Anatomy track, the background knowledge configurations significantly outperformed all other systems in terms of recall and F_1. Another indicator of the value of background knowledge is the fact that *all* best-performing schema matching systems of the 2020 and 2021 campaigns use external background knowledge (see Table 3.3).

In [144], Faria et al. evaluate strategies for matching biomedical ontologies. The experiments show a clear performance increase when background knowledge is used. In terms of exploitation strategies, the authors recommend using cross-references (if available) over lexical expansion.

While evaluating an approach to building a background knowledge resource for ontology matching, Annane et al. [23] also analyze the performance of the YAM++ matching system with and without background knowledge finding that the matcher configuration which uses background knowledge significantly outperforms the version without additional resources. They report that the better performance is mainly due to a higher recall.

In an extensive survey on the systems participating in the OAEI Anatomy track from 2007 to 2016, Dragisic et al. report that "[f]or the systems that participated with a version using biomedical auxiliary sources and a version not using biomedical auxiliary sources, the F-measure for the one with biomedical auxiliary sources was always higher" [113].

Missing background knowledge was named as one of the ten challenges for ontology matching in 2008 [479]; this was re-affirmed in 2013 [130], and it is still under active research.

3.4.2 Background Knowledge Selection in Ontology Matching

As there are often multiple potentially beneficial sources of background knowledge available for ontology matching, some authors propose heuristics to determine the benefit of a background knowledge source in order to select one before

[15] There is no results paper for the OAEI 2013 participation of Gomma. However, the system is described in the paper of the 2012 campaign [172].

performing the match operation. Nasser et al. [532] define four criteria for automatic background knowledge selection:

1. *type independence*: A selection system should be capable to handle various serialization formats.

2. *domain independence*: A selection system should be domain-independent and be able to select sources for any domain.

3. *multilingualism*: A selection system should be language-independent, i.e., support cross-lingual ontology matching.

4. *optimality*: A selection system should return the best background knowledge source from the corpus.

Based on their universal requirements, they propose an approach that models the selection task as an information retrieval problem. Ontologies and background sources are indexed using *term frequency–inverse document frequency* (TF-IDF); the ontologies are then regarded as query on the background knowledge sources.

In the LogMapBio system, Chen et al. [81] apply a relatively simple lexical algorithm to identify suitable mediating ontologies from BioPortal [164, 571]. In the OAEI 2020 campaign, the system achieved a significantly higher recall and F_1 measure than the classic LogMap matching system.

Faria et al. [146] propose a heuristic called *Mapping Gain* which is based on the number of additional correspondences found given a baseline alignment. Quix et al. [425] use a keyword-based vector similarity approach to identify suitable background knowledge sources. Similarly, Hartung et al. [187] introduce a metric, called *effectiveness*, which is based on the mapping overlap between the ontologies to be matched.

3.4.3 Background Knowledge in Ontology Matching Over Time

Tables 3.4 to 3.7 list all background knowledge sources that have been used by the systems evaluated in this survey together with the actual systems that use the corresponding knowledge source. As multiple papers exist for some systems, the first documented usage of the knowledge source by the matching system is referenced. Consequently, there is no guarantee that the latest system still uses the specified sources. *WeSeE Match*, for example, used the *Microsoft Bing* search engine in its 2012 version [383] but switched to the *FARO Web Search* framework in 2013 [385]. Therefore, different papers are referenced for the system. For each knowledge source, the systems in column *Used by System* are ordered according to the publication year. Since this survey covers a large time period, not all resources used in the past are still available; therefore, column *Resource Available* indicates whether the resource is still available to researchers. Due to the frequent usage of *WordNet* [149], systems that use this source are listed in Tables 3.8 and 3.9, which are organized according to the same methodology as Tables 3.4 to 3.7. Tables 3.4 to 3.9 also include some non-OAEI matching systems (indicated by italics).

Figure 3.3 shows the cumulative usage of background knowledge sources that have been referenced in at least four different publications. The by far most often used external knowledge resource is *WordNet* [149]. Further often used resources are the Unified Medical Language System *(UMLS)* [318] as well as the *Microsoft Bing Translation API*. When looking at the distribution of the usage counts in Figure 3.3, a power-law distribution can be recognized: Most systems use the same knowledge source; although many knowledge sources exist, most are used only by very few systems. It is important to note that the long-tail in the distribution is actually much longer, as shown in the figure, because the latter only lists sources used by at least four different matching system publications.

In Figure 3.6, background knowledge source usage is plotted over time. As in the figure before, only sources are depicted, which are used at least four times by the papers included in this survey. What is visible from the figure (and also from Tables 3.4, 3.5, 3.6, 3.7, 3.8, and 3.9) is that background knowledge has been used from very early on. In the first OAEI in 2004, for example, the *OWL-Lite Alignment* (OLA) [120] matching system already uses WordNet to retrieve synonym sets. A look at the usage over time (Figure 3.6) reveals that only a few sources have been used in the early days of ontology matching. With the progression of time, more and more resources are evaluated. However, only a few sources show a consistently high application, in particular WordNet, the Microsoft Translation API, UBERON, and UMLS. We can also observe spikes of usage, i.e., a resource has been used within a short time frame in multiple papers but not afterward: Examples here are *Swoogle* [103], a Semantic Web search engine[16], or the *Google Search API*.

[16] The search engine is not online anymore.

Knowledge Source	**Source Description**	**Resource Available**	**Used by System**
Apertium [155]	A free open-source platform for machine translation.	yes	*Bella et al.* (2017) [30]
BabelNet [362]	Multilingual, large knowledge graph derived through the integration of multiple knowledge sources such as WordNet and Wikipedia.	yes	LYAM++ (2015) [530] *Helou et al.* (2016) [192] *Biniz et al.* (2017) [37] EVOCROS (2018) [99] *Kolyvakis et al.* (2018) [280]
BERT [102]	A transformer-based language model.	yes	*Neutel et al.* (2021) [363] KGMatcher (2021) [137] Fine-TOM (2021) [278] TOM (2021) [281]
Big Huge Thesaurus	Web API for synonyms and antonyms.	yes	*SOCOM++* (2012) [160] HotMatch (2012) [94]
Bing Search Engine API	Cloud API for the Microsoft Bing Web search engine.	yes	*Fu et al.* (2011) [159] WeSeE Match (2012) [383] *SOCOM++* (2012) [160] SYNTHESIS (2013) [285]
Bing Translator / Microsoft Translator	Cloud API for the Microsoft Bing translation service.	yes	*SOCOM* (2010) [158] *Spohr et al.* (2011) [505] WeSeE Match (2012) [383] YAM++ (2012) [365] *Koukourikos et al.* (2013) [284] AML (2014) [141] XMap (2014) [105] *Kachroudi et al.* (2014) [257] LogMap (2015) [245] CLONA (2015) [2] KEPLER (2017) [253] *Kachroudi & Yahia* (2018) [256]
BioBERT [298]	A language model pre-trained on medical text.	yes	*MEDTO* (2021) [184]
BioPortal [164, 571]	A repository of interlinked biomedical ontologies.	yes	LogMapBio (2014) [246] *Annane et al.* (2016) [22] *Zaveri & Dumontier* (2016) [605] Lily (2018) [524] *Annane et al.* (2018) [23]
ConceptNet [503]	A freely-available word graph collected from multiple sources.	yes	*Kolyvakis et al.* (2018) [280]
Cooking Dictionary	A collection of term definitions in the cooking domain.	yes	*van Hage et al.* (2005) [179]
DBpedia [300]	A knowledge graph extracted from Wikipedia info boxes.	yes	*BLOOMS* (2010) [231] LDOA (2011) [255] *Grütze et al.* (2012) [175]
DOID [466]	The Human Disease Ontology (DOID).	yes	AML (2014) [141] *Ochieng & Kyanda* (2018) [376] *Annane et al.* (2018) [23]
DOLCE [161]	The descriptive ontology for lingusitic and cognitive engineering (DOLCE) is an upper ontology.	yes	*Mascardi et al.* (2010) [334] *Davarpanah et al.* (2015) [95]
FAROO Web Search	A framework for Web search.	yes	WeSeE Match (2013) [385]
fastText model	A model trained with facebook's AI reserach (FAIR) fastText [43] framework.	yes	OntoConnect (2020) [71] *Neutel et al.* (2021) [363]

Table 3.4: Knowledge sources and matching systems that use them (part 1 of 4). Referenced is the first documented usage by the matching system. Systems that did not participate in the OAEI are italicized. Named systems are referred to using their system name.

Knowledge Source	**Source Description**	**Resource Available**	**Used by System**
FIBO	The Financial Industry Business Ontology (FIBO).	yes	DESKMatcher (2020) [351]
FMA	The Foundational Model of Anatomy (FMA).	yes	AOAS (2007) [610] *Groß et al.* (2011) [171] GOMMA (2012) [172] *Petrov et al.* (2013) [392]
Google NNLM	A neural text embedding model available through TensorFlow Hub by Google.	yes	KGMatcher (2021) [137]
Freelang	A translation API (available as offline and as online version).	yes	Medley (2012) [188]
Google Search API	Cloud API for the Google Web search engine.	yes	*Pan et al.* (2005) [379] *van Hage et al.* (2005) [179] *PROMPT-V* (2007) [250] X-SOM (2007) [92] *Gligorov et al.* (2007) [168] *KMSS* (2009) [606] *Mao et al.* (2011) [333] MapSSS (2013) [76] *Jiang et al.* (2014) [238]
Google Translation API	A translation Web API by Google.	yes	*SOCOM* (2010) [157] *Fu et al.* (2011) [159] *SOCOM++* (2012) [160] RiMom (2013) [615] LogMap (2014) [246] *Helou et al.* (2016) [192] *NuSM* (2017) [30] *Destro et al.* (2017) [98]
Google Universal Sentence Encoder [69, 602]	Pre-trained encoder by Google (monolingual [69] and multilingual [602]).	yes	VeeAlign (2020) [227]
Google Word2Vec Vectors	Word2vec models by Google.	yes	*Bulygin* (2018) [56] *Bulygin & Stupnikov* (2019) [57]
HowNet [110]	An online sememe knowledge base in Chinese and English.	yes	*Li et al.* (2006) [306] *Wang et al.* (2008) [566]
ImageNet	A large database of images.	yes	*Doulaverakis et al.* (2015) [111]
iTranslate4	API for machine translation.	no	*Koukourikos et al.* (2013) [284]
KGvec2go [404]	Pre-trained RDF2vec embeddings.	yes	ALOD2Vec (2020) [403]
Lanes API	Language Analysis Essentials (LANES) API. Does not seem to be online anymore.	no	HotMatch (2012) [94]
Medical Subject Headings (MeSH) [513]	The Medical Subject Headings (MeSH) are a controlled vocabulary thesaurus.	yes	AML (2014) [141] *Ochieng & Kyanda* (2018) [376] *Real et al.* (2020) [428] *Annane et al.* (2018) [23]
Medline	Bibliographic database of the National Library of Medicine. Medline is a subset of PubMed.	yes	DisMatch (2016) [447] *OntoEmma* (2018) [558]
MyMemory API	A translation REST API provided by translated.com.	yes	GOMMA (2012) [172]
Ontology Lookup Service (OLS)	Repository and Web APIs for biomedical ontologies.	yes	*PAXO* (2020) [186]
OpenCyc [302]	Open-source version of the Cyc knowledge base by Cycorp. No longer available.	no	*Mascardi et al.* (2010) [334] *Davarpanah et al.* (2015) [95]
Paraphrase DB (PPDB) [386]	A very large collection of paraphrases.	yes	*DeepAlignment* (2018) [279]
PubMed	Bibliographic database maintained by the National Library of Medicine.	yes	*Fang et al.* (2013) [138] *Li* (2020) [305]

Table 3.5: Knowledge sources and matching systems that use them from part (2 of 4). Referenced is the first documented usage by the matching system. Systems that did not participate in the OAEI are italicized. Named systems are referred to using their system name.

Knowledge Source	**Source Description**	**Resource Available**	**Used by System**
RadLex	A radiology lexicon.	yes	*Groß et al.* (2011) [171]
SAP Term	Definitions of terms in SAP software.	not publicly	DESKMatcher (2020) [351]
SBERT [432]	A BERT modification so that similarity can be determined via cosine distance	yes	*MEDTO* (2021) [184]
SDL FreeTranslation	An online translation service.	no	*SOCOM* (2010) [157]
SPECIALIST Lexicon	Contains common English words as well as biomedial vocabulary.	yes	FCA-Map (2016) [613] LogMap (2018) [243] *Real et al.* (2020) [428]
SUMO [373]	The suggested upper merged ontology (SUMO), an upper ontology.	yes	*Mascardi et al.* (2010) [334]
Swoogle [103]	A search engine for the Semantic Web. No longer available.	no	SCARLET (2007) [449, 451] *Vazquez & Swoboda* (2007) [545] Spider (2008) [450]
synonyms-fr.com	A Web service to retrieve French synonyms and antonyms.	yes	*Fu et al.* (2011) [159]
UBERON [355, 178]	A cross-species anatomical ontology.	yes	*Groß et al.* (2011) [171] AgreementMaker (2011) [88] GOMMA (2012) [172] AML (2013) [145] LYAM++ (2016) [531] CroMatcher (2016) [177] POMap (2017) [288] Lily (2020) [222]
UMLS [318]	The unified medical language system is a compendium of vocabularies in the biomedical domain.	yes	NLM (2006) [609] AOAS (2007) [610] ASMOV (2007) [235] RiMom (2007) [308] SAMBO (2007) [520] AgreementMaker (2009) [87] LogMap (2011) [248] *Groß et al.* (2011) [171] GOMMA (2012) [172] *Fernández et al.* (2012) [151] AML (2013) [145] *Amin et al.* (2014) [19] LILY (2018) [524] FCA-Map (2018) [79] *OntoEmma* (2018) [558]
Universal Knowledge Core (UKC)	A multilingual lexical resource.	yes	*NuSM* (2017) [30]
WebIsALOD [467, 198]	Web-extracted hypernymy relations provided as an RDF knowledge graph.	yes	ALOD2Vec Matcher (2018) [409]
Webtranslator API	A Java translation API.	yes	AUTOMS (2012) [282] WeSeE Match (2013) [385]
Wikipedia Corpus	Text corpus of the online encyclopedia *Wikipedia*.	yes	CIDER-CL (2013) [169] *Zhang et al.* (2014) [612] *Todorov et al.* (2014) [533] DisMatch (2016) [447] *Li* (2020) [305]

Table 3.6: Knowledge sources and matching systems that use them from (part 3 of 4). Referenced is the first documented usage by the matching system. Systems that did not participate in the OAEI are italicized. Named systems are referred to using their system name.

Knowledge Source	**Source Description**	**Resource Available**	**Used by System**
Wikipedia MediaWiki API	Web API of the online encyclopedia *Wikipedia*.	yes	*BLOOMS* (2010) [231, 232] *SOCOM* (2010) [158] *Fu et al.* (2011) [159] WikiMatch (2012) [197] *OntoEmma* (2018) [558]
Wikisynonyms	Semantic lexicon built from Wikipedia redirects.	yes	*Kolyvakis et al.* (2018) [280] *DeepAlignment* (2018) [279]
Wiktionary	A community-built dictionary; an RDF version [470] is also available.	yes	*Lin & Krizhanovsky* (2011) [314] Wiktionary Matcher (2019) [402]
WordNet [149]	A well-known database of English synsets.	yes	see Tables 3.8 and 3.9
WordsAPI	A Web API for (English) word definitions, multiple word relations, and more.	yes	*Hnatkowska et al.* (2021) [214]
YAGO [515]	A large knowledge base extracted from multiple sources.	yes	*Todorov et al.* (2014) [533]
Yahoo Image Search	A search engine for images on the Web.	yes	*Doulaverakis et al.* (2015) [111]
Yahoo Search	A search engine for the Web.	yes	*Vazquez & Swoboda* (2007) [545]
Yandex Translation API	A translation Web API by the Yandex search engine.	yes	CroLOM (2016) [267] SimCat (2016) [271] *Ibrahim et al.* (2020) [223]

Table 3.7: Knowledge sources and matching systems that use them from (part 4 of 4). Referenced is the first documented usage by the matching system. Systems that did not participate in the OAEI are italicized. Named systems are referred to using their system name.

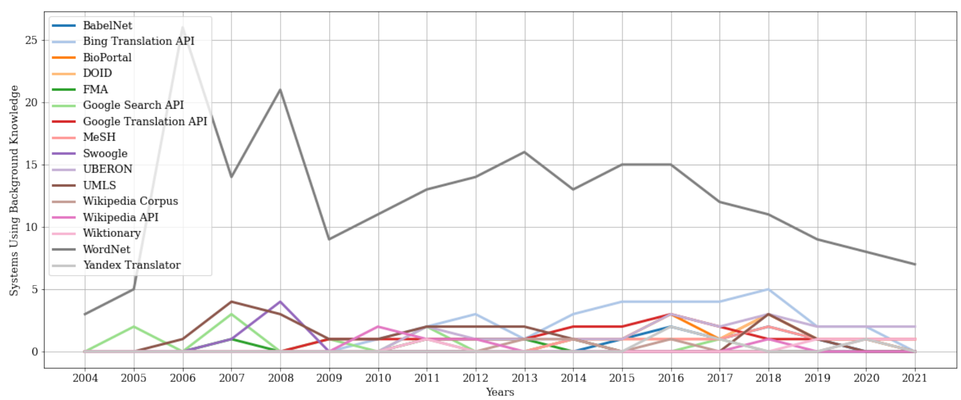


Figure 3.6: Number of publications of this survey using a particular knowledge source over time.

3.4.4 Most Used Background Knowledge Resources

In the following, the ten most used external resources in ontology matching (see Figure 3.3) are shortly introduced.

WordNet WordNet is a database of English words grouped in sets which represent a particular meaning, so-called *synsets*; further semantic relationships,

Knowledge Source	**Used by System**	
WordNet	OLA (2004) [120]	*Cardoso et al.* (2008) [65]
	ASCO (2004) [296]	*Zhang et al.* (2008) [608]
	RiMOM (2004) [523]	*OMIE* (2008) [46]
	MoA$_I$ (2005) [272]	*Fatemi et al.* (2008) [148]
	oMap (2005) [512]	*Wang et al.* (2008) [566]
	CROSI (2005) [258]	*SECCO* (2008) [393]
	Mongiello & Totaro (2005) [350]	*Lera et al.* (2008) [303]
	Aleksovski & Klein (2005) [11]	Agreement Maker (2009) [87]
	OWL-Ctx (2006) [371]	*Eckert et al.* (2009) [114]
	AUTOMS (2006) [283]	*Zhong et al.* (2009) [616]
	DSSim (2006) [359]	*Xia et al.* (2009) [581]
	HMatch (2006) [67]	*Fernández et al.* (2009) [152]
	Aleksovski et al. (2006) [12, 10]	Eff2Match (2010) [83]
	Park et al. (2006) [381, 380]	*Mascardi et al.* (2010) [334]
	Alasoud et al. (2006) [8]	NBJLM (2010) [560]
	Sen et al. (2006) [469]	*ontoMATCH* (2010) [328]
	Reynaud & Safar (2006) [434]	*IROM* (2010) [471]
	Abolhassani et al. (2006) [3]	*Cheatham* (2010) [75]
	Chen et al. (2006) [78]	*Wang et al.* (2010) [565]
	iMapper (2006) [514]	*SOCOM* (2010) [158]
	ontoDNA (2006) [276]	CSA (2011) [536]
	Nagy et al. (2006) [360]	LogMap (2011) [248]
	ACAOM (2006) [564, 563]	MaasMatch (2011) [461]
	Trojahn et al. (2006) [453]	OMReasoner (2011) [475]
	Wang et al. (2006) [570]	Optima (2011) [527]
	Kim et al (2006) [274]	YAM++ (2011) [367]
	Wang et al. (2006) [569]	*Lin & Krizhanovsky* (2011) [314]
	ASMOV (2007) [235]	*Sadaqat et al.* (2011) [234]
	SEMA (2007) [504]	*Thayasivam & Doshi* (2011) [526]
	X-SOM (2007) [92]	*MAMA* (2011) [73]
	iG-Match (2007) [207]	*Vaccari et al.* (2012) [541]
	Tan & Lambrix (2007) [519]	*Liu et al.* (2012) [322]
	Trojahn et al. (2007) [455]	*Acampora et al.* (2012) [5]
	PROMPT-V (2007) [250]	*OARS* (2012) [233]
	Jin et al. (2007) [251]	*Fernández et al.* (2012) [151]
	IAOM (2007) [578]	*FuzzyAlign* (2012) [153]
	Sen et al. (2007) [468]	*OACLAI* (2012) [310]
	UFOme (2007) [394]	*Song et al.* (2012) [502]
	MapPSO (2008) [41]	*Schadd & Roos* (2012) [462]
	Alasoud et al. (2008) [9]	*Gulic et al.* (2013) [176]
	Jeong-Woo et al. (2008) [501]	*MAPSOM* (2013) [252]
	e-CMS (2008) [292]	*Acampora et al.* (2013) [6, 4]
	Kaza & Chen (2008) [262]	AML (2013) [145]
	Trojahn et al. (2008) [456, 454, 457]	XMap (2013) [104]
	Ichise (2008) [224]	SPHeRe (2013) [265]

Table 3.8: Matching systems using WordNet; Part 1 of 2. Referenced is the first documented usage by the matching system. Systems that did not participate in the OAEI at some point in time are italicized. Ontology integration systems are indicated by a subscript I. Named systems are referred to using their system name.

Knowledge Source	Used by System	
	ServOMap (2013) [259]	*Vennesland et al.* (2018) [546, 547]
	Kumar & Harding (2013) [287]	*Refoufi & Benarab* (2018) [430]
	SMILE (2013) [24]	*Kolyvakis et al.* (2018) [279]
	Petrov et al. (2013) [392]	*Bulygin et al.* (2018) [56]
	Lin et al. (2013) [316]	*Kachroudi & Yahia* (2018) [256]
	Fang et al. (2013) [138]	ONTMAT1 (2019) [166]
	UFOM (2014) [611]	Lily (2020) [222]
	Todorov et al. (2014) [533]	*WeGO++* (2019) [427]
	Xue et al. (2014) [595, 597, 598]	*Bulygin & Stupnikov* (2019) [57]
	Jaiboonlue et al. (2014) [230]	*Biniz & Fakir* (2019) [38]
	AOT/AOTL (2014) [268]	*Xue & Chen* (2019) [587]
	InsMT/InsMTL (2014) [269]	*WeGo++* (2019) [427]
	Chaker et al. (2014) [70]	*Yang* (2019) [601]
	Schadd & Roos (2014) [463]	*Ibrahim et al.* (2020) [223]
	ServOMBI (2015) [266]	*Real et al.* (2020) [428]
	DKP-AOM (2015) [134]	*Xue & Chen* (2020) [588]
	Kiren & Shoaib (2015) [275]	*Lv et al.* (2021) [330]
	Nguyen & Conrad (2015) [369]	*Zhu et al.* (2021) [617]
	Wang (2015) [567]	*Xue et al.* (2021) [599]
	Xue et al. (2015) [594, 596, 593, 590, 585]	
	Benaissa et al. (2015) [33]	
	Schadd & Roos (2015) [464]	
WordNet	ALIN (2016) [497]	
	CroLOM (2016) [267]	
	CroMatcher (2016) [177]	
	OMI-DL (2016) [323]	
	Anam et al. (2016) [21]	
	Xie et al. (2016) [582]	
	Mountasser et al. (2016) [352]	
	Idoudi et al. (2016) [225]	
	Xue et al. (2016) [592]	
	ALINSyn (2017) [592]	
	Liu et al. (2016) [321]	
	HSOMap (2016) [156]	
	FCA-Map (2016) [613]	
	KEPLER (2017) [253]	
	ONTMAT (2017) [165]	
	Xue et al. (2017) [586, 591, 589]	
	He et al. (2017) [189]	
	OIM-SIM$_I$ (2017) [607]	
	SANOM (2018) [349]	
	EVOCROS (2018) [99]	
	FCA-MapX (2018) [79]	
	Ochieng & Kyanda (2018) [376]	
	Roussille et al. (2018) [445]	

Table 3.9: Matching systems using WordNet; Part 2 of 2. Referenced is the first documented usage by the matching system. Systems that did not participate in the OAEI at some point in time are italicized. Ontology integration systems are indicated by a subscript I. Named systems are referred to using their system name.

such as *hypernymy*[17] and *hyponymy*[18], also exist in the database. The resource is publicly available.[19] In fact, *WordNet* is so heavily used that there exists a dedicated survey paper titled "A survey of exploiting WordNet in ontology matching" [315]. The resource is under a permissive license and can also be used for commercial purposes.[20]

Bing/Microsoft Translation API The Microsoft Translation API[21], formerly also known as Bing Translation API, allows, among other functions such as language detection, for translating a text string from a source language to a target language. The cloud API can be accessed through any programming language. Since the service is provided in a cloud infrastructure, the translation service is continuously improved. These changes impede the reproducibility of matching systems using the API. The service is not free, but as of 2021, 2 million characters of translation/detection per month are not charged.[22]

UMLS The *Unified Medical Language System* (UMLS) is a manually-built compendium of vocabularies in the biomedical domain. The UMLS is maintained by the *United States National Library of Medicine* (NLM). UMLS can be used without charge, but a download[23] requires a registration at the NLM.

UBERON In the anatomy domain, the *Uber-anatomy ontology* (UBERON) [355, 178] is an ontology for multiple species comprising of more than 13,000 classes (as of 2021). Since UBERON defines a canonical model, it can be used as a "hub ontology" to solve various integration problems in the anatomy domain. The ontology can be used on its own but also in combination with other anatomical ontologies such as the *Foundational Model of Anatomy* (FMA). Particularly the bridging ontologies which connect UBERON to other ontologies (such as UBERON to FMA) make the resource interesting for the task of ontology match-

[17] A *hypernym* or *hyperonym* is a concept which is superordinate to another one. In computer science, it is often represented as an *IS-A* relationship. For example, *animal* is a hypernym of *cat*. [357]

[18] A *hyponym* is a concept which is subordinate to another one. In computer science, it is often represented as an *IS-A* relationship. For example, *cat* is a hyponym of *animal*. [357]

[19] see `https://wordnet.princeton.edu/download`

[20] see `https://wordnet.princeton.edu/license-and-commercial-use`

[21] see `http://www.microsoft.com/translator`

[22] see `https://azure.microsoft.com/en-us/pricing/details/cognitive-services/translator/`

[23] see `https://www.nlm.nih.gov/research/umls/index.html`

ing in this domain. UBERON is publicly available and can be directly downloaded[24] without any registration.

Google Translation API The Google Translation API [25] is very similar to the Microsoft Translation API: It is also a continuously improved cloud service. The Google Translation API is not free, but as of 2021, a translation of 500,000 characters per month is free of charge.[26]

BioPortal The *National Center for Biomedical Ontology* (NCBO) developed and maintains BioPortal[27] [164, 571], a Web repository of interlinked biomedical ontologies. The portal grants access to biomedical ontologies and terminologies developed in various Semantic Web formats. Via *representational state transfer* (REST) services, users can query (among other things) for ontologies, their metadata, and also for individual ontology terms. Registered users can also submit ontology mappings. This allows for community-created integration content. Particularly interesting in the area of ontology matching are the mapping services provided: Mappings can be easily obtained for a term or for a given ontology. The BioPortal services and data can be used free of charge.

DOID The Human Disease Ontology (DO, very often also abbreviated with *DO-ID*) [466] contains, as of 2021, more than 10,800 human diseases which are described through an ontology; its identifiers start with the prefix `DOID`. The resource is built by a community of experts. The disease ontology contains mappings to other vocabularies such as MeSH (see below), ICD[28], or SNOMED-CT[29] concepts. It is publicly available[30] under a very permissive license (CC0).

Google Search API The Google Search API[31] allows for performing Web searches programmatically. Like the Google Translation API, it is not free, but as of 2021, 100 search queries per day are free of charge.

[24] see `http://uberon.org`
[25] see `https://cloud.google.com/translate`
[26] see `https://cloud.google.com/translate/pricing`
[27] see `https://bioportal.bioontology.org/`
[28] ICD stands for “International Classification of Diseases”.
[29] SNOMED-CT stands for “Systematized Nomenclature of Medicine Clinical Terms”.
[30] see `https://disease-ontology.org/`
[31] see `https://developers.google.com/custom-search/v1/overview`

MeSH The *Medical Subject Headings* (MeSH) [513] form the controlled vocabulary thesaurus, which is used to index medical articles. It is built by experts and maintained by the NLM. The data is freely available online for download in multiple formats (including RDF).[32] The dataset is available under a permissive license.

BabelNet BabelNet[33] [362] is a large multilingual knowledge graph that integrates (originally) Wikipedia and WordNet. Later, additional resources such as Wiktionary were added. The integration between the resources is performed in an automated manner. The dataset does not just contain lemma-based knowledge but also instance data (named entities) such as the singer and songwriter *Trent Reznor*. For BabelNet 3.6, an RDF version exists [117]. The dataset can be queried via a *user interface* (UI), SPARQL, and an HTTP API (a Java and a Python client are also available). The dataset is under a restrictive license, and the number of free queries is limited. However, researchers can request access to the indices for non-commercial research projects.

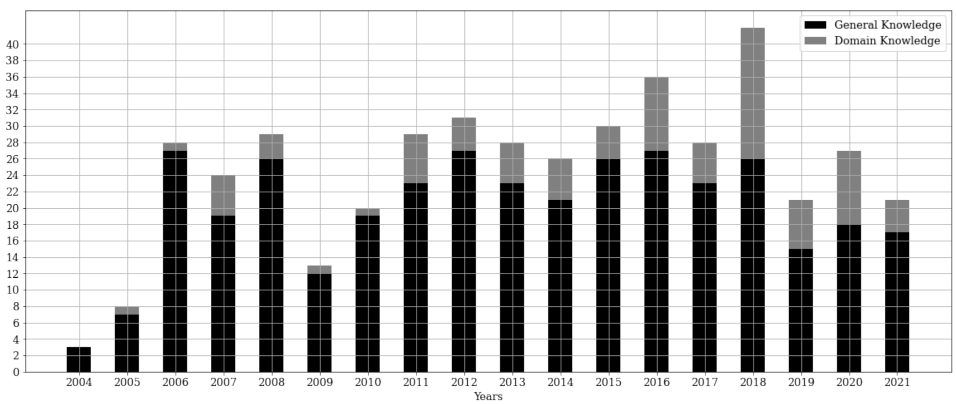


Figure 3.7: Aggregated number of publications of this survey using external background knowledge in ontology matching. Domain-specific background knowledge sources are colored in light gray; general-purpose background knowledge sources are colored in black.

[32]see `https://www.nlm.nih.gov/databases/download/mesh.html`
[33]see `https://babelnet.org/`

3.5 Categorization of Background Knowledge in Ontology Matching

3.5.1 Classification System

Multiple approaches for categorizing general matching techniques have been proposed [426, 478, 129]. The matching techniques further studied in this survey can be broadly categorized as *context-based* approaches according to Euzenat and Shvaiko [129, 478] or as *schema-only based* approaches according to Rahm and Bernstein [426].[34] Rahm et al. do not group background knowledge sources while Euzenat et al. distinguish *formal resources*, i.e.„ those on which reasoning can be applied, and *informal resources*, i.e., those on which reasoning cannot be applied. The latter authors further name the dimensions *breadth*, *formality*, and *status* [132]. In this survey, we propose a more fine-grained categorization with a clear distinction between the background knowledge source that is used and the strategy that is applied to exploit the given knowledge source.

Target Domain Background knowledge sources for matching can be grouped by their *target domain* or *target purpose*. Here, it can be differentiated between *domain-specific* assets and *general-purpose* assets. While general-purpose background knowledge is intended to improve the overall matching quality on any task, domain-specific background knowledge is intended to improve the matching performance within a specific domain or even for a specific matching task. An example of a widely used general-purpose knowledge source is *WordNet*; a case in point for a popular domain-specific knowledge source is the UMLS. The distinction between domain-specific and domain-independent (lexical and grammatical) sources is also made by Real et al. [428] who show in a recent publication that the inclusion of domain-specific lexical- and grammatical knowledge can significantly improve matching systems in domain-specific tasks. In Figure 3.7, the aggregated usage of background knowledge in schema matching systems is plotted per year. It is visible that – up to date – general-purpose knowledge sources are used more often than domain-specific knowledge sources. This finding is intuitive since general-purpose datasets are easier to find, and their application makes sense for any matcher, whereas domain-specific datasets may be harder to find (depending on the matching task) and require a concrete, domain-bound matching problem. It is also visible that the research community

[34] This is naturally not precise. WordNet and other lexical resources, for example, are not classified as formal/informal resource-based but instead as language-based, according to Euzenat and Shvaiko.

initially started with general-purpose background knowledge and explored domain-specific sources at a later stage. Most publications using external background knowledge sources (general and domain-specific) were published in 2018. It is important to note that this survey does not cover the full year of 2021.

Structuredness Independent of the domain, the knowledge sources can be split in *structured sources* and *unstructured sources*. Structured data is organized according to a known data schema, whereas unstructured data is not. An example for a structured external data source in ontology matching is *WordNet*; an example for a general-purpose unstructured data source in ontology matching is the entirety of *Wikipedia* texts, whereas *SAP Term*, a set of definitions of terms in SAP software, is an example of a domain-specific unstructured resource. Unstructured external resources are rarely used in ontology matching. We, therefore, only classify into textual and non-textual unstructured resources whereby we did observe merely one publication [111] using non-textual, unstructured sources (i.e., images).

Structured sources appear in different variations (*type*): (i) Lexical and taxonomic resources, (ii) factual databases, (iii) Semantic Web datasets, and (iv) pre-trained neural models. Lexical and taxonomic resources, as well as pre-trained neural models, can again be subdivided into monolingual and multilingual resources.[35] Semantic Web datasets can be subdivided into single datasets and interlinked datasets.

An overview of the proposed classification system is presented in Figure 3.8; in Table 3.10, all resources covered in this survey are categorized according to the presented classification system. In the following, we will further define each structured resource and provide examples for all fine-grained categories.

[35]Theoretically, the other structured resources can also be mono- or multilingual – however, the focus of the knowledge provided there is rather factual, and the language is typically not the core property of the knowledge resource. Therefore, we decided against a subdivision here in favor of clarity.

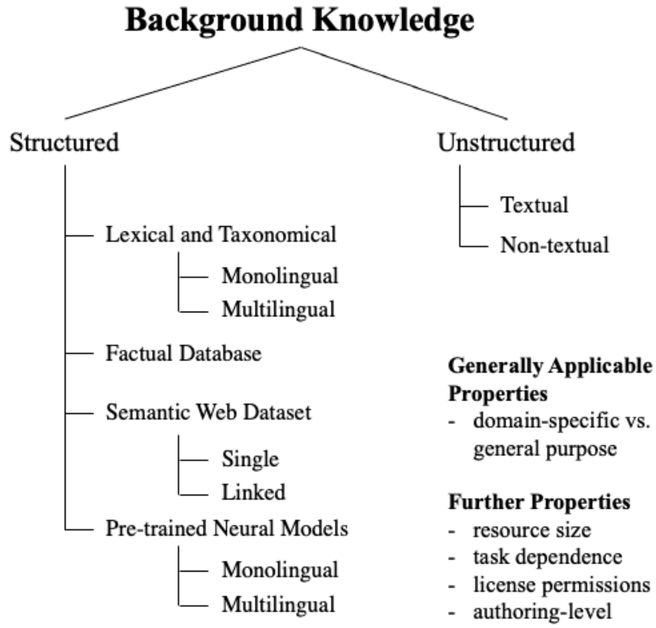


Figure 3.8: Classification of background knowledge sources that are used for matching.

Lexical and Taxonomical Knowledge Lexical and taxonomical knowledge is the most exploited external type of knowledge in ontology matching. The most commonly used resource in this class in our study is *WordNet*. The resource is monolingual; this means it is available in only one language, i.e., English. Similar resources exist in other languages, such as the German thesaurus *GermaNet* [182] – however, since most ontology matching benchmark datasets are provided in English, our study is consequently also skewed towards English resources. Concerning multilingual lexical knowledge, dictionaries and dictionary-like resources, such as APIs, are heavily used for multilingual ontology matching. In our study, we found substantial usage of the *Microsoft Bing Translation* API but also of other general-purpose translation APIs. Although not appearing in the tables, domain-specific multilingual resources exist, for example, the *Fachwörterbuch Versicherungswirtschaft und -recht*[36] [421].

Factual Databases A factual database provides (non-lexical) facts that can be included in the matching process. An example here might be a database of

[36] German book title, translates to *dictionary of insurance and insurance law*.

postal codes and cities. We did not find any significant usage of such a resource despite imaginable use case scenarios. An example for a domain-specific database would be *MEDLINE*, the bibliographic database of the *National Library of Medicine* which is used by the *DisMatch* [447] and *OntoEmma* [558] matching systems.

Semantic Web Dataset A *Semantic Web* (SW) dataset is a knowledge base developed with technologies from the Semantic Web technology stack, such as RDF or OWL files. The category includes knowledge graphs with or without instance data where we define a knowledge graph slightly broader than in its original sense [384] and also count domain-specific graphs. We also consider SPARQL endpoints as SW datasets in this survey, as well as plain ontologies.

We further differentiate between (i) *single* and (ii) *linked* SW datasets. A single dataset is, in this case, an individual knowledge graph or ontology.

An example for a general-purpose single SW dataset would be *DBpedia* [300] (used e.g. by *LDOA* [255]), *WebIsALOD* [467, 198] (used e.g. by *ALOD2Vec Matcher* [409]), or *Wikidata*. An example for a domain-specific single SW dataset would be the *Financial Industry Business Ontology* (FIBO) used for instance in [351].

An example for a domain-specific linked SW dataset in this sense would be some or all *BioPortal* [571] ontologies together with their mappings, while an example for a general-purpose linked SW dataset would be any two linked general-purpose knowledge graphs.

Pre-trained Neural Models A recent development is the application of deep learning in a multitude of applications. A pre-trained neural model in this classification system may be an API exposing latent representations of concepts, such as KGvec2go[37] [404], or a pre-trained model such as the *Google Universal Sentence Encoder*[38] [69, 602] used by *VeeAlign* [227].

3.5.2 Further Relevant Properties

Further properties of background knowledge sources that are not used here for the proposed classification are (i) *resource size*, (ii) *task dependence*, (iii) *license permissions*, and (iv) *authoring level*. Those properties are important in particular when it comes to the strategies that are applied to exploit the background knowledge.

[37] see `http://www.kgvec2go.org/`

[38] see `https://tfhub.dev/google/universal-sentence-encoder-large/`

The resource size may limit the utility provided by the source – a small general knowledge thesaurus, for example, may only be of limited use – but may at the same time limit the exploitation strategy that can be used; the *RDF2vec* [442] embedding approach (a comparatively scalable embedding approach) is very hard to apply to the *BabelNet* (RDF) knowledge graph [117] due to its sheer size. Surprisingly, the most used general-purpose background knowledge resource, WordNet, is relatively small compared to community-built resources such as BabelNet, Wiktionary, or Wikidata.

The task-dependency also limits the options to exploit the source (see Section 3.7). A very specific Web-API providing only a very specific service may limit the strategy to the simple call of the service.

While license permissions are not of utmost concern to the research community, they are very important in the enterprise world when it comes to the actual application of matching systems in the real world for commercial purposes.

The level of authoring or trust of a knowledge source is affecting the exploitation strategy as well. Generally, four main categories can be observed: (1) *expert-built resources* such as WordNet, (2) *community-built resources* such as Wiktionary, (3) *semi-automatically built resources* such as BabelNet, and (4) *automatically built resources* such as WebIsALOD. It can be assumed that the amount of trust decreases from (1) to (4): A deeply reviewed, expert-built dictionary such as WordNet may be used with less caution than a community-built online dictionary like Wiktionary or a heuristically extracted dataset such as WebIsALOD. The quality of the matching results is likely not, in every case, proportional to the level of trust since it depends on the exploitation strategy used and the concrete resource. Automatically-trained neural language models, for instance, have a low authoring level but may produce very good results.

3.6 Categorization of Linking Approaches

In order to exploit an external knowledge source, the concepts in one or both of the ontologies to be matched need to be linked to the knowledge source. The linking process is also known as *anchoring* or *contextualization* [132]. For example, to determine whether the classes `http://mouse.owl#MA_0002390` and `http://human.owl#NCI_C33743` of the OAEI Anatomy track [42] are similar using Wiktionary, the URIs have to be first linked to one or more Wiktionary entries. In this case, the label of the first can be used to link it to the entry of "temporalis" and the label of the latter can be used to link it to the entry of "temporal

Background Knowledge Type				Background Knowledge Source
Domain-specific	**Structured**	**Lexical and Taxonomical**	**Monolingual**	RadLex SPECIALIST Lexicon
			Multilingual	–
		Factual Database		Medline PubMed
		Semantic Web Dataset	**Single**	DOID FMA FIBO Medical Subject Headings (MeSH) UBERON
			Linked	BioPortal Ontology Lookup Service (OLS) UMLS
		Pre-trained Neural Model	**Monolingual**	BioBERT
			Multilingual	–
	Unstructured	**Textual**		Cooking Dictionary SAP Term
		Non-Textual		–
General-Purpose	**Structured**	**Lexical and Taxonomical**	**Monolingual**	Big Huge Thesaurus Paraphrase DB (PPDB) synonyms-fr.com Universal Knowledge Core (UKC) Wikipedia MediaWiki API (non-text serach) Wikisynonyms WordNet WordsAPI
			Multilingual	Apertium Bing/Microsoft Translator Freelang Google Translation API HowNet iTranslate4 Lanes API MyMemory API SDL FreeTranslation Webtranslator API Yandex Translation API
		Factual Database		–
		Semantic Web Dataset	**Single**	BabelNet DBnary DBpedia ConceptNet DOLCE OpenCyc SUMO Swoogle WebIsALOD YAGO
			Linked	–
		Pre-trained Neural Model	**Monolingual**	BERT fastText model Google Word2Vec Vectors KGvec2go SBERT
			Multilingual	Google Universal Sentence Encoder
	Unstructured	**Textual**		Bing Search Engine API FARO Web Search Google Search API Wikipedia Corpus Wikipedia MediaWiki API (for text search) Yahoo Search
		Non-Textual		ImageNet Yahoo Image Search

Table 3.10: Background knowledge sources sorted according to their type.

muscle". Within the knowledge source, we can then find a synonymy relation between the two entries and derive a degree of similarity.

While many publications address the concrete application of a background source for ontology matching, few discuss the actual linking problem. However, since linking is the first step in exploiting a knowledge source, it significantly determines the quality of the outcome. In a visionary paper by Sabou et al. [451], online ontologies obtained with a Semantic Web search engine have been used for ontology matching. Out of the 1,000 correspondences checked manually, 217 false ones have been identified. The authors find that out of those, 53% are due to anchoring errors. This emphasizes the need for a solid anchoring strategy.

The linking process is typically dependent on the knowledge source used and can be as simple as forwarding a label (e.g., when using the Google search API) or as complicated as the ontology matching problem itself (e.g., when another knowledge graph shall be used).

For linking, we distinguish two goals: (i) finding at most one link for each concept in an ontology and (ii) finding up to many links for each concept in an ontology. Multiple links can be sensible in the case of partial linking; for example, a concept with label "derivatives exchange" may be linked to "derivatives" and "exchange" in cases where there is no match for the complete concept. Other reasons for multi-linking are datasets with homonyms[39] or knowledge sources that explicitly provide multiple senses for strings. For the latter two cases, a *Word Sense Disambiguation* (WSD) approach may help to decide on a smaller set of links.

In terms of classifying linking approaches, we propose a classification system consisting of four categories: (i) given links, (ii) direct label linking, (iii) fuzzy linking, (iv) Word Sense Disambiguation. The proposed classification system is summarized in Figure 3.9. In the following, we will introduce each category in detail and provide examples. It is important to note that not every linking strategy can be applied to each dataset; WSD, for instance, can only be applied if there are multiple senses available in the background dataset.

[39] *Homonyms* are words that have the same writing (homographs) or the same pronunciation (homophones) but different senses [325]. An example would be the word "bank" in two different contexts: It may refer to the financial institution in one case and to a seating-accommodation in the other case. To be precise, for the linking problem at hand, only *homographs* are challenging.

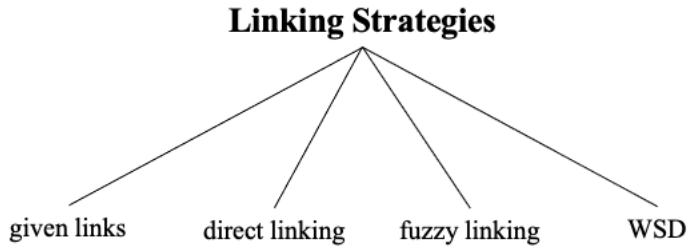


Figure 3.9: Categorization of Linking Approaches

Given Links In few cases, linking can be omitted if the external knowledge source already contains links, e.g., in the form of `owl:sameAs` or `owl:equivalentClass` statements. A case in point is Wikidata where multiple identifiers are typically specified; the concept *pneumonia* (Q12192[40]), for instance, lists more than 30 identifiers for other datasets – among them IDs for *MeSH*, *BabelNet*, the *Disease Ontology*, *Freebase*, or *UMLS*.

Direct Label Linking Given the sparse information provided in publications concerning the linking strategy, it can be assumed that in most cases, linking is performed by directly looking up a potentially normalized label. This works particularly well if the external dataset has a very large coverage of concepts or even provides synonyms such as lexical and large taxonomical background knowledge datasets. Recent matching systems that apply this kind of linking are, for example, *FCA-MapX* [79], *ONTMAT1* [166], or *Wiktionary Matcher* [402, 410].

Fuzzy Linking The linking process can also be based on only parts of a label, n-grams within a label, or expanded labels. Such linking approaches fall under the *fuzzy linking* category. The underlying goal of this strategy is to find more links than through direct label linking. Naturally, this strategy is attractive if the background dataset is small and/or the concepts in it are described by a single label (without stating alternative names, abbreviations, synonyms, etc.). Mascardi et al. [334], for instance, match two ontologies to an upper ontology and then use the obtained two alignments to derive a final alignment; they perform an involved (upper ontology) matching/linking operation including synonymy expansion and substring-based approaches.

[40] see `https://web.archive.org/web/20201113010038/https://www.wikidata.org/wiki/Q12192`

Word Sense Disambiguation (WSD) We did not find matching systems that try to *actually* disambiguate the sense of a label through Word Sense Disambiguation (i.e., which try to settle with *one* correct sense) – despite the heavy usage of WordNet (which is built around senses).[41] Instead, similarity approaches that can handle multiple senses are typically used. The *NBJLM* [560] matching system narrows down the number of WordNet synsets – but only to reduce the computational complexity.

3.7 Categorization of Background Knowledge Exploitation Approaches

In Section 3.5, the background knowledge resources used in ontology matching have been presented and categorized. The second main dimension of this survey is the exploitation strategy of the background resource. In many cases, there are multiple options to beneficially use an external knowledge source.

We classify exploitation strategies into four groups: (i) factual queries, (ii) structure-based approaches, (iii) statistical/neural approaches, and (iv) logic-based approaches. A factual query is a request for one or more data records contained in the background resource. Structure-based approaches exploit structural elements in the background knowledge source. Statistical or neural approaches apply statistics or deep learning to the background knowledge source or consume an existing pre-trained model. Lastly, logic-based approaches employ reasoning with the externally provided resource. In the following, the categories are further described, and extensive examples are provided. An overview of the proposed classification system is provided in Figure 3.10.

[41] Some authors consider WordNet metrics such as the *Resnik word similarity* [433] or *Wu-Palmer* [580] as WSD (e.g., [37]) – however, we regard averaging synset similarity scores or picking the maximum score across multiple synset comparisons not as *real* Word Sense Disambiguation; the obtained similarity through such approaches is a *word* similarity rather than a disambiguated *sense* similarity.

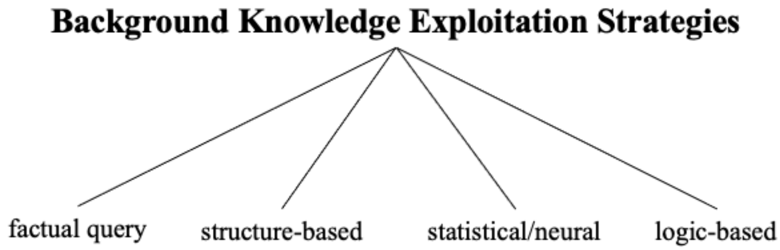


Figure 3.10: Overview of the types of background knowledge exploitation strategies.

Factual Queries A factual query is the extraction of an existing record from the knowledge source. This type of exploitation strategy is the most common one and used since the early days of (semi-) automated ontology matching. An example for retrieving factual information would be retrieving synonyms from *WordNet* (applied by many matching systems e.g. *RiMom* [307], *AgreementMaker* [87], or *FCA-Map* [79]) or from *DBnary* [470] (e.g. by *Wiktionary Matcher* [402, 410]).

Structure-based Approaches Structure-based methods require a structural dimension in the background resource, such as a tree or graph structure. Elements to be compared are typically projected into the background source, and the structure is used to derive a new fact between the projected elements, such as equivalence or subsumption. Structure-based approaches are often applied on WordNet to determine similarity, such as the path-based approaches by Wu and Palmer [580] or Jian and Conrath [237] (both used, for example, by the *YAM++* matching system [367]) or the information-based approach proposed by Lin [313] (used for example by the *RiMom* [522] matching system).[42] Many more *WordNet*-based approaches that fall into the structure-based category of this survey have been proposed and used in ontology matching; we direct the interested reader to the survey by Lin et al. [315]. Structure-based approaches have not only been used together with WordNet but have also been applied on other datasets such as overlap-based metrics based on *WebIsALOD* [417]. A structural approach on Wikipedia categories is applied by *BLOOMS* [231] where concepts are linked into the Wikipedia taxonomy, and an overlap measure on taxonomy sub-trees is defined to determine similarity. Given a repository of ontologies

[42] There is in some cases no clear boundary between structure-based and statistical approaches since structure-based approaches typically apply statistics. We classify an approach to be structure-based if the focus is the exploitation of the structure of the knowledge source.

together with correspondences, Annane et al. [22] apply a structure-based strategy, where they first form a so-called *global mapping graph*. Source and target ontology are linked to the latter, and a path-based strategy is applied so that the correspondences with the highest confidence can be extracted.

Due to their nature, structure-based approaches are not (obviously) applicable to factual databases or pre-trained neural models.

Statistical/Neural Approaches Statistical approaches apply a statistical process to the data derived from the external knowledge source. The *WeSeE-Match* system [383, 385], for instance, builds virtual documents from search engine results and derives a similarity estimate by applying a strategy that is based on the TF-IDF vectors of the documents.

Neural approaches employ artificial neural networks either directly on the background knowledge source or re-use existing pre-trained models. For example, the background knowledge source may be transformed into a vector space [409] or the background knowledge source is already a vector space that may be used directly to link the schemas to be matched [227] in a vector space. We also count neural APIs into this category; *ALOD2Vec Matcher* [403], for example, uses in its most recent version the API of *KGvec2go* [404] to obtain vectors for concepts. While this could be seen as a factual query, we still consider this strategy to be a neural one due to the nature of the approach. It is important to note that we focus only on strategies applied to the background knowledge – a matching system that uses neural networks to configure weights of various features (e.g., the 2011 version of *CIDER* [170]) does not fall in this category, and neither does a matching system that applies a neural model to the ontologies that are to be matched such as *DOME* [200]; the reason for this decision is that the latter two system types do not actually use external background knowledge for their matching strategy. Systems that apply statistical approaches are not novel – however, systems that apply neural methods are relatively recent (the oldest ones of this survey being from 2018, e.g., [409]), not plentiful in numbers, and achieve mixed results. This is most likely due to the novelty of this exploitation strategy. Notable in this category is the *VeeAlign* [227] matching system, which uses a sentence encoder as external knowledge and achieved the best results on the *Conference* [77] track in the OAEI 2020.

Logic-based Approaches Logic-based approaches apply reasoning on or together with the external resources. This class of approach is also referred to as *context-based matching* [326] or *indirect matching*[43]. Typical external resources are upper ontologies, domain-ontologies, knowledge graphs, or linked data. We differentiate reasoning from the *factual queries* in that a reasoning operation goes beyond querying a graph with an `ASK` query for equivalence or any other relation between two concepts. Logic-based approaches are already envisioned in the earlier days of ontology matching. An archetypal setup of such an approach is presented in Figure 3.11 which was first presented by Sabou et al. [449] and slightly adapted for this survey: Elements of the ontologies to be matched are linked to the external ontology (Sabou et al. call this step *anchoring*, Euzenat et al. refer to this step as *contextualization*, see Section 3.6) and reasoning is applied to derive correspondences. It is important to note that reasoning can also be applied across multiple ontologies: Locoro et al. [326] generalize and significantly extend the approach by Sabou et al.; they perform reasoning also across more than one intermediate ontology. Their proposed generalized framework consisting of seven logical steps[44] is particularly applicable for logic-based approaches. However, we did not find broad usage of logic-based exploitation approaches in past and current (OAEI and non-OAEI) ontology matching systems that go beyond singled out experiments. Approaches that fall into this category are Sabou et al., who use *Swoogle* to retrieve ontologies from the Web. *BLOOMS+* [232] does not strictly reason on the external resource but applies a context similarity measure based on the overlap of superclasses that could be seen as such. *Mascardi et al.* [334] perform experiments on multiple upper ontologies (*DOLCE* [161], *SUMO* [373], *OpenCyc* [302])[45] following a similar approach of exploiting the transitivity of equivalence relations. Strictly speaking, Mascardi et al. are also not performing a *real* reasoning operation as defined at the beginning of this paragraph. Despite the clear vision of the latter two publications, upper ontology approaches that exploit actual reasoning have not gained traction so far.

[43] The term *indirect matching* may also refer to structure-based approaches such as the works by Annane et al. [23, 22]. This is due to the fact that in this survey, we differentiate between structure-based approaches (such as a path-based algorithm) and logic-based approaches – a distinction that other authors do not make.

[44] The steps are namely: (i) ontology arrangement, (ii) contextualization, (iii) ontology selection, (iv) local inference, (v) global inference, (vi) composition, and (vii) aggregation.

[45] SUMO stands for "suggested upper merge ontology", DOLCE stands for "descriptive ontology for linguistic and cognitive engineering", and OpenCyc is a subset of the Cyc knowledge base by Cycorp that is not available anymore.

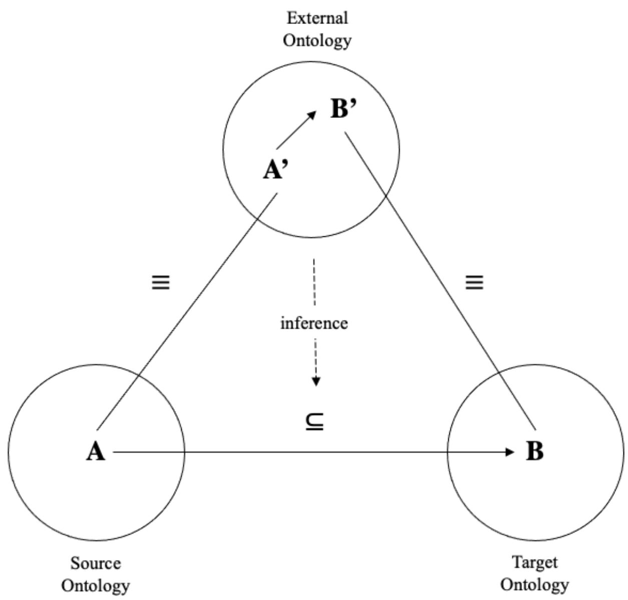


Figure 3.11: A logic-based exploitation strategy on an external ontology, initially presented by Sabou et al. [449], adapted. A and B represent concepts from the ontologies to be matched that are linked to $A^{\prime}$ and $B^{\prime}$ in the external ontology.

3.8 Directions for Future Work

In Section 3.5, we proposed a classification system for background knowledge sources, and in Section ,3.7 we presented a classification system for exploitation approaches. In this section, we will overlap those to a matrix and will position the systems evaluated in this survey there. We will use this matrix as a starting point for discussions of white spots in the area of background knowledge-based ontology matching. We further outline interesting observations, shortfalls, and biases found in the ontology matching domain.

3.8.1 White Spots

Tables 3.11 (domain knowledge) and 3.12 (general knowledge) present the systems evaluated in this study in a source/strategy matrix. The exploitation strategy (columns) in the table follows the proposed classification, which is summarized in Figure 3.10. The rows represent the background knowledge type and follow the proposed classification, which is summarized in Figure 3.8. Irrelevant combinations of source and strategy are grayed out in the tables. Empty or rarely

Background Knowledge (Domain-Specific)				Strategy: Factual Queries	Strategy: Structure-based	Strategy: Logic-based	Strategy: Statistical/Neural
Domain-Specific	**Structured**	**Lexical and Taxonomical**	**Monolingual**	*Groß et al.* (2011) [171] AML (2014) [141] FCA-Map (2016) [613] *Ochieng & Kyanda* (2018) [376] LogMap (2018) [243] *Real et al.* (2020) [428]		–	*Fang et al.* (2013) [138]
			Multilingual			–	
		Factual Database			–	–	DisMatch (2016) [447] *OntoEmma* (2018) [558] *Li* (2020) [305]
		Semantic Web Dataset	**Single**	AOAS (2007) [610] GOMMA (2012) [172] AML (2014) [141] LAYM++ (2016) [531] CroMatcher (2016) [177] POMap (2017) [288] *Ochieng & Kyanda* (2018) [376] Lily (2020) [222]	*Petrov et al.* (2013) [392] *Annane et al.* (2018) [23]		DESKMatcher (2020) [351]
			Linked	NLM (2006) [609] AOAS (2007) [610] ASMOV (2007) [235] SAMBO (2007) [520] LogMap (2011) [248] *Fernández et al.* (2012) [151] GOMMA (2012) [172] AML (2013) [145] *Amin et al.* (2014) [19] LogMapBio (2014) [246] *Zaveri & Dumontier* (2016) [605] Lily (2018) [524] FCA-Map (2018) [79] *PAXO* (2020) [186]	*Petrov et al.* (2013) [392] *Annane et al.* (2016) [22] *Annane et al.* (2018) [23]		RiMom (2007) [308] *OntoEmma* (2018) [558]
		Pre-trained Neural Models	**Monolingual**	–	–		*MEDTO* (2021) [184]
			Multilingual	–	–		
	Unstructured	**Textual**		–	–	–	*van Hage et al.* (2005) [179] DESKMatcher (2020) [351]
		Non-Textual		–	–	–	

Table 3.11: Systems in the background knowledge type / exploitation method type matrix (domain-specific background knowledge).

Background Knowledge (General Purpose)				Strategy: Factual Queries	Structure-based	Logic-based	Statistical/Neural
General-Purpose	**Structured**	**Lexical and Taxonomical**	**Monolingual**	HotMatch (2012) [94] *SOCOM++* (2012) [160] [many WordNet systems, see Tables 3.8 and 3.9] *Mao et al.* (2011) [333] *Hnatkowska et al.* (2021) [214]	*BLOOMS* (2010) [231] [many WordNet systems, see Tables 3.8 and 3.9]	–	*DeepAlignment* (2018) [279]
			Multilingual	*Li et al.* (2006) [306] *Wang et al.* (2008) [566] *SOCOM* (2010) [157, 158] / *SOCOM++* (2012) [160] *Spohr et al.* (2011) [505] *Fu et al.* (2011) [159] AUTOMS (2012) [282] WeSeE Match (2012) [383] YAM++ (2012) [365] Medley (2012) [188] RiMom (2013) [615] *Koukourikos et al.* (2013) [284] GOMMA (2012) [172] AML (2014) [141] *Kachroudi et al.* (2014) [257] XMap (2014) [105] LogMap (2015) [245] CLONA (2015) [2] CroLOM (2016) [267] SimCat (2016) [271] *Helou et al.* (2016) [192] *Destro et al.* (2017) [98] KEPLER (2017) [253] *Bella et al.* (2017) [30] *NuSM* (2017) [30] *Kachroudi & Yahia* (2018) [256] Wiktionary Matcher (2019) [402] *Ibrahim et al.* (2020) [223]		–	*Kolyvakis et al.* (2018) [280]
		Factual Database			–	–	
		Semantic Web Dataset	**Single**	*Vazquez & Swoboda* (2007) [545] Spider (2008) [450] LDOA (2011) [255] *Davarpanah et al.* (2015) [95] EVOCROS (2018) [99]	*BLOOMS* (2010) [231] *Grütze et al.* (2012) [175] *Todorov et al.* (2014) [533]	SCARLET (2007) [449, 451]	LYAM++ (2015) [530] ALOD2Vec (2018) [417, 409] *Kolyvakis et al.* (2018) [280]
			Linked				
		Pre-trained Neural Models	**Monolingual**	–	–		*Bulygin* (2018) [56] *Bulygin & Stupnikov* (2019) [57] VeeAlign (2020) [227] ALOD2Vec (2020) [403] KGMatcher (2021) [137] Fine-TOM (2021) [278] TOM (2021) [281] *MEDTO* (2021) [184] *Neutel et al.* (2021) [363]
			Multilingual	–	–		VeeAlign (2020) [227]
	Unstructured	**Textual**		–	–	–	*van Hage et al.* (2005) [179] *Pan et al.* (2005) [379] *Vazquez & Swoboda* (2007) [545] *Gligorov et al.* (2007) [168] *PROMPT-V* (2007) [250] X-SOM (2007) [92] *KMSS* (2009) [606] *Mao et al.* (2011) [333] WeSeE Match (2012) [383] WikiMatch (2012) [197] CIDER-CL (2013) [169] MapSSS (2013) [76] DisMatch (2016) [447] OntoConnect (2020) [71] *Neutel et al.* (2021) [363]
		Non-Textual		–	–	–	*Doulaverakis et al.* (2015) [111]

Table 3.12: Systems in the background knowledge type / exploitation method type matrix (general-purpose background knowledge).

filled white cells hint at yet underexplored and potentially interesting research directions in the area of background knowledge-based ontology matching.

From the tables, we see that general-purpose background knowledge is used more often than domain-specific background knowledge.[46] The most often used background knowledge types are lexical and taxonomical resources, with WordNet being the clear winner. Clearly not often used are unstructured, non-textual data, pre-trained neural models, and general-purpose Semantic Web datasets.[47] It is important to note that the heavy usage of linked data in Table 3.11 is mainly due to UMLS falling in that category – almost all systems listed use this single resource. Hence, the general application of linked data is not yet common, too. Interestingly, the application of general-purpose textual data has been explored in multiple publications, whereas there is merely a single application of domain-specific free text.

It is quickly visible that factual queries are most often used regarding the strategy. When it comes to yet underexplored research directions of background knowledge usage, we see that in terms of the approaches used, logic-based and neural-based strategies are interesting and promising research directions. Pre-trained embedding models and architectures, for instance, are up to 2020 rarely used but may be very promising given breakthroughs in other scientific communities. An increase in publications in 2021 in this category may indicate that scientific interest is already moving in this direction. Structural approaches are almost completely limited to the English WordNet. The exploration of structural methods on multilingual datasets as well as on Semantic Web datasets may yield interesting results given good results on the English WordNet and given that this class of approaches is typically intuitive to understand and can be comprehended by humans (unlike neural models).

3.8.2 It's a Biomedical World

If we take a closer look at the domain-specific knowledge sources used, it is striking that almost all datasets are from the biomedical domain. This may be due to a particularly prolific bioinformatics community that holds open standards and open data high – however, the skewness of ontology matching publications towards the biomedical domain must be pointed out. In Figure 3.3 (cumulative background knowledge usage), it is striking that all domain-specific datasets are from the biomedical domain. This domain focus is also visible when looking at

[46] Note that systems that use WordNet (see Tables 3.8 and 3.9) are not explicitly listed for better clarity in Table 3.12.

[47] The low usage of factual databases may be due to the fact that the community prefers knowledge presented in a graph.

OAEI tracks, where almost all domain-specific problems are from this domain. This fact is likely self-enforcing: New researchers use existing evaluation datasets and existing background knowledge and quickly find themselves in this domain area.

Nonetheless, ontology matching is a problem in all domains that are concerned with data management which makes it ubiquitous. Enterprise schema matching and integration challenges in the business world, for example, are not reflected at all in OAEI tracks.[48] In addition, there are indications that top-performing OAEI schema matching systems perform comparatively badly on real-world business integration tasks [401]. More insights on the generalization of current matching methods, properties of matching problems in other domains, or further well-performing domain-specific or general-purpose datasets are desirable.

An interesting research direction is, therefore, also to broaden the domain-focus of the ontology matching problem and to evaluate which background datasets and exploitation strategies are applicable in other domains. Therefore, new and publicly available benchmark datasets from more domains are required to support research efforts in this area. New challenges may come to light, such as missing domain-specific knowledge sources not being broadly available [396]. The provisioning of further evaluation datasets in other domains is a clear desideratum.

3.8.3 Multilinguality

A further bias besides a domain focus is the focus on monolingual ontology matching. At the OAEI, there is currently only one multilingual matching task with few participants. The techniques currently applied are purely lookup-based despite advances in machine translation.

Multilingual ontology matching requires the addition of external resources; hence, we can find many multilingual background sources in Tables 3.4 to 3.7. However, when we compare the resource/strategy matrix in Tables 3.11 and 3.12, we quickly see that there are many systems that use general-purpose multilingual resources, but there is not a single system that uses domain-specific multilingual resources. This may be due to the fact that there are, at the moment, no benchmark datasets for more advanced multilingual matching tasks available –

[48] In the years 2016 and 2017, there was a *Process Model Matching Track* at the OAEI. While the topic of process model matching is relevant for the industry, the dataset was limited to the domains of university admissions in 2016 and additionally birth registrations in 2017. At the OAEI, the overall participation in the track was rather low, with only four systems in two years: AML [142, 139], DKP [135], LogMap [244, 242], and I-Match [270].

despite this being a relevant problem in the real world. The current multilingual evaluation datasets are all from the conference domain with a rather low level of domain complexity.

It could be further observed that, although many diverse, multilingual resources such as Wikidata or EuroVoc[49] exist, most multi-lingual matchers use translation APIs with a simple factual query strategy. This setup limits reproducibility and transparency.

Interesting research directions are the exploration of new multilingual matching methods and datasets as well as the exploration of multilingual matching challenges in domain-specific settings. The provisioning of further evaluation datasets is also for the aspect of multilingualism a desideratum. Given well-performing and publicly available deep-learning models from the *natural language processing* (NLP) domain, their application should also be considered for the ontology matching task.

3.8.4 The English Bias

Another language-based bias is the focus on aligning schemas that are semantically described in the English language. The research community currently mainly solves English-English alignment problems.[50] This bias can already be seen when reviewing the most common evaluation datasets – but this bias is also found in the background knowledge used: The majority of background knowledge sources listed in Tables 3.4 to 3.7 are available in English as *main* language (with the exception of some translation-oriented datasets such as translation APIs). It is unlikely that this setting reflects the real-world situation.

An interesting research direction is, therefore, the exploration of non-English rooted ontology matching problems with non-English background knowledge sources. As with the multilingual bias, the community would greatly benefit from the provisioning of more evaluation datasets.

3.8.5 Manual Background Knowledge Selection

While multiple automatic background knowledge selection approaches have been proposed (see Subsection 3.4.2), we did not find significant usage of documented automated selection processes in the publications reviewed for this survey. Up

[49] EuroVoc is a multilingual thesaurus by the Publications Office of the European Union. See `https://op.europa.eu/en/web/eu-vocabularies`

[50] It has to be mentioned here that this survey only considers publications published in English (see C1 in Table 3.2) which may skew the observations. However, given that English is the lingua franca in the ontology matching community, we assume that this skew is small.

to date, the majority of background knowledge sources in ontology matching are either bound to one predefined source or use a few hand-picked resources. With the exception of LogMapBio, most matching systems which apply an automated selection approach are presented in the context of background knowledge selection. Hence, self-configuring matching systems that select their own background resources based on a particular matching problem are still an interesting area of research. Very recent approaches, such as the usage of pre-trained language models that are fine-tuned on the matching task, do not solve this task (but instead emphasize the importance since the pre-trained model also needs to be selected).

3.8.6 Linking

Our analysis of how concepts are linked into the background knowledge source revealed that most matching systems do not perform elaborated linking techniques but use a direct string lookup. While this may be sufficient for some background datasets, there is indication that in some cases, linking is a significant component in the performance of background knowledge-based matching systems [451, 450].

A reason for the negligence when it comes to linking might be that Word Sense Disambiguation is perceived as too hard. Another reason might be due to the fact that schemas to be integrated are often derived from the same domain, which significantly reduces the amount of *concept and definiens* and *concept* mismatches [548] induced by homonyms since words will often refer to the same senses. For example, when two ontologies from the financial services domain use the term "bank", they likely both refer to the sense of a financial institution – an elaborated WSD approach would not provide any value here. Existing evaluation datasets are all more or less from the same domain and do not reflect this problem appropriately.

However, when large external knowledge bases are to be matched or when the schemas to be matched are large and diverse such as in the case of knowledge graph matching, WSD may significantly improve the results obtained with external background knowledge. This finding is in line with a recent publication on knowledge graph matching by Hertling and Paulheim [202] who show that state-of-the-art matching systems perform badly when it comes to matching non-related or weakly-related knowledge graphs due to non-disambiguated homonyms.

An interesting research direction is consequently the development, evaluation, and comparison of multiple linking approaches and their effect on the performance of automated matching systems. We also see a need for the provisioning of additional matching gold standards in the area of knowledge graph matching as well as matching of weakly related schemas.

3.9 Conclusion

Since the early 2000s, the understanding of the (automated) ontology matching problem, as well as the development of advanced matching systems, have greatly improved. Nonetheless, the ontology matching problem is not solved and will stay an interesting research area for the years to come. One key to coming closer to the solution is the deeper integration of background knowledge within the ontology matching process.

In this survey, we reviewed all ontology matching systems that participated in the OAEI from 2004 until today, as well as systematically selected ontology matching systems in terms of what background knowledge sources they use, which linking approach they employ, and how they use the external knowledge. We classify background knowledge resources in multiple structured and unstructured classes according to their purpose (domain-specific or general-purpose). The main structured knowledge source types are (i) lexical and taxonomical resources, (ii) factual databases, (iii) Semantic Web datasets, and (iv) pre-trained neural models. The main unstructured resource types are (i) textual and (ii) non-textual. In our review, we found that mostly general-purpose structured knowledge is used in ontology matching. Most systems to date make use of simple lexical and taxonomical sources. Yet underexplored sources of background knowledge are unstructured resources, pre-trained neural models, general-purpose knowledge graphs, and linked data.

We further presented a classification system for linking strategies consisting of four categories: (i) given links, (ii) direct linking, (iii) fuzzy linking, and (iv) Word Sense Disambiguation. Although linking is important when it comes to exploiting external knowledge sources, we found that most systems use direct label linking.

Concerning the strategy that is used to exploit knowledge sources, we presented a classification system consisting of four categories: (i) factual queries, (ii) structure-based approaches, (iii) logic-based approaches, and (iv) statistical/neural approaches. We found that a look-up strategy of facts is most commonly used. Structure-based strategies are almost exclusively applied on WordNet. Despite a clear vision, logic-based approaches did not gain much traction in recent years. A novel research area in terms of exploitation strategies is neural approaches which are currently barely used but showed very good results in other domains.

In our survey, we found multiple biases when it comes to ontology matching with background knowledge: (i) A focus on biomedical matching tasks, (ii) a focus on monolingual matching, and (iii) a focus on matching schemas rooted in the English language. In particular, the business world where integration problems are plentiful and multi-faceted is hardly considered in current research efforts. Although the focus of this survey is the usage of external knowledge within the ontology matching process, we consider the identified biases to be generally applicable.

Part II

A Framework for Knowledge Graph Matching

In this dissertation, new approaches to matching with general-purpose background knowledge sources are presented. Both, novel background knowledge sources as well as novel background knowledge exploitation strategies are developed, analyzed, and evaluated. Hence, a general architecture is required – mainly for two aspects: (1) Evaluation and (2) matcher development.

Existing frameworks lacked detailed evaluation capabilities and novel requirements, such as the independence of a concrete programming language. In order to perform extensive evaluations required for this dissertation, such as comparisons of matching systems down to the level of correspondences, ablation studies, and significance tests, the *Matching EvaLuation Toolkit* (MELT) was developed.

The framework provides simple, programming language-independent, APIs to develop matching modules. Over the course of this dissertation, MELT was gradually extended so that all main matching contributions are available to the research community. Since 2020, MELT has been officially endorsed by the OAEI. Over the short time frame of this dissertation, MELT has already experienced significant third-party usage. The MELT Dashboard allows for exploring alignments in an interactive way and is also used at the OAEI. The *machine learning* (ML) extension provides powerful tools for supervised ML in ontology matching.

A key point of this dissertation is the focus on code reusability and value creation for the research community beyond the reporting of novel and interesting results. Therefore, all matching components presented are included in MELT; this includes components that are not covered in this part of the thesis. As of today, MELT features more than 50 matching components[51] and more than 30 filters[52]. It is important to note that MELT is complementary to existing frameworks. It enables researchers, for instance, to combine a SEALS matching component with a MELT matching component or to run significance tests for SEALS packages.

Besides development and evaluation capabilities, MELT also provides matcher fine-tuning and packaging modules (which support the existing platforms SEALS and HOBBIT). The framework further includes a programming language independent matcher format ("Web Interface")[53] together with an evaluation client[54] for SEALS, HOBBIT, and Web Interface Docker packages.

In the following chapter, the main framework is presented. Chapter 5 addresses evaluations for non-technical users via a Web UI. In Chapter 6, dedicated

[51] see `https://dwslab.github.io/melt/matcher-components/full-matcher-list`

[52] see `https://dwslab.github.io/melt/matcher-components/full-filter-list`

[53] see `https://dwslab.github.io/melt/matcher-packaging/web`

[54] `https://dwslab.github.io/melt/matcher-evaluation/client`

machine learning components are presented. Albeit not included in this part of the dissertation, Chapters 16 and 18 also make significant contributions to the MELT framework in the area of background knowledge-based ontology and knowledge graph matching: In Chapter 16, multiple transformer extensions for MELT are presented; for the evaluations performed in Chapter 18, out-of-the-box support for multiple external knowledge resources was added to the framework.

Lastly, it is important to emphasize that technology is not an end in itself. Therefore, this part is not solely focused on the contribution of new concepts and software artifacts – but also contains an extensive set of novel matchers, evaluations, and analyses.

Chapter 4

Matching EvaLuation Toolkit

In this chapter, we present the main component of the *Matching EvaLuation Toolkit* (MELT), a software toolkit to facilitate ontology matcher development, configuration, evaluation, and packaging. Compared to existing tools in the ontology matching domain, our framework offers detailed evaluation capabilities on the correspondence level of alignments as well as extensive group evaluation possibilities. A particular focus is put on a streamlined development and evaluation process along with ease of use for matcher developers and evaluators. Our contributions are twofold: We present an open-source matching toolkit that integrates well into existing platforms, as well as an exemplary analysis of two OAEI 2018 tracks demonstrating the advantages and analytical capabilities of MELT.

The work presented in this short chapter has been published before as: Hertling, Sven; Portisch, Jan; Paulheim, Heiko. MELT - Matching Evaluation Toolkit. In: Lecture Notes in Computer Science Semantic Systems - The Power of AI and Knowledge Graphs. 15th International Conference, SEMANTiCS 2019. Karlsruhe, Germany. September 9–12, 2019. [203]

4.1 Introduction

Ontology matching or ontology alignment is the non-trivial task of finding correspondences between entities of a set of given ontologies [128]. The matching can be performed manually or through the use of an automated matching system. For systematically evaluating the quality of such matchers, the *Ontology Alignment Evaluation Initiative* (OAEI) has been running campaigns [121] every year since 2005. Unlike other evaluation campaigns where researchers submit *datasets* as solutions to report their results (such as Kaggle[1]), the OAEI requires participants to submit a matching *system*, which is then executed on-site. After the evaluation, the results are publicly reported[2]. Therefore, execution and evaluation platforms have been developed, and OAEI participants are required to package and submit their matching system for the corresponding platform. Two well-known platforms are used in the ontology matching community: The *Semantic Evaluation At Large Scale* (SEALS)[3] [162, 577] and the more recent *Holistic Benchmarking of Big Linked Data* (HOBBIT)[4] [368].

Based on the results of the OAEI 2018 campaign [14], only 4 out of 12 tracks were available in HOBBIT (*LargeBio*, *Link Discovery*, *SPIMBENCH*, *KnowledgeGraph*). Out of 19 matchers that were submitted in the 2018 campaign, only 6 matchers supported both, SEALS and HOBBIT, and 2 supported HOBBIT exclusively. The remaining 11 matchers supported only SEALS. While one reason for the low HOBBIT adoption might be its novelty, it also requires more steps to package a matcher for the HOBBIT platform and knowledge of the Docker[5] virtualization software. In particular, for new entrants to the ontology matching community, the existing tooling might appear overwhelmingly complicated. In addition to potential obstacles for matcher development and submission, another observation from the OAEI campaigns is that the evaluation varies greatly among the different tracks that are offered e.g., *Anatomy* results contain *Recall+* as well as alignment coherence, whereas the *Conference* track focuses on different reference alignments. Due to limited group evaluation capabilities in existing frameworks, some track organizers even developed their own evaluation systems.

For these reasons we present the *Matching EvaLuation Toolkit* (MELT)[6] – an open-source toolkit for ontology matcher development, fine-tuning, submission, and evaluation. The target audience is matching system developers as well as researchers who run evaluations on multiple matching systems such as OAEI track organizers. Likewise, system developers can use this tool to analyze the performance and errors of their systems in order to improve it. Furthermore, they can package and submit the system easily to OAEI campaigns.

The rest of this chapter is structured as follows: Section 4.2 describes other work in the field of alignment visualization and evaluation. Section 4.3 gives an

[1] `https://www.kaggle.com`
[2] `http://oaei.ontologymatching.org/2018/results/index.html`
[3] `http://www.seals-project.eu`
[4] `http://project-hobbit.eu`
[5] `https://www.docker.com`
[6] `https://github.com/dwslab/melt`

overview of the MELT framework and its possibilities, whereas Section 4.4 shows an exemplary analysis of the latest systems submitted to the OAEI. We finish with an outlook on future developments.

4.2 Related Work

As MELT can be used both for evaluating ontology matching tools as well as visualizing matching results, we discuss related works in both fields.

4.2.1 Matching and Alignment Evaluation Platforms

OAEI campaigns consist of multiple problem sets, so-called *tracks*. Each track has its organizers who provide the datasets, including reference alignments, execute the matching systems, and prepare the results page for the participants and the whole community. The track contains one or more test cases that correspond to a specific matching task consisting of two ontologies and a reference alignment. In 2010, three tracks (*Benchmark*, *Anatomy*, and *Conference*) were adjusted to be run with the SEALS platform [119]. One year later, participants of OAEI campaigns had to implement a matching interface, and the SEALS client was the main tool used for executing and evaluating matchers. The interface contains a simple method (`align()`), which receives a *Uniform Resource Locator* (URL) for the source and a URL for the target ontology and has to return a URL that points to a file containing all correspondences in the alignment format[7]. This format is defined and used by the *Alignment API* [96].

Starting in 2017, a second evaluation platform, called *HOBBIT*, was added [249]. One difference compared to SEALS is that the system has to be submitted as a Docker image to a *GitLab* instance[8], and in the corresponding project, a matcher description file has to be created. After submission of the matching system, the whole evaluation runs on servers of the HOBBIT platform. Thus, the source code for evaluating the matchers has to be submitted as a Docker image as well. All Docker containers communicate with each other over a message broker (*RabbitMQ*[9]). Hence, the interface between a system and the evaluation component can be arbitrary. To keep a similar interface to SEALS, the data generation component transfers two ontologies, and the system adapter receives the URL to these files. It should return a file similar to the SEALS interface.

---

[7] `http://alignapi.gforge.inria.fr/format.html`

[8] `https://master.project-hobbit.eu`

[9] `https://www.rabbitmq.com`

Working with alignments in Java code can be achieved with the *Alignment API* [96]. It is the most well-known API for ontology matching and can be used for loading and persisting alignments as well as for evaluating them with a set of possible evaluation strategies. Moreover, it provides some matching systems which are also used in OAEI campaigns as a baseline. Unfortunately, it is not yet enabled to be used with the maven build system[10]. Therefore, instead of using this API, some system developers created their own classes to work with alignments and to store them on disk[11] in order to be compatible with the evaluation interface.

Alignment Visualization A lot of work has been done in the area of analyzing, editing, and visualizing alignments or ontologies with a graphical user interface. One example is *Alignment Cubes* [226], which allows an interactive visual exploration and evaluation of alignments. An advantage is a fine-grained analysis on the level of an individual correspondence. It further allows to visualize the performance history of a matcher, for instance, which correspondences a matcher found in the most recent OAEI campaign but not in the previous one. Another framework for working with alignment files is *VOAR* [472, 473]. It is a Web-based system where users can upload ontologies and alignments. *VOAR* then allows the user to render them with multiple visualization types. The upload size of ontologies, as well as alignments, is restricted so that very large files cannot be uploaded.

Similar to *VOAR*, the *SILK workbench* [549] is also a Web-based tool with a focus on link/correspondence creation between different datasets in the *Linked Open Data Cloud*[12]. Unlike *VOAR*, it usually runs on the user's computer. Matching operations (such as Levenshtein distance [304]) are visualized as nodes in a computation graph. The found correspondences are displayed and can be modified to further specify which concepts should be matched.

Further visualization approaches were pursued by matching system developers to actually fine-tune their systems. All these visualizations are therefore very specific to a particular matching approach. One such example is *YAM++* [364], which is a matching system based on a machine learning approach. Results are visualized in a split view where the class hierarchy of the two input ontologies is shown on each side lines are drawn between the matched classes. The user can modify the alignment with the help of this *graphical user interface* (GUI). In a similar way, the developers of *COMA++* [27] created a user interface for their

[10] `https://maven.apache.org/`

[11] `https://github.com/ernestojimenezruiz/logmap-matcher/tree/master/src/main/java/uk/ac/ox/krr/logmap_lite/io`

[12] `https://lod-cloud.net`

results. A visualization of whole ontologies is not implemented by the current tools but can be achieved with the help of *VOWL* [327] or *Web Protégé* [539], for instance.

Our proposed framework, MELT, allows for detailed and reusable analyses such as the ones presented in this section due to its flexible metrics and evaluators. An overview of the framework is presented in the following section.

4.3 Matching Evaluation Toolkit

MELT is a software framework implemented in Java which aims to facilitate matcher development, configuration, packaging, and evaluation. In this section, we will first introduce *Yet Another Alignment API*, an API for ontology alignment which is integrated into the framework. Afterward, the matcher development process in MELT is introduced. Subsections 4.3.3 and 4.3.4 cover specific aspects of the framework that have not yet been explicitly addressed in the community: The implementation of matchers outside of the Java programming language (4.3.3) and the chaining matching workflows (4.3.4). After explaining the tuning component of the framework, this section closes with the matcher evaluation process in MELT.

4.3.1 YAAA: Yet Another Alignment API

To allow for a simple development workflow, MELT contains *Yet Another Alignment API* (YAAA). It is similar to the *Alignment API* presented earlier but contains additional improvements such as maven support and arbitrary indexing possibilities of correspondence elements allowing queries such as "retrieve all correspondences with a specific source". This is very helpful for a fast evaluation of large-scale test cases containing large reference or system alignments. The indexing is done with the *cqengine* library[13]. The API is, in addition, capable of serializing and parsing alignments. It also makes sure that all characters are escaped and that the resulting XML is actually parseable[14]. As explainability is still an open issue in the ontology matching community [112, 561], YAAA also allows for extensions to correspondences and alignments. This means that additional information such as debugging information or human-readable explanations can be added. If there is additional information available in the alignment, it will also be printed by the default `CSVEvaluator` which allows for immediate

[13] `https://github.com/npgall/cqengine/`

[14] This is not always the case for other implementations.

consumption in the analysis and evaluation process and hopefully fosters the usage of additional explanations in the alignment format.

It is important to note that MELT does not require the usage of YAAA for parameter tuning, executing, or packaging a matcher – but also works with other APIs such as the *Alignment API*. This allows evaluating matchers that were not developed using YAAA (see Section 4.4).

4.3.2 Matcher Development Workflow

In order to develop a matcher in Java with MELT, the first step is to decide which matching interface to implement. The most general interface is encapsulated in class `MatcherURL` which receives two URLs of the ontologies to be matched together with a URL referencing an input alignment. The return value should be a URL representing a file with correspondences in the alignment format. Since this interface is not very convenient, we also provide more specialized classes. In the `matching-yaaa` package we set the alignment library to YAAA. All matchers implementing interfaces from this package have to use the library and get at the same time an easier-to-handle interface of correspondences. In further specializations, we also set the Semantic Web framework, which is used to represent the ontologies. For a better usability, the two most well-known frameworks are integrated into MELT: *Apache Jena*[15] [66] (`MatcherYAAAJena`) and the *OWL API*[16] [219] (`MatcherYAAAOwlApi`). As the latter two classes are organized as separate maven projects, only the libraries which are actually required for the matcher are loaded. In addition, further services were implemented, such as an ontology cache which ensures that ontologies are parsed only once. This is helpful, for instance, when the matcher accesses an ontology multiple times, when multiple matchers work together in a pipeline, or when multiple matchers shall be evaluated. We explicitly chose a framework-independent architecture so that developers can use the full functionality of the frameworks they already know rather than having to understand an additional wrapping layer. The different levels at which a matcher can be developed as well as how the classes presented in this section work together, are displayed in Figure 4.1.

4.3.3 External Matching

The current ontology matching development and evaluation frameworks that are available focus on the Java programming language. As researchers apply advances in machine learning and natural language processing to other domains,

[15] `https://jena.apache.org`
[16] `http://owlcs.github.io/owlapi/`

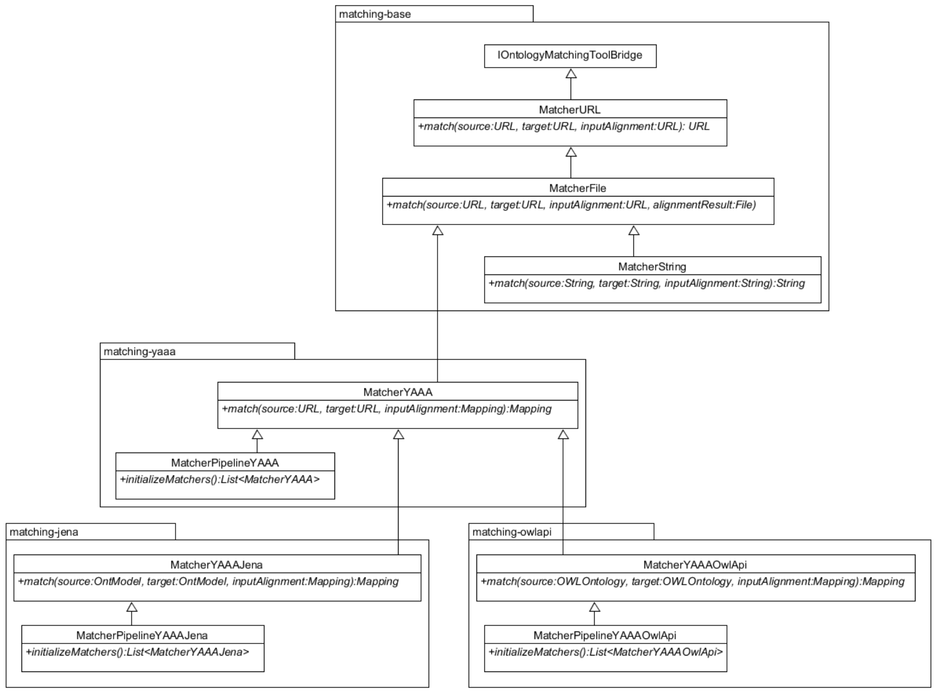


Figure 4.1: Different Possibilities to Implement Matchers

they often turn to Python because leading machine learning libraries such as *scikit-learn*[17], *TensorFlow*[18], *PyTorch*[19], *Keras*[20], or *gensim* [431] are not easily available for the Java language. In the 2018 OAEI campaign, the first tools using such frameworks for ontology matching have been submitted [14].

To accommodate for the changes outlined, MELT allows to develop a matcher in any other programming language and wrap it as a SEALS or HOBBIT package. Therefore, class `MatcherExternal` has to be extended. It has to transform the given ontology URIs and input alignments to an executable command line call. The interface for the external process is simple. It receives the input variables via the command line and outputs the results via the standard output of the process – similar to many Unix command line tools. An example of a ma-

[17] `https://scikit-learn.org/`
[18] `https://www.tensorflow.org/`
[19] `https://pytorch.org/`
[20] `https://keras.io/`

tcher implemented in Python is available on GitHub[21]. It also contains a simple implementation of the alignment format to allow Python matchers to serialize their correspondences.

When executing the matcher with the SEALS client, the matching system is loaded into the *Java virtual machine* (JVM) of the SEALS client (evaluation code) with a customized class loader. This raises two points: 1) The code under test is executed in the same JVM and can probably access the code for evaluation. 2) The used class loader from the *JCL library*[22] does not implement all methods (specifically `getPackage()` and `getResource()`) of a class loader. However, these methods are used by other Java libraries[23] to load operating-system-dependent files contained in the jar file. Thus, some libraries do not work when evaluating a matcher with SEALS. Another problem is that all libraries used by the matching system may collide with libraries used by SEALS. This can cause issues with *Jena* and other Semantic Web frameworks because of the same JVM instance. To solve this issue, `MatcherExternal` can not only be used for matchers written in another programming language but also for Java matches, which use dependencies that are incompatible with the SEALS platform.

4.3.4 Pipelining Matchers

Ontology matchers often combine multiple matching approaches and sometimes consist of the same parts. An example would be a string-based matching of elements and the application of a stable marriage algorithm or another matching refinement step on the resulting similarity matrix.

Following this observation, MELT allows for the chaining of matchers: The alignment of one matcher is then the input for the next matcher in the pipeline. The ontology caching services of MELT mentioned above prevent performance problems arising from repetitive loading and parsing of ontologies.

In order to execute a matcher pipeline, classes `MatcherPipelineYAAA` (for matchers that use different ontology management frameworks), `MatcherPipelineYAAAJena` (for pure *Jena* pipelines), and `MatcherPipelineYAAAOwlApi` (for pure *OWL API* pipelines) can be extended. Here the `initializeMatchers()` method has to be implemented. It returns matcher instances as a `List` in the order in which they shall be executed. These reusable parts of a matcher can

[21] `https://github.com/dwslab/melt/tree/master/examples/externalPythonMatcher`

[22] `https://github.com/kamranzafar/JCL/blob/master/JCL/src/xeus/jcl/AbstractClassLoader.java`

[23] An example would be class `SQLiteJDBCLoader` in `sqlite-jdbc` which uses these class loader methods.

easily be uploaded to GitHub to allow other developers to use common functionality[24].

4.3.5 Tuning Matchers

Many ontology matching systems require parameters to be set at design time. Those can significantly influence the matching system's performance. An example of a parameter would be the threshold parameter of a matcher utilizing a normalized string distance metric. For tuning such a system, MELT offers a `GridSearch` functionality. It requires a matcher and one or more parameters together with their corresponding search spaces, i.e., the values that shall be tested. The Cartesian product of these values is computed, and each system configuration (an element of the Cartesian product which is a tuple of values) runs on the specified test case. The result is an `ExecutionResultSet` which can be further processed like any other result of matchers in MELT. To speed up the execution, class `Executor` was extended and can run matchers in parallel. Properties can be specified by a simple string. Therefore, the `JavaBeans` specification[25] is used to access the properties with so-called setter-methods. This strategy also allows to change properties of nested classes or any list or map. An example of a matcher tuning can be found in the MELT repository[26].

4.3.6 Evaluation Workflow

MELT defines a workflow for matcher execution and evaluation. Therefore, it utilizes the vocabulary used by the OAEI: A matcher can be evaluated on a `TestCase`, i.e., a single ontology matching task. One or more test cases are summarized in a `Track`. MELT contains a built-in `TrackRepository` which allows to access to all OAEI tracks and test cases at design time without actually downloading them from the OAEI Web page. At runtime `TrackRepository` checks whether the required ontologies and alignments are available in the internal buffer; if data is missing, it is automatically downloading and caching it for the next access. The caching mechanism is an advantage over the SEALS platform, which downloads all ontologies again at runtime, which slows down the evaluation process if run multiple times in a row.

[24] Other GitHub dependencies can be included by using `https://jitpack.io`, for instance.

[25] `https://www.oracle.com/technetwork/java/javase/documentation/spec-136004.html`

[26] `https://github.com/dwslab/melt/blob/master/examples/simpleJavaMatcher/src/test/java/de/uni_mannheim/informatik/dws/ontmatching/demomatcher/EvaluateMatcher.java`

One or more matchers are given, together with the track or test case on which they shall be run, to an `Executor`. The `Executor` runs a matcher or a list of matchers on a single test case, a list of test cases, or a track. The `run()` method of the `Executor` returns an `ExecutionResultSet`. The latter is a set of `ExecutionResult` instances that represent individual matching results on a particular test case. Lastly, an `Evaluator` accepts an `ExecutionResultSet` and performs an evaluation. Therefore, it may use one or more `Metric` objects. MELT contains various metrics, such as a `ConfusionMatrixMetric`, and evaluators. Nonetheless, the framework is designed to allow for the further implementation of evaluators and metrics.

After the `Executor` has run, an `ExecutionResult` can be refined by a `Refiner`. A refiner takes an individual `ExecutionResult` and makes it smaller. An example is the `TypeRefiner` which creates additional execution results depending on the type of the alignment (classes, properties, datatype properties, object properties, instances). Another example of an implemented refiner is the `ResidualRefiner` which only keeps non-trivial correspondences and can be used for metrics such as recall+ (see Subsection 2.6.6). Refiners can be combined. This means that MELT can calculate very specific evaluation statistics, such as the residual precision of datatype property correspondences.

A novelty of this framework is also the granularity at which alignments can be analyzed: The `EvaluatorCSV` writes every correspondence in a CSV format together with further details about the matched resources and the performed refinements. This allows for an in-depth analysis in various spreadsheet applications such as LibreOffice Calc, where through the usage of filters, analytical queries can be performed such as “false-positive datatype property matches by matcher X on test case Y”.

4.4 Exemplary Analysis of OAEI 2018 Results

In order to demonstrate the capabilities of MELT, a small analysis of the OAEI 2018 results for the *Conference* and *Anatomy* track has been performed and is presented in the following.

The *Conference* track consists of 16 ontologies from the conference domain. For the exemplary analysis, we evaluated all matching systems that participated in the 2018 campaign: *ALIN* [498], *ALOD2Vec* [409], *AML* [143], *DOME* [200], *FCAMapX* [79], *Holontology* [446], *KEPLER* [254], *Lily* [524], *LogMap* and *LogMapLt* [243], *SANOM* [349], as well as *XMap* [106].

The *Anatomy* track consists of a mapping between the human anatomy and the anatomy of a mouse. In the 2018 campaign, the same matchers mentioned above participated with the addition of *LogMapBio*, a matcher from the *LogMap* family [243].

First, the resulting alignments for *Anatomy*[27] and *Conference*[28] have been downloaded from the OAEI Web site. As both result sets follow the same structure every year, the MELT functions `Executor.loadFromAnatomyResultsFolder()` and `Executor.loadFromConferenceResultsFolder()` were used to load the results. The resulting `ExecutionResultSet` was then handed over to the `MatcherSimilarityMetric` and rendered using the `MatcherSimilarityLatexHeatMapWriter`. As the *Conference* track consists of multiple test cases, the results have to be averaged. Here, out of the available calculation modes in MELT, micro-average was chosen as this calculation mode is also used on the official results page[29] to calculate precision and recall scores. Altogether, the analysis was performed with a few lines of Java code.[30]

Tables 4.1 and 4.2 show the Jaccard overlap [228] of the correspondences rendered as a heat map where darker shades indicate higher similarity. The Jaccard coefficient $J \in [0,1]$ between two alignments a_1 and a_2 with correspondences $corr(a_1)$ and $corr(a_2)$ was obtained as follows:

$$J(a_1, a_2) = \frac{|corr(a_1) \cap corr(a_2)|}{|corr(a_1) \cup corr(a_2)|}$$

In Table 4.1, it can be seen that – despite the various approaches that are pursued by the matching systems – most of them arrive at very similar alignments. One outlier in this statistic is *Holontology*. This is due to the very low number of correspondences overall found by this matching system (456 as opposed to ALIN, which had the second-smallest alignment with 928 matches).

Similarly, the matching systems of the *Conference* track also show commonalities in their alignments, albeit the similarity here is less pronounced compared to the *Anatomy* track: The median similarity (excluding perfect similarities due to self-comparisons) of matching systems for *Anatomy* is $median_{Anatomy} = 0.7223$ whereas the median similarity for *Conference* is $median_{Conference} = 0.5917$. The lower matcher similarity median indicates that *Conference* is a harder matching task because the matching systems have more disagreement about certain correspondences.

[27] `http://oaei.ontologymatching.org/2018/results/anatomy/oaei2018-anatomy-alignments.zip`

[28] `http://oaei.ontologymatching.org/2018/conference/data/conference2018-results.zip`

[29] `http://oaei.ontologymatching.org/2018/results/conference/`

[30] The code to run the analysis can be found on GitHub: `https://github.com/dwslab/melt/tree/master/examples/analyzingMatcherSimilarity`

Table 4.1: OAEI Anatomy 2018 Alignment Similarity

	ALIN	ALOD2Vec	AML	DOME	FCAMapX	Holontology	KEPLER	Lily	LogMap	LogMapBio	LogMapLt	POMAP++	SANOM	XMap
ALIN	1	0.93	0.62	0.97	0.72	0.47	0.79	0.63	0.66	0.6	0.81	0.63	0.62	0.65
ALOD2Vec	0.93	1	0.65	0.94	0.77	0.45	0.81	0.67	0.7	0.63	0.84	0.66	0.64	0.68
AML	0.62	0.65	1	0.62	0.76	0.3	0.74	0.72	0.8	0.82	0.72	0.83	0.79	0.83
DOME	0.97	0.94	0.62	1	0.73	0.47	0.79	0.64	0.66	0.6	0.81	0.63	0.62	0.66
FCAMapX	0.72	0.77	0.76	0.73	1	0.35	0.75	0.69	0.82	0.77	0.89	0.77	0.75	0.78
Holontology	0.47	0.45	0.3	0.47	0.35	1	0.38	0.3	0.32	0.29	0.39	0.31	0.3	0.31
KEPLER	0.79	0.81	0.74	0.79	0.75	0.38	1	0.69	0.78	0.72	0.75	0.76	0.71	0.76
Lily	0.63	0.67	0.72	0.64	0.69	0.3	0.69	1	0.7	0.68	0.69	0.72	0.72	0.72
LogMap	0.66	0.7	0.8	0.66	0.82	0.32	0.78	0.7	1	0.9	0.81	0.81	0.8	0.81
LogMapBio	0.6	0.63	0.82	0.6	0.77	0.29	0.72	0.68	0.9	1	0.74	0.8	0.78	0.78
LogMapLt	0.81	0.84	0.72	0.81	0.89	0.39	0.75	0.69	0.81	0.74	1	0.74	0.74	0.75
POMAP++	0.63	0.66	0.83	0.63	0.77	0.31	0.76	0.72	0.81	0.8	0.74	1	0.79	0.83
SANOM	0.62	0.64	0.79	0.62	0.75	0.3	0.71	0.72	0.8	0.78	0.74	0.79	1	0.78
XMap	0.65	0.68	0.83	0.66	0.78	0.31	0.76	0.72	0.81	0.78	0.75	0.83	0.78	1

In a second step, the same result from the `MatcherSimilarityMetric` has been printed by another writer (`MatcherSimilarityLatexPlotWriter`) which plots the *mean absolute deviation* (MAD) on the X-axis and the F_1 score on the Y-axis. The MAD was obtained for each matcher by applying

$$MAD = \frac{1}{n}\sum_{i=1}^{n} |x_i - mean(X)|$$

where X is the set of Jaccard similarities for a particular matcher. The resulting plots are shown in Figures 4.2 and 4.3. It can be seen that the matchers form different clusters: *Anatomy* matchers with a high F_1 measure also have a high deviation. Consequently, those matchers are likely candidates for a combination to achieve better results. On *Conference*, on the other hand, good combinations cannot be derived because the best matchers measured by their F_1 score tend not to deviate much in their resulting alignments.

In addition to the evaluations performed using the matcher similarity metric, the `EvaluatorCSV` was run using the OAEI 2018 matchers on the *Anatomy* and *Conference* tracks. The resulting CSV file contains one row for each correspondence together with additional information about each resource that is mapped (e.g., label, comment, or type) and with additional information about the correspondence itself (e.g., residual match indicator or evaluation result). All files are available online for further analysis on correspondence level.[31]

[31] `https://github.com/dwslab/melt/tree/master/examples/analyzingMatcherSimilarity`

Table 4.2: OAEI Conference 2018 Alignment Similarity

	ALIN	ALOD2Vec	AML	DOME	FCAMapX	Holontology	KEPLER	Lily	LogMap	LogMapLt	SANOM	XMap
ALIN	1	0.75	0.65	0.84	0.63	0.77	0.53	0.43	0.72	0.76	0.52	0.6
ALOD2Vec	0.75	1	0.58	0.87	0.67	0.75	0.61	0.37	0.67	0.86	0.5	0.54
AML	0.65	0.58	1	0.61	0.58	0.56	0.53	0.45	0.71	0.59	0.63	0.64
DOME	0.84	0.87	0.61	1	0.67	0.81	0.59	0.39	0.7	0.86	0.52	0.56
FCAMapX	0.63	0.67	0.58	0.67	1	0.6	0.55	0.41	0.62	0.66	0.51	0.53
Holontology	0.77	0.75	0.56	0.81	0.6	1	0.53	0.37	0.64	0.72	0.49	0.52
KEPLER	0.53	0.61	0.53	0.59	0.55	0.53	1	0.41	0.57	0.62	0.5	0.54
Lily	0.43	0.37	0.45	0.39	0.41	0.37	0.41	1	0.46	0.39	0.48	0.51
LogMap	0.72	0.67	0.71	0.7	0.62	0.64	0.57	0.46	1	0.7	0.63	0.66
LogMapLt	0.76	0.86	0.59	0.86	0.66	0.72	0.62	0.39	0.7	1	0.51	0.56
SANOM	0.52	0.5	0.63	0.52	0.51	0.49	0.5	0.48	0.63	0.51	1	0.61
XMap	0.6	0.54	0.64	0.56	0.53	0.52	0.54	0.51	0.66	0.56	0.61	1

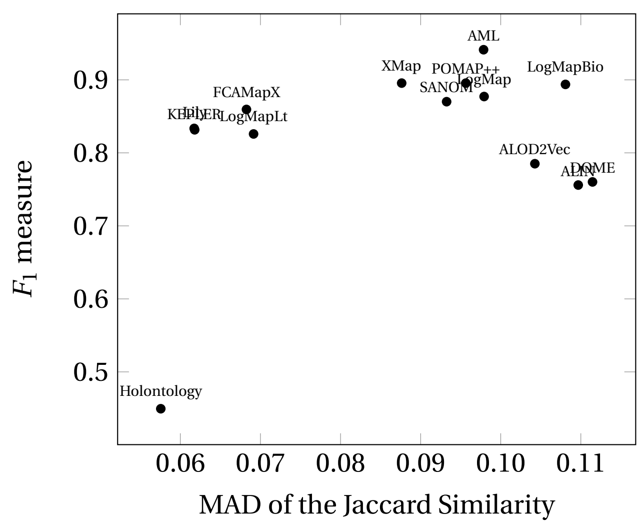


Figure 4.2: Matcher Comparison Using MAD and F_1 on the *Anatomy* Dataset

4.5 Conclusion

With MELT, we have presented a framework for ontology matcher development, configuration, packaging, and evaluation. We hope to lower the entry barriers into the ontology matching community by offering a streamlined development process. MELT can also simplify the work of researchers who evaluate multiple

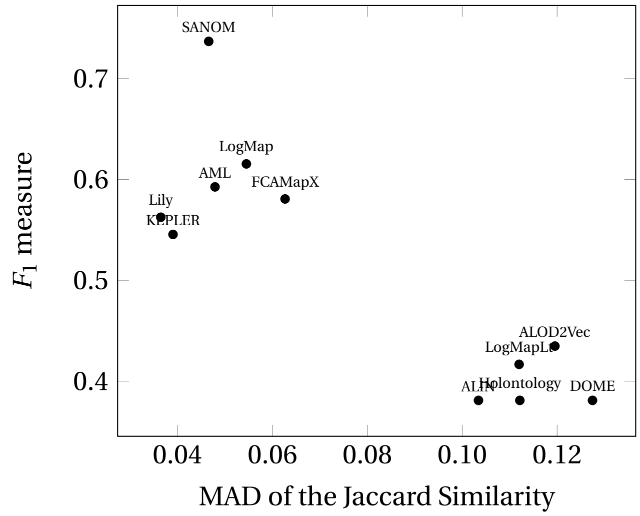


Figure 4.3: Matcher Comparison Using MAD and F_1 on the *Conference* Dataset

matchers on multiple datasets such as OAEI track organizers through its rich evaluation capabilities.

The evaluation capabilities were demonstrated for two OAEI tracks exemplarily by providing a novel view on matcher similarity. The MELT framework, as well as the code used for the analyses presented in this chapter, are open-source and freely available.

Chapter 5

Visual Analysis of Ontology Matching Results with the MELT Dashboard

In the previous chapter, the Matching EvaLuation Toolkit has been introduced. A core feature of MELT is the capability to evaluate and to analyze ontology alignments programmatically through an extensive set of evaluation classes.

In this chapter, an extension is presented which allows (also non-technical) users to analyze and to compare alignments through a Web interface without any set-up efforts. Compared to existing static evaluation interfaces in the ontology matching domain, this dashboard allows for interactive self-service analyses such as a drill down into the matcher performance for data type properties or into the performance of matchers within a certain confidence threshold. In addition, the dashboard offers detailed group evaluation capabilities that allow for the application in broad evaluation campaigns such as the Ontology Alignment Evaluation Initiative.

The interactive dashboard is actively used by the community in the OAEI campaigns 2019 [490], 2020 [491], and 2021 [492].

The work presented in this short chapter has been published before as: Portisch, Jan♠; Hertling, Sven♠; Paulheim, Heiko. Visual Analysis of Ontology Matching Results with the MELT Dashboard. In: The Semantic Web: ESWC 2020 Satellite Events. 2020. [400]

5.1 Architecture

The dashboard can be used for matchers that were developed in *MELT* but also allows for the evaluation of external matchers that use the well-known alignment format of the *Alignment API*. It is implemented in Java and is included by default in the *MELT* 2.0 release, which is available through the maven central repository[1]. The `DashboardBuilder` class is used to generate an HTML[2] file. Without further parameters, a default page can be generated that allows for an in-depth analysis. Alternatively, the dashboard builder allows to completely customize a dashboard before generation – for instance, by adding or deleting selection controls and display panes. After the generation, the self-contained Web page can be viewed locally in the Web browser or be hosted on a server. The page visualization is implemented with *dc.js*[3], a JavaScript charting library with *crossfilter*[4] support. Once generated, the dashboard can be used also by non-technical users to analyze and compare matcher results.

As matching tasks (and the resulting alignment files) can become very large, the dashboard was developed with a focus on performance. For the *OAEI 2019 KnowledgeGraph* track [199, 202], for instance, more than 200,000 correspondences are rendered, and results are recalculated on the fly when the user performs a drill-down selection.

5.2 Use Case and Demonstration

One use case for the framework is OAEI campaigns. The Ontology Alignment Evaluation Initiative is running evaluation campaigns [121] every year since 2005. Researchers submit generic matching systems for predefined tasks (so-called *tracks*), and the track organizers post the results of the systems on each track. The results are typically communicated on the OAEI Web page in a static fashion through one or more tables.[5]

In order to demonstrate the capabilities of the dashboard, we generated pages for the following tracks: *Anatomy*, *Conference*, and *KnowledgeGraph*. We included the first two tracks in one dashboard[6] to show the multi-track capabil-

[1] `https://mvnrepository.com/artifact/de.uni-mannheim.informatik.dws.melt`

[2] *HTML* stands for "HyperText Markup Language".

[3] `https://dc-js.github.io/dc.js/`

[4] `http://crossfilter.github.io/crossfilter/`

[5] For an example, see the *Anatomy Track* results page 2019: `http://oaei.ontologymatching.org/2019/results/anatomy/index.html`

[6] Demo link: `https://dwslab.github.io/melt/anatomy_conference_dashboard.html`

ities of the toolkit. The *KnowledgeGraph* dashboard[7] was officially used in the OAEI 2019 campaign and shows that the dashboard can handle also combined schema and instance matching tasks at scale. The code to generate the dashboards is available in the `example` folder of the *MELT* project.[8] It can be seen that merely a few lines of code are necessary to generate comprehensive evaluation pages.

An annotated screenshot of the controls for the *Anatomy/Conference* dashboard is depicted in Figure 5.1. Each numbered element is clickable in order to allow for a sub-selection. For example, in element (2), the *Conference* track has been selected, and all elements in the dashboard show the results for this subselection. The controls in the given sample dashboard are as follows: (1) selection of the track, (2) selection of the track/test case (the *Conference* track is selected with all test cases), (3) confidence interval of the matchers (an interval of $[0.59, 1.05]$ is selected), (4) relation (only equivalence for this track), (5) matching systems, (6) the share of true/false positives (TP/FP) and false negatives (FN), (7)/(8) the type of the left/right element in each correspondence (e.g., class, object property, datatype property), (9) the share of residual true positives (i.e., non-trivial correspondences generated by a configurable baseline matcher), (10) the total number of correspondences found per test case – the performance result of each match (TP/FP/FN) is color coded, and (11) the color-coded correspondences found per matcher.
Below the controls, the default dashboard shows the performance results per matcher, i.e., micro and macro averages of precision (P), recall (R), and F-score (F_1) in a table as well as concrete correspondences in a further table (both are not shown in Figure 5.1). The data and all controls are updated automatically when a selection is performed. For example, if the *Anatomy* track is selected (control (2)) for matcher *Wiktionary* [402] (control (5)), and only false negative correspondences (control (6)) are desired, the correspondence table will show examples of false negative matches for the *Wiktionary* matching system on the *Anatomy* track.

5.3 Conclusion

In this chapter, we presented the *MELT Dashboard*, an interactive Web user interface for ontology alignment evaluation. The tool allows to generate dashboards easily and to use them for a detailed evaluation in a drill-down fashion.

[7] Demo link: `http://oaei.ontologymatching.org/2019/results/knowledgegraph/knowledge_graph_dashboard.html`

[8] `https://github.com/dwslab/melt/tree/master/examples/meltDashboard`

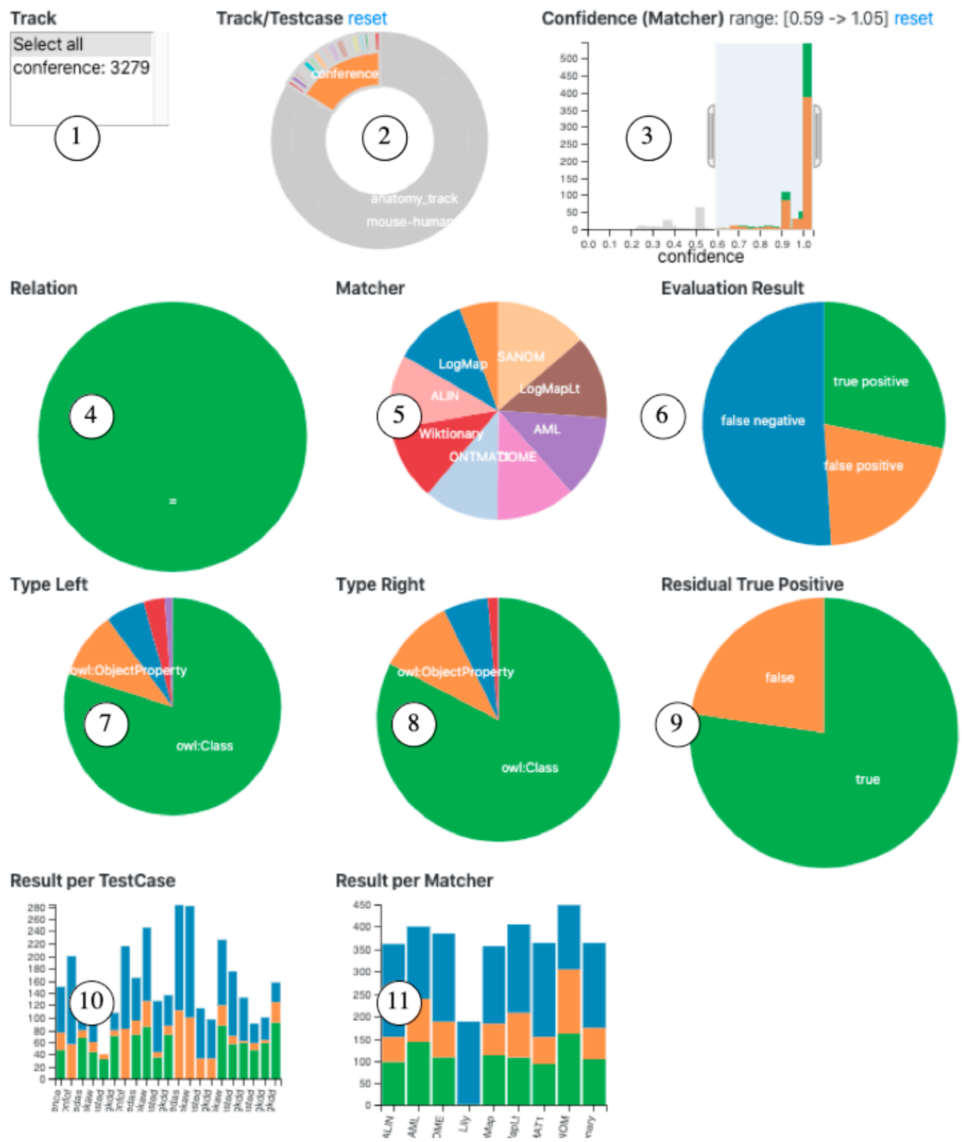


Figure 5.1: Dashboard for the *OAEI Anatomy/Conference Tracks.* The numbered controls are clickable to drill down into the data. If clicked, all elements change automatically to reflect the current selection.

With the new functionality, we hope to increase the transparency and the understanding of matching systems in the ontology alignment community and to make in-depth evaluation capabilities available to a broader audience without the need of installing any software. The first usage in the OAEI 2019 campaign showed that the dashboard can be used for broad evaluation campaigns of multiple matchers on multiple matching tasks.

Chapter 6

Supervised Ontology and Instance Matching in MELT

In this chapter, a machine learning extension to the Matching EvaLuation Toolkit is presented, which facilitates the application of supervised learning for ontology and instance matching. The extension is used to evaluate two supervised machine learning matchers: (1) A latent, RDF2vec-based matching approach and (2) a multi-feature approach for knowledge graphs.

The work presented in this short chapter has been published before as: Hertling, Sven♠; Portisch, Jan♠; Paulheim, Heiko. Supervised Ontology and Instance Matching with MELT. In: The Fifteenth International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020). 2020. [204]

6.1 Introduction

Many similarity metrics and matching approaches have been proposed and developed up to date. They are typically implemented as engineered systems which apply a process-oriented matching pipeline. Manually combining metrics, also called *features* in the machine learning jargon, is typically very cumbersome. Supervised learning allows researchers and developers to focus on adding and defining features and to leave the weighting of those and the decision making to a machine. This approach may also be suitable for developing generic matching systems that self-adapt depending on specific datasets or domains. Here, it makes sense to test and evaluate multiple classifiers at once in a fair, i.e., reproducible, way. Furthermore, recent advances in machine learning – such as in the area of knowledge graph embeddings – may also be applicable to the ontology and instance matching community. The existing evaluation and development platforms, such as the *Alignment API* [96], *SEALS* [162, 577] or the *HOBBIT* [368] framework, make the application of such advances not as simple as it could be.

In this chapter, we present *MELT-ML*, an extension to the *Matching EvaLuation Toolkit* (MELT). Our contribution is twofold: Firstly, we present a machine learning extension to the MELT framework (available in MELT 2.6), which simplifies the application of advanced machine learning algorithms in matching systems and helps researchers evaluate systems that exploit such techniques. Secondly, we present and evaluate two novel approaches in an exemplary manner implemented and evaluated with the extension in order to demonstrate its functionality. We show that RDF2vec [442] embeddings derived directly from the ontologies to be matched are capable of representing the internal structure of an ontology but do not provide any value for matching tasks with differently structured ontologies when evaluated as the only feature. We further show that multiple feature generators and a machine learning component help to obtain a high precision alignment in the *Ontology Alignment Evaluation Initiative knowledge graph* track [216, 199].

6.2 Related Work

Classification is a flavor of *supervised learning* and denotes a machine learning approach where the learning system is presented with a set of records carrying a *class* or *label*. Given those records, the system is trained by trying to predict the correct class. [320] Transferred to the ontology alignment domain, the set of records can be regarded as a collection of correspondences where some of the correspondences are correct (class *true*) and some correspondences are false (class *false*). Hence, the classification system at hand is binary.

The application of supervised learning is not new to ontology matching. In fact, even in the very first edition of the OAEI[1] in 2004, the *OLA* matching system [120] performed a simple optimization of weights using the provided reference alignments. In the past, multiple publications [224, 114, 474, 366, 290] addressed supervised learning in ontology matching, occasionally also referred to as *matching learning*. Unsupervised machine learning approaches are less often used but have been proposed for the task of combining matchers as well [354].

More recently, Nkisi-Orji et al. [374] present a matching system that uses a multitude of features and a random forest classifier. The system is evaluated on

[1] Back then the competition was actually referred to as *EON Ontology Alignment Contest*.

the OAEI *conference* track [77] and the EuroVoc dataset but did not participate in the actual evaluation campaign. Similarly, Wang et al. [558] present a system called *OntoEmma* which exploits a neural classifier together with 32 features. The system is evaluated on the *large biomed* track. However, the system did not participate in an OAEI campaign either. It should be mentioned here that a comparison between systems that have been trained with parts of the reference and systems that have not is not really fair (despite being the typical approach).

Also, a recent OAEI-participating matching system applies supervised learning: The *POMap++* matching system [290] uses a local classifier, which is not based on the reference alignment but on a locally created gold standard. The system also participated in the last two recent OAEI campaigns [291, 289].

The implementations of the approaches are typically not easily reusable or available in a central framework.

6.3 The MELT Framework

Overview MELT [203] is a framework written in Java for ontology and instance matcher development, tuning, evaluation, and packaging. It supports both, HOBBIT and SEALS, two heavily used evaluation platforms in the ontology matching community. The core parts of the framework are implemented in Java, but the evaluation and packaging of matchers implemented in other languages are also supported. Since 2020, MELT is the official framework recommendation by the OAEI, and the MELT track repository is used to provide all track data required by SEALS. MELT is also capable of rendering Web dashboards for ontology matching results so that interested parties can analyze and compare matching results on the level of correspondences without any coding efforts [400] (see the previous chapter). This has been pioneered at the OAEI 2019 for the *knowledge graph* track.[2] MELT is open-source[3], under a permissive license, and is available on the maven central repository[4].

Different Gold Standard Types Matching systems are typically evaluated against a reference alignment. A reference alignment may be complete or only partially complete. The latter means that not all entities in the matching task are aligned and that any entity not appearing in the gold standard cannot be judged. Therefore, the following five levels of completeness can be distinguished: (i) complete,

[2]For a demo of the MELT dashboard, see `https://dwslab.github.io/melt/anatomy_conference_dashboard.html`

[3]`https://github.com/dwslab/melt/`

[4]`https://mvnrepository.com/artifact/de.uni-mannheim.informatik.dws.melt`

(ii) partial with complete target and complete source, (iii) partial with complete target and incomplete source, (iv) partial with complete source and incomplete target, (v) partial with incomplete source and incomplete target. If the reference is complete, all correspondences not available in the reference alignment can be regarded as wrong. If only one part of the gold standard is complete (ii, iii, and iv), every correspondence involving an element of the complete side that is not available in the reference can be regarded as wrong. If the gold standard is incomplete (v), the correctness of correspondences not in the gold standard cannot be judged. For example, given that the gold standard is partial with complete target and complete source (case ii), and given the correspondence $< a, b, =, 1.0 >$, the correspondence $< a, c, =, 1.0 >$ could be judged as wrong because it involves a, which is from the complete side of the alignment. On the other hand, the correspondence $< d, e, =, 1.0 >$ cannot be judged because it does not involve any element from the gold standard. This evaluation setting is used, for example, for the OAEI *knowledge graph* track. OAEI reference datasets are typically complete with the exception of the *knowledge graph* track. The completeness of references influences how matching systems have to be evaluated. MELT can handle all stated levels of completeness. The completeness can be set for every `TestCase` separately using the enum `GoldStandardCompleteness`. The completeness also influences the generation of negative correspondences for a gold standard in supervised learning. MELT supports matching system developers also in this use case.

6.4 Supervised Learning Extensions in MELT

6.4.1 Python Wrapper

As researchers apply advances in machine learning and natural language processing to other domains, they often turn to Python because leading machine learning libraries such as *scikit-learn*[5], *TensorFlow*[6], *PyTorch*[7], *Keras*[8], or *gensim*[9] are not easily available for the Java language. In order to exploit functionalities provided by Python libraries in a consistent manner without a tool break, a wrapper is implemented in MELT which communicates with a Python backend via *Hypertext Transfer Protocol* (HTTP), as depicted in Figure 6.1. The server

[5]https://scikit-learn.org/
[6]https://www.tensorflow.org/
[7]https://pytorch.org/
[8]https://keras.io/
[9]https://radimrehurek.com/gensim/

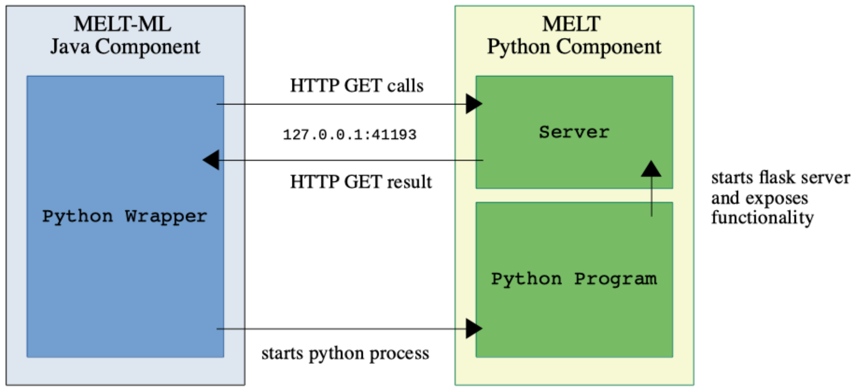


Figure 6.1: Python Code Execution in MELT

works out-of-the-box requiring only that Python and the libraries listed in the `requirements.txt` file are available on the target system. The MELT-ML user can call methods in Java which are mapped to a Python call in the background. As of MELT 2.6, functionality from *gensim* and *scikit-learn* are wrapped.

6.4.2 Generation of Training Data

Every classification approach needs features and class labels. In the case of matching, each example represents a correspondence, and the overall goal is to have an ML model which is capable of deciding if a correspondence is correct or not. Thus, the matching component can only work as a filter, e.g., it can only remove correspondences of an already generated alignment.

For training such a classifier, positive and negative examples are required. The positive ones can be generated by a high precision matcher or by an externally provided alignment such as a sample of the reference alignment or manually created correspondences. As mentioned earlier, no OAEI track provides a dedicated alignment for training. Therefore, MELT provides a new `sample(int n)` method in the `Alignment` class for sampling n correct correspondences as well as `sampleByFraction(double fraction)` for sampling a $fraction$ in range $(0,1)$ of correct correspondences.

Negative examples can be easily generated in settings where the gold standard is complete or partially complete (with complete source and/or target, see Section 6.3). The reason is that any correspondence with an entity appearing in the positive examples can be regarded as incorrect. Thus, a recall-oriented matcher can generate an alignment, and all such correspondences represent the negative class. In cases where the gold standard is partial, and the source and/or target is incomplete, each negative correspondence has to be manually created.

6.4.3 Generation of Features

The features for the correspondences are generated by one or more matches, which can be concatenated in a pipeline or any other control flow. MELT provides an explicit framework for storing the feature values in correspondence extensions (which are by default also serialized in the alignment format). The correspondence method `addAdditionalConfidence(String key, double confidence)` is used to add such feature values (more convenience methods exist).

MELT already provides some out-of-the-box feature generators in the form of so-called *filters* and *matchers*. A *matcher* detects new correspondences. As of MELT 2.6, 17 matchers are directly available (e.g., different string similarity metrics). A *filter* requires an input alignment and adds the additional confidences to the correspondences or removes correspondences below a threshold. In MELT, machine learning is also included via a filter (`MachineLearningScikitFilter`). As of MELT 2.6, 21 filters are available. A selection is presented in the following:

SimilarNeighboursFilter Given an initial alignment of instances, the `SimilarNeighboursFilter` analyzes for each of the instance correspondences how many already matched neighbors the source and target instances share. It can be further customized to also include similar literals (defined by string processing methods). The share of neighbors can be added to the correspondence as absolute value or relative to the total numbers of neighbors for source and target. For the latter, the user can choose from `min` (size of the intersection divided by the minimum number of neighbors of source or target), `max`, `jaccard` (size of intersection divided by the size of the union), and `dice` (twice the size of the intersection divided by the sum of source and target neighbors).

CommonPropertiesFilter This filter selects instance matches based on the overlap of properties. The idea is that equal instances also share similar properties. Especially in the case of homonyms, this filter might help. For instance, given two instances with label “bat”, the string may refer to the mammal or to the racket where the first sense has properties like “taxon”, “age”, or “habitat” and the latter one has properties like “material”, “quality”, or “producer”. This filter, of course, requires already matched properties. The added confidence can be further customized similarly to the previous filter. Furthermore, property URIs are by default filtered to exclude properties like `rdfs:label`.

SimilarHierarchyFilter This component analyzes any hierarchy for given instance matches, such as type hierarchy or a category taxonomy as given in the *knowledge graph* track. Thus, two properties are needed: 1) instance to hierarchy property, which connects the instance to the hierarchy (in case of type hierarchy, this is `rdf:type`) 2) hierarchy property which connects the hierarchy (in case of type hierarchy this is `rdfs:subClassOf`). This filter needs matches in the hierarchy, which are counted similarly to the previous filters. Additionally, the confidence can be computed by a hierarchy level-dependent value (the higher the match in the hierarchy, the lower the confidence). `SimilarTypeFilter` is a reduced version of it by just looking at the direct parent.

BagOfWordsSetSimilarityFilter This filter analyzes the token overlap of the literals given by a specific property. The tokenizer can be freely chosen as well as the overlap similarity.

MachineLearningScikitFilter The actual classification part is implemented in class `MachineLearningScikitFilter`. In the standard setting, a five-fold cross validation is executed to search for the model with the best f-measure. The following models and hyper parameters are tested:

- *Decision trees* optimized by minimum leaf size and maximum depth of tree (1-20)
- *Gradient boosted trees* optimized by maximum depth (1,6,11,16,21) and number of trees (1,21,41,61,81,101)
- *Random forest* optimized by number of trees (1-100 with 10 steps) and minimum leaf size (1-10)
- *Naïve Bayes* (without specific parameter tuning)
- *Support vector machine* (SVM) with radial base function kernel; C and gamma are tuned according to [221]
- *Neural network* (NN) with one hidden layer in two different sizes $F/2+2$, $sqrt(F)$, and two hidden layers of $F/2$ and $sqrt(F)$, where F denotes the number of features

All of these combinations are evaluated automatically with and without feature normalization (`MinMaxScaler` which scales each feature to a range between zero and one). The best model is then trained on the whole training set and applied to the given alignment.

6.4.4 Analysis of Matches

A correspondence which was found by a matching system and which appears in the reference alignment is referred to as *true positive*. A *residual true positive* correspondence is a true positive correspondence that is not trivial as defined by a trivial alignment. The trivial alignment can be given or calculated by a simple baseline matcher. String matches, for instance, are often referred to as trivial. Given a reference alignment, a system alignment, and a trivial alignment, the *residual recall* can be calculated as the share of non-trivial correspondences found by the matching system [7, 131].

If a matcher was trained using a sample of the reference alignment and is also evaluated on the reference alignment, a true positive match can only be counted as meaningful if it was not available in the training set before. In MELT, the baseline matcher can be set dynamically for an evaluation. Therefore, for supervised matching tasks where a sample from the reference is used, the sample can be set as the baseline solution (using the `ForwardMatcher`) so that only additionally found matches are counted as residual true positives. Using the alignment cube file[10], residual true positives can be analyzed at the level of individual correspondences.

6.5 Exemplary Analysis

6.5.1 RDF2vec Vector Projections

Experiment In this experiment, the ontologies to be matched are embedded, and a projection is used to determine matches. *RDF2vec* is a knowledge graph embedding approach which generates random walks for each node in the graph to be embedded and afterward runs the *word2vec* [344, 345] algorithm on the generated walks. Thereby, a vector for each node in the graph is obtained. The RDF graph is used in *RDF2vec* without any pre-processing such as in other approaches like *OWL2Vec* [218]. The embedding approach chosen here has been used on external background knowledge for ontology alignment before [409].

[10] The alignment cube file is a CSV file listing all correspondences found and not found (together with filtering properties) that is generated by the `EvaluatorCSV`.

Multifarm Test Case	P	R	R+	F	# of TP	# of FP	# of FN
iasted-iasted	0.8232	0.7459	0.6111	0.7836	135	29	46
conference-conference	0.7065	0.5285	0.1967	0.6047	65	27	58
confOf-confOf	0.9111	0.5541	0.1081	0.6891	41	4	33

Table 6.1: Performance of RDF2vec projections on the same ontologies in the multifarm track. *P* stands for *precision*, *r* stands for *recall*, and *R+* for *residual recall*. R+ refers here to the fraction of correspondences found that were previously not available in the training set. *# of ...* refers to the number of *true positives (TP)*, *false positives (FP)*, and *false negatives (FN)*. Details about the track can be found in [337]

In this setting, we train embeddings for the ontologies to be matched. In order to do so, we integrate the *jRDF2vec*[11] [405] framework into MELT in order to train the embedding spaces. Using the functionalities provided in the MELT-ML package, we train a linear projection from the source vector space into the target vector space. In order to generate a training dataset for the projection, the `sampleByFraction(double fraction)` method is used. For each source, the closest target node in the embedding space is determined. If the confidence for a match is above a threshold t, the correspondence is added to the system alignment.

Here, we do not apply any additional matching techniques such as string matching. The approach is fully independent of any stated label information. The exemplary matching system is available online as an example.[12]

Results For the vector training, we generate 100 random walks with a depth of 4 per node and train *skip-gram* (SG) embeddings with 50 dimensions, minimum count of 1, and a window size of 5. We use a sampling rate of 50% and a threshold of 0.85. While the implemented matcher fails to generate a meaningful residual recall when the two ontologies to be matched are different, it performs very well when the ontologies are of the same structure as in the *multifarm* track. Here, the approach generates many residual true positives with a residual recall of up to 61% on *iasted-iasted* as seen in Table 6.1. Thus, it could be shown that *RDF2vec* embeddings do contain structural information of the knowledge graph that is embedded.

[11] `https://github.com/dwslab/jRDF2Vec`

[12] `https://github.com/dwslab/melt/tree/master/examples/RDF2vecMatcher`

6.5.2 Knowledge Graph Track Experiments

Experiment In this experiment, the instances of the OAEI *knowledge graph* track are matched. First, a basic matcher (`BaseMatcher`) is used to generate a recall oriented alignment by applying simple string matching on the property values of `rdfs:label` and `skos:altLabel`. The text is compared once using string equality and once in a normalized fashion (non-ASCII characters are removed, and the whole string is lowercased).

Given this alignment, the above-described feature generators/filters are applied in isolation to re-rank the correspondences, and afterward, the `NaiveDescendingExtractor` [338] is used to create a one-to-one alignment based on the best confidence.

In contrast to this, another supervised approach is tried out. After executing the `BaseMatcher`, all feature generators are applied after each other, where each filter adds one feature value. The feature values are calculated independently of each other. This results in an alignment where each correspondence has the additional confidences in its extensions. As a last step, the `MachineLearningScikitFilter` is executed. The training alignment is generated by sampling all correspondences from the `BaseMatcher` where the source *or* target is involved. The correspondence is a positive training example if the source *and* the target appear in the input alignment (which is, in our case, the sampled reference alignment) and a negative example in all other cases.

The search for the machine learning model is executed as five-fold cross validation, and the best model is used to classify all correspondences given by the `BaseMatcher`. The whole setup is available on GitHub[13].

Results In all filters, the absolute number of overlapping entities is used (they are normalized during a grid search for the best model). In the `SimilarNeighboursFilter`, the literals are compared with text equality, and the hierarchy filter compares the categories of the Wiki pages. The `SimilarTypeFilter` analyzes the direct classes which are extracted from templates (indicated by the text 'infobox'). The results for this experiment are depicted in Table 6.2, which shows that no one feature can be used for all test cases because different Wiki combinations (test cases) require different filters. The `BaseMatcher` already achieves a good f-measure which is also in line with previous analyses [202]. When executing the `MachineLearningScikitFilter` the precision can be increased for three test cases, and the associated drop in the recall is relatively small. It can be

[13] `https://github.com/dwslab/melt/tree/master/examples/supervisedKGTrackMatcher`

further seen that there is not one single optimal classifier out of the classifiers tested.

6.6 Conclusion

With MELT-ML, we have presented a machine learning extension for the MELT framework, which facilitates feature generation and feature combination. The latter are included as *filters* to refine existing matches. MELT also allows for the evaluation of ML-based matching systems.

We further would like to emphasize that a special machine learning track with dedicated training and testing alignments might benefit the community, would increase the transparency in terms of matching system performance, and might further increase the number of participants since researchers use OAEI datasets for supervised learning, but there is no official channel to participate if parts of the reference alignment are required.

	mcu-marvel			memoryalpha-memorybeta			memoryalpha-stexpanded			starwars-swg			starwars-swtor		
Approach	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
BaseMatcher	0.8548	0.6796	0.7572	0.8740	0.8978	0.8858	0.8675	0.9264	0.8960	0.9001	0.7318	0.8072	0.9007	0.9146	0.9076
CommonPropertiesFilter	0.8823	0.6614	0.7560	0.9310	0.8785	0.9040	0.9370	0.8968	0.9165	0.9257	0.7162	0.8076	0.9371	0.8999	0.9181
SimilarHierarchyFilter	0.8823	0.6614	0.7560	0.9361	0.8830	0.9088	0.9527	0.9107	0.9312	0.9281	0.7181	0.8097	0.9440	0.9057	0.9245
BagOfWordsSetSimilarityFilter	0.8823	0.6614	0.7560	0.9340	0.8810	0.9067	0.9406	0.8991	0.9194	0.9292	0.7190	0.8107	0.9348	0.8976	0.9159
SimilarNeighboursFilter	0.8912	0.6687	0.7641	0.9467	0.8916	0.9183	0.9600	0.9171	0.9380	0.9375	0.7254	0.8179	0.9317	0.8947	0.9128
SimilarTypeFilter	0.8823	0.6614	0.7560	0.9247	0.8727	0.8980	0.9303	0.8899	0.9096	0.9222	0.7135	0.8045	0.9326	0.8962	0.9140
ML (sample=0.2)	0.8831	0.6620	0.7567	0.9636	0.8592	0.9084	0.9648	0.8887	0.9252	0.9292	0.7190	0.8107	0.9621	0.8778	0.9180
	SVM			Random Forest			SVM			SVM			Random Forest		
ML (sample=0.4)	0.8831	0.6620	0.7567	0.9636	0.8599	0.9088	0.9734	0.8690	0.9182	0.9315	0.7199	0.8121	0.9445	0.8903	0.9166
	Random Forest			Random Forest			Neural Network			Neural Network			Random Forest		
ML (sample=0.6)	0.8831	0.6620	0.7567	0.9685	0.8575	0.9096	0.9667	0.8916	0.9276	0.9367	0.7153	0.8112	0.9565	0.8903	0.9222
	Random Forest			Decision Tree			Neural Network			SVM			SVM		

Table 6.2: Precision (P), recall (R), and f-measure (F) for all five test cases of the *knowledge graph* track using different matching approaches. Details about the track can be found in [202]. For the ML approaches, the optimal classifier (given the evaluated ones outlined in Subsection 6.4.3) is stated below the scores.

Part III

Knowledge Graph Embeddings

In recent years, knowledge graph embeddings emerged as a method to exploit knowledge graphs in downstream applications such as data mining or link prediction. In this dissertation, we are interested in exploring and assessing knowledge graph embeddings for the task of matching ontologies and knowledge graphs. Therefore, in accordance with the research questions presented in Section 1.1, we take a closer look at existing methods and what they actually learn. Chapter 7 introduces the topic in depth and compares knowledge graph embeddings presented exclusively for the task of link prediction with knowledge graph embeddings presented for data mining.

One potential downside of embeddings is the fact that they are typically expensive regarding the training process and regarding their concrete application. Both challenges need to be addressed if embeddings shall be used for ontology matching. Chapter 8 introduces an approach to easily consume embeddings of very large knowledge graphs which is used later in this dissertation used for matching ontologies[14]. In the subsequent Chapter 9, a novel approach is proposed to efficiently train embeddings on very large graphs. This technique is applied in Chapter 18 of Part IV for matching. Chapters 10 and 11 propose concrete adaptions to an existing embedding approach to improve the performance in downstream tasks.

In Chapter 12, we introduce a benchmark that is rooted in description logics constructors to systematically evaluate embedding approaches in general and the RDF2vec configurations presented in this dissertation in particular.

In Chapter 13, a comprehensive evaluation of multiple RDF2vec and enhancements presented in this dissertation part is performed. Combinations of RDF2vec configurations are evaluated as well as multiple benchmark models. Therefore, not only default datasets are used but also the newly developed gold standard presented in the previous chapter.

This part closes with an outlook on how the embedding approach mainly used in this part (RDF2vec) can be generally applied for the task of ontology matching (Chapter 14).

[14] See Chapter 17.

Chapter 7

Knowledge Graph Embedding for Data Mining vs. Knowledge Graph Embedding for Link Prediction – Two Sides of the Same Coin?

In this chapter, the reader is introduced to the topic of knowledge graph embeddings, i.e., projections of entities and relations to lower-dimensional spaces. They have been proposed mainly for two purposes: (1) providing an encoding for data mining tasks and (2) predicting links in a knowledge graph. Both lines of research have been pursued rather in isolation from each other with their own benchmarks and evaluation methodologies. In this chapter, it is evaluated in how far both tasks are actually related. It is shown in two sets of experiments that both approaches can be used for both tasks. The differences in the similarity functions evoked by the different embedding approaches is discussed.

The work presented in this chapter has been published before as: Portisch, Jan; Heist, Nicolas; Paulheim, Heiko. Knowledge Graph Embedding for Data Mining vs. Knowledge Graph Embedding for Link Prediction - Two Sides of the Same Coin?. In: Semantic Web Journal (SWJ). 13(3). Pp. 399–422. 2022. [399]

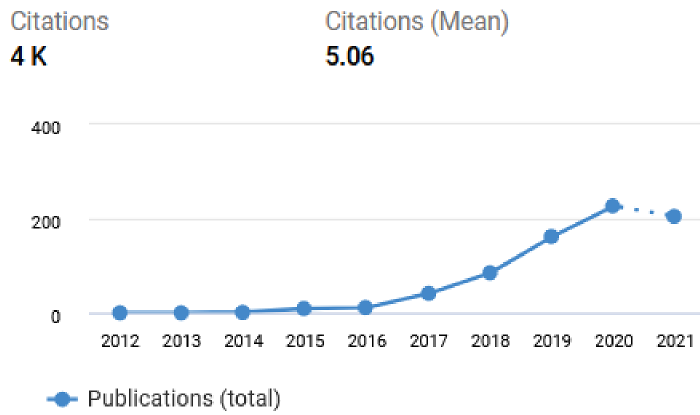


Figure 7.1: Publications with *Knowledge Graph Embedding* in their title or abstract, created with dimensions.ai (as of November 15th, 2021)

7.1 Introduction

In the recent past, the topic of knowledge graph embedding – i.e., projecting entities and relations in a knowledge graph into a numerical vector space – has gained a lot of traction. An often-cited survey from 2017 [559] lists already 25 approaches, with new models being proposed almost every month, as depicted in Figure 7.1.

Even more remarkably, two mostly disjoint strands of research have emerged in that vivid area. The first family of research works focuses mostly on *link prediction* [183], i.e., the approaches are evaluated in a knowledge graph refinement setting [384]. The optimization goal here is to distinguish correct from incorrect triples in the knowledge graph as accurately as possible. The evaluations of this kind of approaches are always conducted within the knowledge graph, using the existing knowledge graph assertions as ground truth.

A second strand of research focuses on the embedding of entities in the knowledge graph for downstream tasks outside the knowledge graph, which often come from the data mining field – hence, we coin this family of approaches *embeddings for data mining*. Examples include the prediction of external variables for entities in a knowledge graph [440], information retrieval backed by a knowledge graph [510], or the usage of a knowledge graph in content-based recommender systems [442]. In those cases, the optimization goal is to create an embedding space which reflects *semantic similarity* as well as possible (e.g., in a recommender system, similar items to the ones in the user interest should be recommended). The evaluations here are always conducted outside the knowledge graph, based on external ground truth.

In this chapter, we want to look at the commonalities and differences between the two approaches. We look at two of the most basic and well-known approaches of both strands, i.e., *TransE* [44] and *RDF2vec* [440], and analyze and compare their optimization goals in a simple example. Moreover, we analyze the performance of approaches from both families in the respective other evaluation setup: we explore the usage of link-prediction-based embeddings for other downstream tasks based on similarity, and we propose a link prediction method based on node embedding techniques such as RDF2vec. From those experiments, we derive a set of insights into the differences between the two families of methods and a few recommendations on which kind of approach should be used in which setting.

7.2 Related Work

As pointed out above, the number of works on knowledge graph embedding is legion, and enumerating them all in this section would go beyond the scope of this dissertation. However, there have already been quite a few survey articles.

The first strand of research works – i.e., knowledge graph embeddings for link prediction – has been covered in different surveys, such as [559], and, more recently, [93, 444, 236]. The categorization of approaches in those reviews is similar, as they distinguish different families of approaches: *translational distance models* [559] or *geometric models* [444] focus on link prediction as a geometric task, i.e., projecting the graph in a vector space so that a translation operation defined for relation r on a head h yields a result close to the tail t.

The second family among the link prediction embeddings are *semantic matching* [559] or *matrix factorization or tensor decomposition* [444] models. Here, a knowledge graph is represented as a three-dimensional tensor, which is decomposed into smaller tensors and/or two-dimensional matrices. The reconstruction operation can then be used for link prediction.

The third and youngest family among the link prediction embeddings are based on deep learning and *graph neural networks* (GNNs). Here, neural network training approaches, such as *convolutional neural networks* (CNNs), capsule networks, or *recurrent neural networks* (RNNs), are adapted to work with knowledge graphs. They are generated by training a deep neural network. Different architectures exist (based on convolutions, recurrent layers, etc.), and the approaches also differ in the training objective, e.g., performing binary classification into true and false triples, or predicting the relation of a triple, given its subject and object [444].

While most of those approaches only consider graphs with nodes and edges, most knowledge graphs also contain literals, e.g., strings and numeric values. Recently, approaches combining textual information with knowledge graph embeddings using language modeling techniques have also been proposed, using techniques such as word2vec and convolutional neural networks [583] or transformer methods [562, 97]. [163] shows a survey of approaches which take such literal information into account. It is also one of the few review articles which consider embedding methods from the different research strands.

Link prediction is typically evaluated on a set of standard datasets and uses a within-KG protocol, where the triples in the knowledge graph are divided into training, testing, and validation set. Prediction accuracy is then assessed on the validation set. Datasets commonly used for the evaluation are FB15k, which is a subset of Freebase, and WN18, which is derived from WordNet [44]. Since it has been remarked that those datasets contain too many simple inferences due to inverse relations, more challenging variants have been proposed, namely FB15k-237 [535] and WN18RR [100]. More recently, evaluation sets based on larger knowledge graphs have been introduced, such as YAGO3-10 [100] and DBpedia50k/DBpedia500k [477].

The second strand of research works, focusing on the embedding for downstream tasks (which are often from the domain of data mining), is not as extensively reviewed, and the number of works in this area is still smaller. One of the more comprehensive evaluations is shown in [85], which is also one of the rare works which includes approaches from both strands in a common evaluation. They show that at least the three methods for link prediction used – namely TransE, TransR, and TransH – perform inferior on downstream tasks compared to approaches explicitly developed for optimizing entity similarity in the embedding space.

A third, yet less closely related strand of research works is *node embeddings for homogeneous graphs*, such as node2vec [173] and DeepWalk [391]. While knowledge graphs come with different relations and are thus considered *heterogeneous*, approaches for homogeneous graphs are sometimes used on knowledge graphs as well by first transforming the knowledge graph into an unlabeled graph, usually by ignoring the different types of relations. Since some of the approaches are defined for undirected graphs, but knowledge graphs are directed, those approaches may also ignore the direction of edges.

For the evaluation of entity embeddings for data mining, i.e., optimized for capturing entity similarity, there are quite a few use cases at hand. The authors in [389] list a number of tasks, including classification and regression of entities

Table 7.1: Co-citation likelihood of different embeddings approaches, obtained from Google scholar, July 12th, 2021. An entry (*row,column*) in the table reads as: *this fraction of the papers citing* column *also cites* row.

	TransE	**TransR**	**RotatE**	**DistMult**	**RESCAL**	**ComplEx**	**RDF2vec**	**KGlove**	**node2vec**	**DeepWalk**
total	*3379*	*1852*	*391*	*1147*	*408*	*1017*	*321*	*73*	*5269*	*5290*
TransE	100,0%	69,7%	80,6%	69,8%	40,9%	73,3%	38,3%	46,6%	5,1%	5,0%
TransR	32,6%	100,0%	36,3%	36,2%	23,8%	38,2%	17,8%	27,4%	3,2%	2,9%
RotatE	10,7%	10,7%	100,0%	22,8%	3,4%	27,3%	3,1%	6,8%	0,5%	0,5%
DistMult	25,2%	26,7%	75,4%	100,0%	16,7%	63,7%	16,2%	21,9%	1,7%	1,5%
RESCAL	22,3%	27,4%	34,0%	38,6%	100,0%	40,8%	14,3%	20,5%	1,6%	1,7%
ComplEx	26,4%	27,2%	73,4%	58,2%	19,9%	100,0%	15,3%	17,8%	1,9%	1,9%
RDF2vec	4,4%	4,6%	4,3%	4,8%	2,5%	5,9%	100,0%	83,6%	1,6%	1,5%
KGlove	1,0%	1,4%	2,0%	1,0%	1,2%	1,4%	13,7%	100,0%	0,3%	0,2%
node2vec	11,1%	11,7%	8,7%	9,4%	5,6%	9,7%	21,5%	31,5%	100,0%	64,1%
DeepWalk	11,7%	11,7%	10,0%	8,8%	6,1%	9,1%	21,8%	23,3%	66,2%	100,0%

based on external ground truth variables, entity clustering, as well as identifying semantically related entities.

Most of the above-mentioned strands exist mainly in their own respective "research bubbles". Table 7.1 shows a co-citation analysis of the different families of approaches. It shows that the Trans* family, together with other approaches for link prediction, forms its own citation network, and so do the approaches for homogeneous networks, while RDF2vec and KGlove are less clearly separated.

Publications which explicitly compare approaches from the different research strands are still rare. In [618], the authors analyze the vector spaces of different embedding models with respect to class separation, i.e., they fit the best linear separation between classes in different embedding spaces. According to their findings, RDF2vec achieves a better linear separation than the models tailored to link prediction.

In [80], an in-KG scenario, i.e., the detection and correction of erroneous links, are considered. The authors compare RDF2vec (with an additional classification layer) to TransE and DistMult on the link prediction task. The results are mixed: While RDF2vec outperforms TransE and DistMult in terms of *mean reciprocal rank* (MRR) and Precision@1, it is inferior in Precision@10. Since the results are only validated on one single dataset, the evidence is rather thin.

Most other research works in which approaches from different strands are compared are related to different downstream tasks. In many cases, the results are rather inconclusive, as the following examples illustrate:

- [68] and [261] both analyze *drug-drug interaction*, using different sets of embedding methods. The finding of [68] is that RDF2vec outperforms

TransE and TransD, whereas in the experiment in [261], ComplEx outperforms RDF2vec, KGlove, TransE, and CrossE, and, in particular, TransE outperforms RDF2vec.

- [29], [80], and [568] all analyze *link prediction in different graphs*. While [29] state that RotatE and TransD outperform TransE, DistMult, and ComplEx, which in turn outperforms node2vec, [80] reports that DistMult outperforms RDF2vec, which in turn outperforms TransE, while [568] reports that KG2vec (which can be considered equivalent to RDF2vec) outperforms node2vec, which in turn outperforms TransE.

- [347] compare the performance of RDF2vec, DistMult, TransE, and SimplE on a set of classification and clustering datasets. The results are mixed. For classification, the authors use four different learning algorithms, and the variance induced by the learning algorithms is most often higher than that induced by the embedding method. For the clustering, they report that TransE outperforms the other approaches.[1]

While this is not a comprehensive list, these observations hint at a need both for more task-specific benchmark datasets as well as for ablation studies analyzing the interplay of embedding methods and other processing steps. Moreover, it is important to gain a deeper understanding of how these approaches behave with respect to different downstream problems and to have more direct comparisons. This chapter aims at closing the latter gap.

7.3 Knowledge Graph Embedding Methods for Data Mining

Traditionally, most data mining methods are working on propositional data, i.e., each instance is a row in a table, described by a set of (binary, numeric, or categorical) features. For using knowledge graphs in data mining, one needs to either develop methods which work on graphs instead of propositional data or find ways to represent instances of the knowledge graph as feature vectors [441]. The latter is often referred to as *propositionalization* [439].

RDF2vec [440] is a prominent example from the second family. It adapts the word2vec approach [344] for deriving word embeddings (i.e., vector representations for words) from a corpus of sentences. RDF2vec creates such sentences

[1] We think that these results must be taken with a grain of salt. To evaluate the clustering quality, the authors use an intrinsic evaluation metric, i.e., the Silhouette score, which is computed in the respective vector space. It is debatable, however, whether Silhouette scores computed in different vector spaces are comparable.

by performing *random walks* on an RDF graph and collecting the sequences of entities and relations, then trains a word2vec model on those sequences. It has been shown that this strategy outperforms other strategies of propositionalization. The relation between propositionalization and embedding methods has also recently been pointed out by [295].

7.3.1 Data Mining is based on Similarity

Predictive data mining tasks are predicting classes or numerical values for instances. Typically, the target is to predict an external variable not contained in the knowledge graph (or, to put it differently: use the background information from the knowledge graph to improve prediction models). One example would be to predict the popularity of an item (e.g., a book, a music album, a movie) as a numerical value. The idea here would be that two items that share similar features should also receive similar ratings. The same mechanism is also exploited in recommender systems: if two items share similar features, users who consumed one of those items are recommended the other one.

RDF2vec has been shown to be usable for such cases since the underlying method tends to create similar vectors for similar entities, i.e., position them closer in vector space [442]. Figure 7.2 illustrates this using a 2D *principal component analysis* (PCA) plot of RDF2vec vectors for movies in DBpedia. It can be seen that clusters of movies, e.g., Disney movies, Star Trek movies, and Marvel-related movies are formed.

Many techniques for predictive data mining rely on similarity in one or the other way. This is more obvious for, e.g., *k-nearest neighbors* (KNN), where the predicted label for an instance is the majority or average of labels of its closest neighbors (i.e., most similar instances), or naïve Bayes, where an instance is predicted to belong to a class if its feature values are most similar to the typical distribution of features for this class (i.e., it is similar to an average member of this class). A similar argument can be made for neural networks, where one can assume a similar output when changing the value of one input neuron (i.e., one feature value) by a small delta. Other classes of approaches (such as support vector machines) use the concept of *class separability*, which is similar to exploiting similarity: datasets with well separable classes have similar instances (belonging to the same class) close to each other, while dissimilar instances (belonging to different classes) are further away from each other [521].

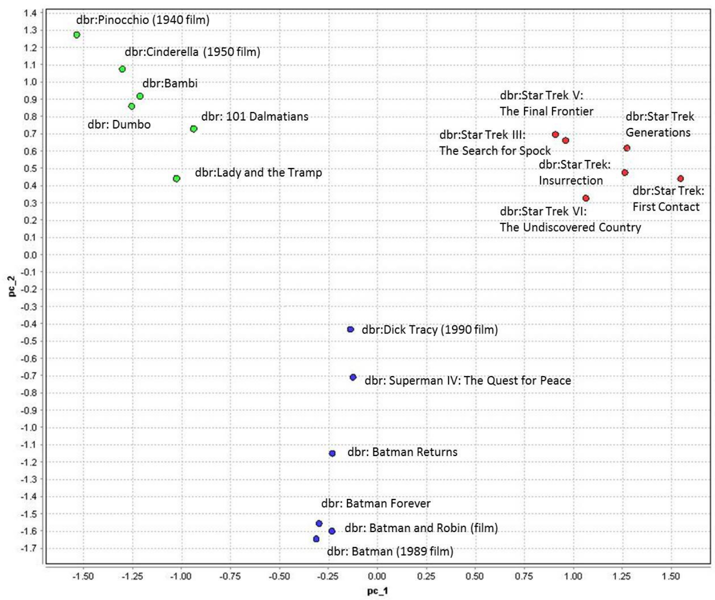


Figure 7.2: RDF2vec embeddings for movies in DBpedia, from [442].

7.3.2 Creating Similar Embeddings for Similar Instances

To understand how (and why) RDF2vec creates embeddings that project similar entities to nearby vectors, we use the running example depicted in Figure 7.3 and Figure 7.4, showing a number of European cities, countries, and heads of those governments.

As discussed above, the first step of RDF2vec is to create random walks on the graph. To that end, RDF2vec starts a fixed number of random walks of a fixed maximum length from each entity. Since the example above is very small, we will, for the sake of illustration, enumerate *all* walks of length 4 that can be created for the graph. Those walks are depicted in Figure 7.5. It is notable that, since the graph has nodes without outgoing edges, some of the walks are actually shorter than 4.

In the next step, the walks are used to train a predictive model. Since RDF2vec uses word2vec, it can be trained with the two flavors of word2vec, i.e., *continuous bag-of-words* (CBOW) and *skip-gram* (SG). The first predicts a word, given its surrounding words, and the second predicts the surroundings, given a word. For the sake of our argument, we will only consider the second variant, depicted in Figure 7.6. Simply speaking, given training examples where the input is the target word (as a one-hot-encoded vector) and the output is the context words

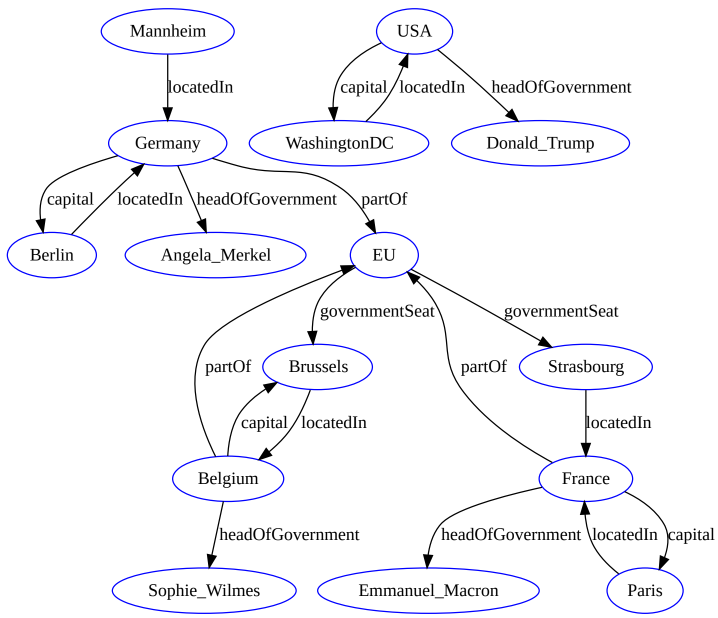


Figure 7.3: Example Graph Used for Illustration

(again, one-hot encoded vectors), a neural network is trained, where the hidden layer is typically of smaller dimensionality than the input. That hidden layer is later used to produce the actual embedding vectors.

To create the training examples, a window with a given size is slid over the input sentences. Here, we use a window of size 2, which means that the two words preceding and the two words succeeding a context word are taken into consideration. Table 7.2 shows the training examples generated for three instances.

A model that learns to predict the context given the target word would now learn to predict the *majority* of the context words for the target word at hand at the output layer called *output* in Figure 7.6, as depicted in the lower part of Table 7.2. Here, we can see that Paris and Berlin share two out of four predictions, and so do Mannheim and Berlin. Angela Merkel and Berlin share one out of four predictions.[2]

Considering again Figure 7.6, given that the activation function which computes the *output* from the *projection* values is continuous, it implies that similar activations on the output layer require similar values on the projection layer.

[2]Note that in the classic formulation of RDF2vec (and word2vec), the position at which a prediction appears does not matter. The order-aware variant *RDF2vec*$_{oa}$ [412] uses an order-aware formulation of word2vec [319].

Table 7.2: Training examples for instances *Paris*, *Berlin*, *Mannheim*, *Angela Merkel*, *Donald Trump*, and *Belgium* (upper part) and majority predictions (lower part).

Target Word	w_{-2}	w_{-1}	w_{+1}	w_{+2}
Paris	France	capital	locatedIn	France
Paris	–	–	locatedIn	France
Paris	–	–	locatedIn	France
Paris	–	–	locatedIn	France
Paris	France	capital	–	–
Paris	France	capital	–	–
Berlin	–	–	locatedIn	Germany
Berlin	Germany	capital	–	–
Berlin	–	–	locatedIn	Germany
Berlin	–	–	locatedIn	Germany
Berlin	Germany	capital	locatedIn	Germany
Berlin	Germany	capital	–	–
Mannheim	–	–	locatedIn	Germany
Mannheim	–	–	locatedIn	Germany
Mannheim	–	–	locatedIn	Germany
Angela Merkel	Germany	headOfGovernment	–	–
Angela Merkel	Germany	headOfGovernment	–	–
Angela Merkel	Germany	headOfGovernment	–	–
Donald Trump	USA	headOfGovernment	–	–
Donald Trump	USA	headOfGovernment	–	–
Belgium	–	–	partOf	EU
Belgium	–	–	capital	Brussels
Belgium	Brussels	locatedIn	–	–
Belgium	–	–	partOf	EU
Belgium	–	–	headOfGovernment	Sophie Wilmes
Belgium	Brussels	locatedIn	headOfGovernment	Sophie Wilmes
Belgium	Brussels	locatedIn	partOf	EU
Belgium	Brussels	locatedIn	capital	Brussels
Belgium	Brussels	locatedIn	–	–
Paris	France	capital	locatedIn	France
Berlin	Germany	capital	locatedIn	Germany
Mannheim	–	–	locatedIn	Germany
Angela Merkel	Germany	headOfGovernment	–	–
Donald Trump	USA	headOfGovernment	–	–
Belgium	Brussels	locatedIn	partOf	EU

```
Berlin locatedIn Germany .
Germany headOfGovernment Angela_Merkel .
Mannheim locatedIn Germany .
Belgium capital Brussels .
Germany partOf EU .
Belgium partOf EU .
Belgium headOfGovernment Sophie_Wilmes .
EU governmentSeat Brussels .
USA capital WashingtonDC .
WashingtonDC locatedIn USA .
France capital Paris .
France headOfGovernment Emmanuel_Macron .
Paris locatedIn France .
Strasbourg locatedIn France .
Germany capital Berlin .
Brussels locatedIn Belgium .
France partOf EU .
USA headOfGovernment Donald_Trump .
EU governmentSeat Strasbourg .
```

Figure 7.4: Triples of the Example Knowledge Graph

Hence, for a well-fit model, the distance on the projection layer of Paris, Berlin, and Mannheim should be comparatively lower than the distance of the other entities, since they activate similar outputs.[3]

Figure 7.7 depicts a two-dimensional RDF2vec embedding learned for the example graph.[4] We can observe that there are clusters of persons, countries, and cities. The grouping of similar objects also goes further – we can, e.g., observe that European cities in the dataset are embedded closer to each other than to Washington D.C. This is in line with previous observations showing that RDF2vec is particularly well suited to create clusters also for finer-grained classes [500]. A predictive model could now exploit those similarities, e.g., for type prediction, as proposed in [264] and [500].

7.3.3 Usage for Link Prediction

From Figure 7.7, we can assume that link prediction should, in principle, be possible. For example, the predictions for heads of governments all point in a sim-

[3]Note that there are still weights learned for the individual connections between the projection and the output layer, which emphasize some connections more strongly than others. Hence, we cannot simplify our argumentation in a way like “with two common context words activated, the entities must be projected twice as close as those with one common context word activated”.

[4]Created with PyRDF2vec [543], using two dimensions, a walk length of 8, and standard configuration otherwise

```
Belgium partOf EU governmentSeat Brussels
Belgium capital Brussels locatedIn Belgium
Belgium partOf EU governmentSeat Strasbourg
Belgium headOfGovernment Sophie_Wilmes
Berlin locatedIn Germany capital Berlin
Berlin locatedIn Germany headOfGovernment Angela_Merkel
Berlin locatedIn Germany partOf EU
Brussels locatedIn Belgium headOfGovernment Sophie_Wilmes
Brussels locatedIn Belgium partOf EU
Brussels locatedIn Belgium capital Brussels
EU governmentSeat Strasbourg locatedIn France
EU governmentSeat Brussels locatedIn Belgium
France headOfGovernment Emmanuel_Macron
France capital Paris locatedIn France
France partOf EU governmentSeat Brussels
France partOf EU governmentSeat Strasbourg
Germany partOf EU governmentSeat Brussels
Germany partOf EU governmentSeat Strasbourg
Germany capital Berlin locatedIn Germany
Germany headOfGovernment Angela_Merkel
Mannheim locatedIn Germany capital Berlin
Mannheim locatedIn Germany headOfGovernment Angela_Merkel
Mannheim locatedIn Germany partOf EU
Paris locatedIn France headOfGovernment Emmanuel_Macron
Paris locatedIn France partOf EU
Paris locatedIn France capital Paris
Strasbourg locatedIn France capital Paris
Strasbourg locatedIn France headOfGovernment Emmanuel_Macron
Strasbourg locatedIn France partOf EU
USA headOfGovernment Donald_Trump
USA capital Washington_DC locatedIn USA
Washington_DC locatedIn USA capital Washington_DC
Washington_DC locatedIn USA headOfGovernment Donald_Trump
```

Figure 7.5: Walks Extracted From the Example Graph

ilar direction. This is in line with what is known about word2vec, which allows for computing analogies, like the well-known example

$$v(King) - v(Man) + v(Woman) \approx v(Queen) \tag{7.1}$$

RDF2vec does not learn relation embeddings, only entity embeddings.[5] Hence, we cannot directly predict links, but we can exploit those analogies. If we want to make a tail prediction like

$$< h, r, ? >, \tag{7.2}$$

[5] Technically, we can also make RDF2vec learn embeddings for the relations, but they would not behave the way we need them.

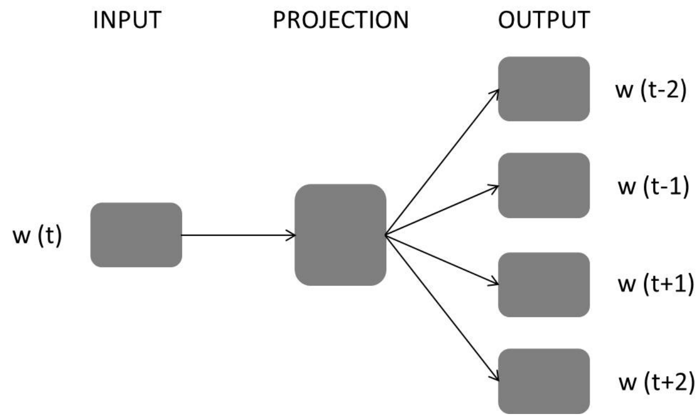


Figure 7.6: The skipgram Variant of word2vec [440]

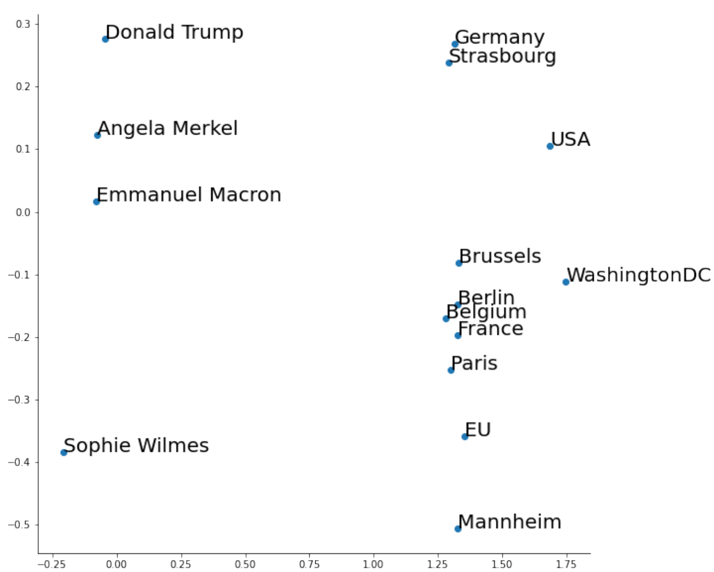


Figure 7.7: The Example Graph Embedded With RDF2vec

we can identify another pair $< h', r, t' >$ and exploit the above analogy, i.e.,

$$t' - h' + h \approx t \tag{7.3}$$

To come to a stable prediction, we would use the average, i.e.,

$$t \approx \frac{\sum_{<h',r,t'>} t' - h' + h}{|< h', r, t' >|}, \tag{7.4}$$

where $|< h', r, t' >|$ is the number of triples which have r as predicate.

With the same idea, we can also average the relation vectors r for each relation that holds between all its head and tail pairs, i.e.,

$$r \approx \frac{\sum_{<h',r,t'>} t' - h'}{|< h', r, t' >|}, \tag{7.5}$$

and thereby reformulate the above equation to

$$t \approx h + r, \tag{7.6}$$

which is what we expect from an embedding model for link prediction. Those approximate relation vectors for the example at hand are depicted in Figure 7.8. We can see that in some (not all) cases, the directions of the vectors are approximately correct: the *partOf* vector is roughly the difference between *EU* and *Germany*, *France*, and *Belgium*, and the *headOfGovernment* vector is approximately the vector between the countries and the politicians cluster.

It can also be observed that the vectors for *locatedIn* and *capitalOf* point in reverse directions, which makes sense because they form connections between two clusters (countries and cities) in opposite directions.

7.4 Knowledge Graph Embedding Methods for Link Prediction

A larger body of work has been devoted to knowledge graph embedding methods for link prediction. Here, the goal is to learn a model which embeds entities and relations in the same vector space.

7.4.1 Link Prediction is based on Vector Operations

As the main objective is link prediction, most models, more or less, try to find a vector space embedding of entities and relations so that

$$t \approx h \oplus r \tag{7.7}$$

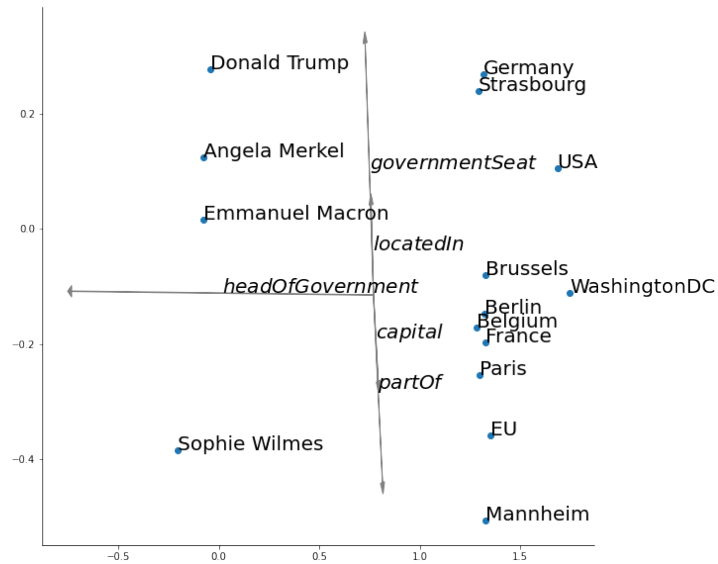


Figure 7.8: Average Relation Vectors for the Example

holds for as many triples $< h, r, t >$ as possible. $\oplus$ can stand for different operations in the vector space; in basic approaches, simple vector addition (+) is used. In our considerations below, we will also use vector addition.

In most approaches, negative examples are created by corrupting an existing triple, i.e., replacing the head or tail with another entity from the graph (some approaches also foresee corrupting the relation). Then, a model is learned which tries to tell apart corrupted from non-corrupted triples. The formulation in the original TransE paper [44] defines the loss function L as follows:

$$L = \sum_{\substack{(h,r,t)\in S,\\ (h',r,t')\in S'}} [\gamma + d(h+r,t) - d(h'+r,t')]_+ \tag{7.8}$$

where γ is some margin, and d is a distance function, i.e., the $L1$ or $L2$ norm. S is the set of statements that are in the knowledge graph, and S' are the corrupted statements derived from them. In words, the formula states for a triple $< h, r, t >$, $h + r$ should be closer to t than to t' for some corrupted tail, similarly for a corrupted head. However, a difference of γ is accepted.

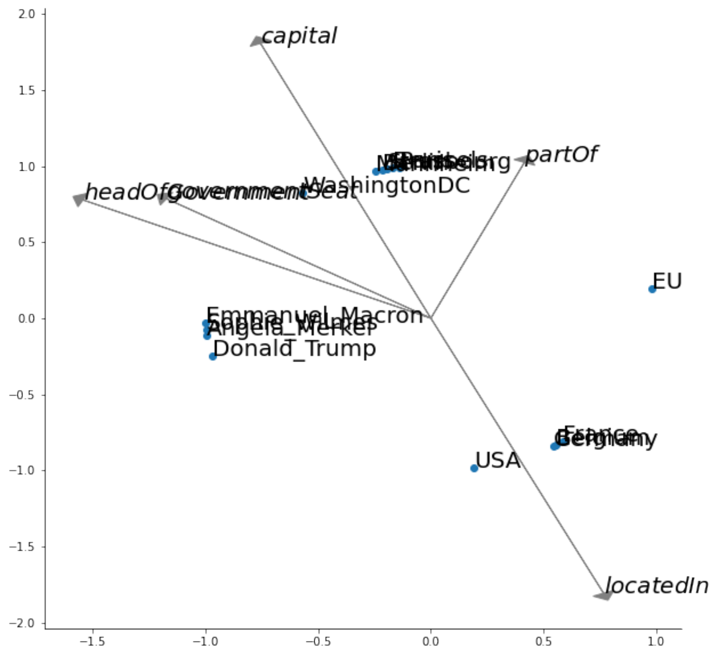


Figure 7.9: Example Graph Embedded by TransE

Figure 7.9 shows the example graph from above, as embedded by TransE.[6] Looking at the relation vectors, it can be observed that they seem approximately accurate in some cases, e.g.,

$$Germany + headOfGovernment \approx Angela\_Merkel,$$

but not everywhere.[7]

Like in the RDF2vec example above, we can observe that the two vectors for *locatedIn* and *capital* point in opposite directions. Also similar to the RDF2vec example, we can see that entities in similar classes form clusters: cities are mostly in the upper part of the space, people in the left, and countries in the lower right part.

[6] Created with PyKEEN [15], using 128 epochs, a learning rate of 0.1, the softplus loss function, and default parameters otherwise, as advised by the authors of PyKEEN: `https://github.com/pykeen/pykeen/issues/97`

[7] This does not mean that TransE does not work. The training data for the very small graph is rather scarce, and two dimensions might not be sufficient to find a good solution here.

7.4.2 Usage for Data Mining

As discussed above, positioning similar entities close in a vector space is an essential requirement for using entity embeddings in data mining tasks. To understand why an approach tailored towards link prediction can also, to a certain extent, cluster similar instances together (although not explicitly designed for this task), we first rephrase the approximate link prediction Equation 7.8 as

$$t = h + r + \eta_{h,r,t}, \tag{7.9}$$

where $\eta_{h,r,t}$ can be considered an error term for the triple $< h, r, t >$. Moreover, we define

$$\eta_{max} = \max_{<h,r,t>\in S} \eta_{h,r,t} \tag{7.10}$$

Next, we consider two triples $< h_1, r, t >$ and $< h_2, r, t >$, which share a relation to an object – e.g., in our example, France and Belgium, which both share the relation *partOf* to *EU*. In that case,

$$t = h_1 + r + \eta_{h_1,r,t} \tag{7.11}$$

and

$$t = h_2 + r + \eta_{h_2,r,t} \tag{7.12}$$

hold. From that, we get[8]

$$\begin{aligned} & h_1 - h_2 && = \eta_{h_2,r,t} - \eta_{h_1,r,t} \\ \Rightarrow \quad & |h_1 - h_2| && = |\eta_{h_2,r,t} - \eta_{h_1,r,t}| \\ & && = |\eta_{h_2,r,t} + (-\eta_{h_1,r,t})| \\ & && \leq |\eta_{h_2,r,t}| + |-\eta_{h_1,r,t}| \\ & && = |\eta_{h_2,r,t}| + |\eta_{h_1,r,t}| \\ & && \leq 2 \cdot \eta_{max} \end{aligned} \tag{7.13}$$

In other words, η_{max} also imposes an upper bound of two entities sharing a relation to an object. As a consequence, the lower the error in relation prediction, the closer are entities which share a common statement.

This also carries over to entities sharing the same two-hop connection. Consider two further triples

[8] Using the triangle inequality for the first inequation.

$< h_{1a}, r_a, h_1 >$ and $< h_{2a}, r_a, h_2 >$. In our example, this could be two cities located in the two countries, e.g., *Strasbourg* and *Brussels*. In that case, we would have

$$h_1 = h_{1a} + r_a + \eta_{h_{1a}, r_a, h_1} \tag{7.14}$$
$$h_2 = h_{2a} + r_a + \eta_{h_{2a}, r_a, h_2} \tag{7.15}$$

Substituting this in (7.11) and (7.12) yields

$$t = h_{1a} + r_a + \eta_{h_{1a}, r_a, h_1} + r + \eta_{h_1, r, t} \tag{7.16}$$
$$t = h_{2a} + r_a + \eta_{h_{2a}, r_a, h_2} + r + \eta_{h_2, r, t}. \tag{7.17}$$

Consequently, using similar transformations as above, we get

$$\begin{aligned} & h_{1a} - h_{2a} && = \eta_{h_{2a}, r_a, h_2} - \eta_{h_{1a}, r_a, h_1} + \eta_{h_2, r, t} - \eta_{h_1, r, t} \\ \Rightarrow \quad & |h_{1a} - h_{2a}| && \leq 4 \cdot \eta_{max} \end{aligned} \tag{7.18}$$

Again, η_{max} constrains the proximity of the two entities h_{1a} and h_{2a}, but only half as strictly as for the case of h_1 and h_2.

7.4.3 Comparing the Two Notions of Similarity

In the examples above, we can see that embeddings for link prediction have a tendency to project similar instances close to each other in the vector space. Here, the notion of similarity is that two entities are similar if they share a relation to another entity, i.e., e_1 and e_2 are considered similar if there exist two statements $< e_1, r, t >$ and $< e_2, r, t >$ or $< h, r, e_1 >$ and $< h, r, e_2 >$,[9] or, less strongly, if there exists a chain of such statements. More formally, we can write the notion of similarity between two entities in link prediction approaches as

$$e_1 \approx e_2 \leftarrow \exists \quad t, r : r(e_1, t) \land r(e_2, t) \tag{7.19}$$
$$e_1 \approx e_2 \leftarrow \exists \quad h, r : r(h, e_1) \land r(h, e_2) \tag{7.20}$$

In other words: two entities are similar if they share a common connection to a common third entity.

RDF2vec, on the other hand, covers a wider range of such similarities. Looking at Table 7.2, we can observe that two entities sharing a common relation to two different objects are also considered similar (*Berlin* and *Mannheim* both share the fact that they are *located in Germany*, hence, their predictions for w_{+1} and w_{+2} are similar).

[9]The argument in Section 7.4.2 would also work for shared relations to common heads.

However, there in RDF2vec, similarity can also come in other notions. For example, *Germany* and *USA* are also considered similar, because they both share the relations *headOfGovernment* and *capital*, albeit with different object (i.e., their prediction for w_1 is similar). In contrast, such similarities do not lead to close projections for link prediction embeddings. In fact, in Figure 7.9, it can be observed that *USA* and *Germany* are further away than *Germany* and other European countries. In other words, the following two notions of similarity also hold for RDF2vec:

$$e_1 \approx e_2 \leftarrow \quad \exists \quad t_1, t_2, r : r(e_1, t_1) \wedge r(e_2, t_2) \tag{7.21}$$

$$e_1 \approx e_2 \leftarrow \quad \exists \quad h_1, h_2, r : r(h_1, e_1) \wedge r(h_2, e_2) \tag{7.22}$$

On a similar argument, RDF2vec also positions entities closer which share *any* relation to another entity. Although this is not visible in the two-dimensional embedding depicted in Figure 7.7, RDF2vec would also create vectors with some similarity for *Angela Merkel* and *Berlin*, since they both have a (albeit different) relation to Germany (i.e., their prediction for w_{-2} is similar). Hence, the following notions of similarity can also be observed in RDF2vec:

$$e_1 \approx e_2 \leftarrow \quad \exists \quad t, r_1, r_2 : r_1(e_1, t) \wedge r_2(e_2, t) \tag{7.23}$$

$$e_1 \approx e_2 \leftarrow \quad \exists \quad h, r_1, r_2 : r_1(h, e_1) \wedge r_2(h, e_2) \tag{7.24}$$

The example with *Angela Merkel* and *Berlin* already hints at a slightly different notion of the interpretation of proximity in the vector space evoked by RDF2vec: not only *similar*, but also *related* entities are positioned close in the vector space. This means that to a certain extent, RDF2vec mixes the concepts of similarity and relatedness in its distance function. We will see examples of this in later considerations, and discuss how they interfere with downstream applications.

7.5 Experiments

To compare the two sets of approaches, we use standard setups for evaluating knowledge graph embedding methods for data mining as well as for link prediction.

7.5.1 Experiments on Data Mining Tasks

In our experiments, we follow the setup proposed in [443] and [389]. Those works propose the use of data mining tasks with external ground truth, e.g., predicting certain indicators or classes for entities. Those entities are then linked to a knowledge graph. Different feature extraction methods – which include the generation of embedding vectors – can then be compared using a fixed set of learning methods.

The setup of [389] comprises six tasks using 20 datasets in total:

- Five classification tasks, evaluated by accuracy. Those tasks use the same ground truth as the regression tasks (see below), where the numeric prediction target is discretized into high/medium/ low (for the Cities, AAUP, and Forbes dataset) or high/low (for the Albums and Movies datasets). All five tasks are single-label classification tasks.

- Five regression tasks, evaluated by *root mean squared error* (RMSE). Those datasets are constructed by acquiring an external target variable for instances in knowledge graphs which is not contained in the knowledge graph per se. Specifically, the ground truth variables for the datasets are: a quality of living indicator for the Cities dataset, obtained from Mercer; average salary of university professors per university, obtained from the AAUP; profitability of companies, obtained from Forbes; average ratings of albums and movies, obtained from Facebook.

- Four clustering tasks (with ground truth clusters), evaluated by accuracy. The clusters are obtained by retrieving entities of different ontology classes from the knowledge graph. The clustering problems range from distinguishing coarser clusters (e.g., cities vs. countries) to finer ones (e.g., basketball teams vs. football teams).

- A document similarity task (where the similarity is assessed by computing the similarity between entities identified in the documents), evaluated by the harmonic mean of Pearson and Spearman correlation coefficients. The dataset is based on the LP50 dataset [299]. It consists of 50 documents, each of which has been annotated with DBpedia entities using DBpedia spotlight [340]. The task is to predict the similarity of each pair of documents.

- An entity relatedness task (where semantic similarity is used as a proxy for semantic relatedness), evaluated by Kendall's Tau. The dataset is based on the KORE dataset [215]. The dataset consists of 20 seed entities from the YAGO knowledge graph, and 20 related entities each. Those 20 related entities per seed entity have been ranked by humans to capture the strength of relatedness. The task is to rank the entities per seed by relatedness.

- Four semantic analogy tasks (e.g., *Athens is to Greece as Oslo is to X*), which are based on the original datasets on which word2vec was evaluated [344].

Task	Dataset	# entities	Target variable
Classification	Cities	212	3 classes (67/106/39)
	AAUP	960	3 classes (236/527/197)
	Forbes	1,585	3 classes (738/781/66)
	Albums	1,600	2 classes (800/800)
	Movies	2,000	2 classes (1,000/1,000)
Regression	Cities	212	numeric [23, 106]
	AAUP	960	numeric [277, 1009]
	Forbes	1,585	numeric [0.0, 416.6]
	Albums	1,600	numeric [15, 97]
	Movies	2,000	numeric [1, 100]
Clustering	Cities and Countries (2k)	4,344	2 clusters (2,000/2,344)
	Cities and Countries	11,182	2 clusters (8,838/2,344)
	Cities, Countries, Albums, Movies, AAUP, Forbes	6,357	5 clusters (2,000/960/1,600/212/1,585)
	Teams	4,206	2 clusters (4,185/21)
Document Similarity	Pairs of 50 documents with entities	1,225	numeric similarity score [1.0,5.0]
Entity relatedness	20x20 entity pairs	400	ranking of entities
Semantic Analogies	(All) capitals and countries	4,523	entity prediction
	Capitals and countries	505	entity prediction
	Cities and States	2,467	entity prediction
	Countries and Currencies	866	entity prediction

Table 7.3: Overview of the Evaluation Datasets

> The datasets were created by manual annotation. In our evaluation, we aim at predicting the fourth element (D) in an analogy $A : B = C : D$ by considering the closest n vectors to $B - A + C$. If the element is contained the top n predictions, we consider the answer correct, i.e., the evaluation metric top-n accuracy. In the default setting of the evaluation framework used, n is set to 2.

Table 7.3 shows a summary of the characteristics of the datasets used in the evaluation. It can be observed that they cover a wide range of tasks, topics, sizes, and other characteristics (e.g., balance). More details on the construction of the datasets can be found in [389] and [443].

Note that all datasets are provided with predefined instance links to DBpedia. For the smaller ones, the creators of the datasets created and checked the links manually; for the larger ones, the linking had been done heuristically. We used the links provided in the evaluation framework as is, including possible linkage errors.

We follow the evaluation protocol suggested in [389]. This protocol foresees the usage of different algorithms on each task for each embedding (e.g., naïve Bayes, Decision Tree, KNN, and *support vector machine* (SVM) for classification), and also performs parameter tuning in some cases. In the end, we report the best results per task and embedding method. Those results are depicted in Table 7.4.

All embeddings are trained on DBpedia 2016-10.[10] For generating the different embedding vectors, we use the DGL-KE framework [614] in the respective standard settings, and we use the RDF2vec vectors provided by the KGvec2go API [404], trained with 500 walks of depth 8 per entity, Skip-Gram, and 200 dimensions. We compare RDF2vec [440], TransE (with L1 and L2 norm) [44], TransR [317], RotatE [516], DistMult [600], RESCAL [370], and ComplEx [538]. To create the embedding vectors with DGL-KE, we use the parameter configurations recommended by the framework, a dimension of 200, and a step maximum of 1,000,000. The RDF2vec$_{oa}$ vectors were generated with the same configuration but using the order-aware variant of skip-gram [319, 412]. For node2vec, DeepWalk, and KGlove, we use the standard settings and the code provided by the respective authors.[11,12,13] For KGlove, we use the Inverse Predicate Frequency, which has been reported to work well on many tasks by the original paper [85].

It is noteworthy that the default settings for node2vec and DeepWalk differ in one crucial property. While node2vec interprets the graph as a directed graph by default and only traverses edges in the direction in which they are defined, DeepWalk treats all edges as undirected, i.e., it traverses them in both directions.

From the table, we can observe a few expected and a few unexpected results. First, since RDF2vec is tailored towards classic data mining tasks like classification and regression, it is not much surprising that those tasks are solved better by using RDF2vec (and even slightly better by using RDF2vec$_{oa}$) vectors. Still, some of the link prediction methods (in particular TransE and RESCAL) perform reasonably well on those tasks. In contrast, KGloVe rarely reaches the performance level of RDF2vec, while the two approaches for unlabeled graphs – i.e., DeepWalk and node2vec – behave differently: while the results of DeepWalk are at the lower end of the spectrum, node2vec is competitive. The latter is remarkable, showing that pure neighborhood information, ignoring the direction and edge labels, can be a strong signal when embedding entities.

Referring back to the different notions of similarity that these families of approaches imply (cf. Section 7.4.3), this behavior can be explained by the tendency of RDF2vec (and also node2vec) to position entities closer in the vector space which are more similar to each other (e.g., two cities that are similar).

[10] The code for the experiments as well as the resulting embeddings can be found at `https://github.com/nheist/KBE-for-Data-Mining`

[11] `https://github.com/D2KLab/entity2vec`

[12] `https://github.com/phanein/deepwalk`

[13] `https://github.com/miselico/globalRDFEmbeddingsISWC`

Since it is likely that some of those dimensions are also correlated with the target variable at hand (in other words: they encode some dimension of similarity that can be used to predict the target variable), classifiers and regressors can pick up on those dimensions and exploit them in their prediction model.

What is also remarkable is the performance on the entity relatedness task. While RDF2vec embeddings, as well as node2vec, KGlove, and, to a lesser extent, DeepWalk, reflect entity relatedness to a certain extent, this is not given for any of the link prediction approaches. According to the notions of similarity discussed above, this is reflected in the RDF2vec mechanism: RDF2vec has an incentive to position two entities closer in the vector space if they share relations to a common entity, as shown in Equations 7.21-7.24. One example is the relatedness of *Apple Inc.* and *Steve Jobs* – here, we can observe the two statements

$$product(AppleInc., iPhone)$$
$$knownfor(SteveJobs, iPhone)$$

in DBpedia, among others. Those lead to similar vectors in RDF2vec according to Equation 7.23. A similar argument can be made for node2vec and DeepWalk, and also for KGlove, which looks at global co-occurrences of entities, i.e., it also favors closer embeddings of related entities.

The same behavior of RDF2vec – i.e., assigning close vectors to *related* entities – also explains the comparatively bad results of RDF2vec on the first two clustering tasks. Here, the task is to separate cities and countries in two clusters, but since a city is also related to the country it is located in, RDF2vec may position that city and country rather closely together (RDF2vec$_{oa}$ changes that behavior, as argued in [412], and hence produces better results for the clustering problems). Hence, that city has a certain probability of ending up in the same cluster as the country. The latter two clustering tasks are different: the third one contains five clusters (cities, albums, movies, universities, and companies), which are less likely to be strongly related (except universities and companies to cities) and therefore are more likely to be projected in different areas in the vector space. Here, the difference between RDF2vec to the best performing approaches (i.e., TransE-L1 and TransE-L2) is not that severe. The same behavior can also be observed for the other embedding approaches for data mining, i.e., node2vec, DeepWalk, and KGlove, which behave similarly in that respect.

The problem of relatedness being mixed with similarity does not occur so strongly for *homogeneous* sets of entities, as in the classification and regression tasks, where all entities are of the same kind (cities, companies, etc.) – here, two companies which are related (e.g., because one is a holding of the other) can also be considered similar to a certain degree (in that case, they are both operating in the same branch). This also explains why the fourth clustering task (where the task is to assign sports teams to clusters by the type of sports) works well for RDF2vec – here, the entities are again homogeneous.

At the same time, the test case of clustering teams can also be used to explain why link prediction approaches work well for that kind of task: here, it is likely that two teams in the same sports share a relation to a common entity, i.e., they fulfill Equations 7.19 and 7.20. Examples include participation in the same tournaments or common former players.

The semantic analogies task also reveals some interesting findings. First, it should be noted that the relations which form the respective analogies (capital, state, and currency) are contained in the knowledge graph used for the computation. That being said, we can see that most of the link prediction results (except for RotatE and RESCAL) perform reasonably well here. Particularly, the first cases (capitals and countries) can be solved particularly well in those cases, as this is a 1:1 relation, which is the case in which link prediction is a fairly simple task. On the other hand, most of the data-mining-centric approaches (i.e., node2vec, DeepWalk, KGlove) solve this problem relatively badly. A possible explanation is that the respective entities belong to the strongly interconnected head entities of the knowledge graphs, and also the false solutions are fairly close to each other in the graph (e.g., US Dollar and Euro are interconnected through various short paths). This makes it hard for approaches concentrating on a common neighborhood to produce decent results here.

On the other hand, the currency case is solved particularly badly by most of the link prediction results. This relation is an n:m relation (there are countries with more than one official, unofficial, or historic currency, and many currencies, like the Euro, are used across many countries. Moreover, looking into DBpedia, this relation contains a lot of mixed usages and is not maintained with very high quality. For example, DBpedia lists 33 entities whose currency is US Dollars[14] – the list contains historic entities (e.g., West Berlin), errors (e.g., Netherlands), and entities which are not countries (e.g., OPEC[15]), but the United States are not among those. For such kinds of relations which contain a certain amount of noise and heterogeneous information, many link prediction approaches are obviously not well suited.

RDF2vec, in contrast, can deal reasonably well with that case. Here, two effects interplay when solving such tasks: (i) as shown above, relations are encoded by the proximity in RDF2vec to a certain extent, i.e., the properties in Equations 7.3 and 7.4 allow to perform analogy reasoning in the RDF2vec space

[14] `http://dbpedia.org/page/United_States_dollar`

[15] *OPEC* stands for “Organization of the Petroleum Exporting Countries”.

Task (Metric)	Dataset	RDF2vec (DM)	RDF2vec$_{OA}$ (DM)	TransE-L1 (LP)	TransE-L2 (LP)	TransR (LP)	RotatE (LP)	DistMult (LP)	RESCAL (LP)	ComplEx (LP)	node2vec (DM)	DeepWalk (DM)	KGloVe (DM)
Classification (ACC)	AAUP	0.676	0.671	0.628	0.651	0.607	0.617	0.597	0.623	0.602	**0.694**	0.549	0.558
	Cities	0.810	**0.837**	0.676	0.752	0.757	0.581	0.666	0.740	0.637	0.774	0.495	0.496
	Forbes	0.610	**0.626**	0.550	0.601	0.561	0.526	0.601	0.563	0.578	0.618	0.490	0.502
	Albums	0.774	0.787	0.637	0.746	0.728	0.550	0.666	0.678	0.693	**0.789**	0.543	0.548
	Movies	0.739	0.736	0.603	0.728	0.715	0.567	0.668	0.693	0.655	**0.763**	0.555	0.563
Clustering (ACC)	Cities and Countries (2K)	0.758	0.931	0.982	**0.994**	0.962	0.510	0.957	0.991	0.955	0.939	0.557	0.623
	Cities and Countries	0.696	0.760	0.953	0.979	0.952	0.691	0.909	**0.990**	0.591	0.743	0.817	0.765
	Cities, Albums, Movies, AAUP, Forbes	0.926	0.928	**0.946**	0.944	0.908	0.860	0.878	0.936	0.914	0.930	0.335	0.525
	Teams	0.917	0.958	0.887	**0.977**	0.844	0.853	0.883	0.881	0.881	0.931	0.830	0.740
Regression (RMSE)	AAUP	68.745	**66.505**	81.503	69.728	88.751	80.177	78.337	72.880	73.665	68.007	103.235	98.794
	Cities	15.601	**13.486**	19.694	14.455	13.558	26.846	19.785	15.137	19.809	15.363	25.323	24.151
	Forbes	36.459	36.124	37.589	38.398	39.803	38.343	38.037	**35.489**	37.877	35.684	41.384	40.141
	Albums	11.930	**11.597**	14.128	12.589	12.789	14.890	13.452	13.537	13.009	15.165	15.129	11.739
	Movies	19.648	**11.739**	23.286	20.635	20.699	23.878	22.161	21.362	22.229	18.877	24.215	24.000
Semantic Analogies (precision@k)	(All) capitals and countries	0.685	0.895	0.709	0.675	**0.938**	0.377	0.782	0.211	0.814	0.284	0.000	0.011
	Capitals and countries	0.648	0.913	0.840	0.792	**0.937**	0.640	0.802	0.312	0.864	0.164	0.000	0.043
	Cities and State	0.342	**0.629**	0.335	0.209	0.392	0.294	0.379	0.089	0.309	0.068	0.000	0.029
	Currency (and Countries)	0.339	**0.427**	0.005	0.285	0.143	0.000	0.001	0.000	0.000	0.420	0.005	0.003
Document Similarity (Harmonic Mean)	LP50	0.348	0.307	0.343	0.397	**0.434**	0.326	0.360	0.344	0.341	0.333	0.243	0.225
Entity Relatedness (KendallTau)	KORE	0.504	**0.779**	0.002	-0.081	0.139	-0.039	0.147	0.087	0.115	0.525	0.129	0.421

Table 7.4
Results of the different data mining tasks. *DM* denotes approaches originally developed for node representation in data mining, *LP* denotes approaches originally developed for link prediction.

in general. Moreover, (ii) we have already seen the tendency of RDF2vec to position *related* entities in relative proximity. Thus, for RDF2vec, it can be assumed that the following holds:

$$UK \approx PoundSterling \quad (7.25)$$

$$USA \approx USDollar \quad (7.26)$$

Since we can rephrase the first equation as

$$PoundSterling - UK \approx 0 \quad (7.27)$$

we can conclude that analogy reasoning in RDF2vec would yield

$$PoundSterling - UK + USA \approx USDollar \quad (7.28)$$

Hence, in RDF2vec, two effects – the preservation of relation vectors as well as the proximity of related entities – are helpful for analogy reasoning, and the two effects also work for rather noisy cases. However, for cases which are 1:1 relations in the knowledge graph with rather clean training data available, link prediction approaches are better suited for analogy reasoning.

7.5.2 Experiments on Link Prediction Tasks

In the second series of experiments, we analyze if we can use embedding methods developed for similarity computation, like RDF2vec, also for link prediction. We use the two established tasks WN18 and FB15k for a comparative study.

While link prediction methods are developed for the task at hand, approaches developed for data mining are not. Although RDF2vec computes vectors for relations, they do not necessarily follow the same notion as relation vectors for link prediction, as discussed above. Hence, we investigate two approaches:

1. We average the difference for each pair of a head and a tail for each relation r, and use that average as a proxy for a relation vector for prediction, as shown in Equation 7.4. The predictions are the entities whose embedding vectors are the closest to the approximate prediction. This method is denoted as *avg*.

2. For predicting the tail of a relation, we train a neural network to predict an embedding vector of the tail-based embedding vectors, as shown in Figure 7.10. The predictions for a triple $< h, r, ? >$ are the entities whose embedding vectors are closest to the predicted vector for h and r. A similar network is trained to predict h from r and t. This method is denoted as *ANN*.

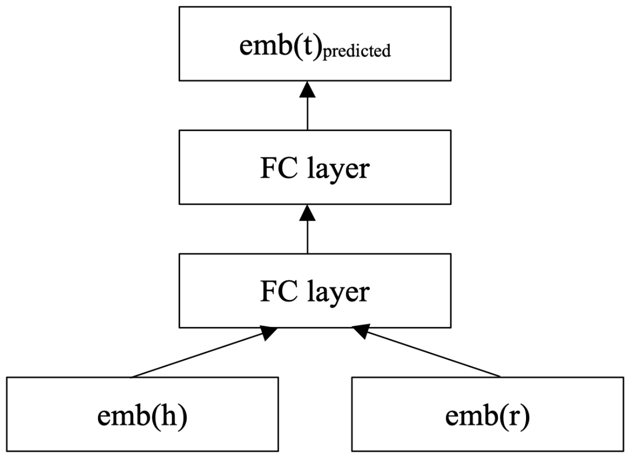


Figure 7.10: Training a Neural Network for Link Prediction With RDF2vec

We trained the RDF2vec embeddings with 2,000 walks, a depth of 4, a dimension of 200, a window of 5, and 25 epochs in SG mode. For the second prediction approach, the two neural networks each use two hidden layers of size 200, and we use 15 epochs, a batch size of 1,000, and mean squared error as loss. KGlove, node2vec, and DeepWalk do not produce any vectors for relations. Hence, we only use the *avg* strategy for those approaches.

The results of the link prediction experiments are shown in Table 7.5.[16] We can observe that the RDF2vec-based approaches perform at the lower end of the spectrum. The avg approach outperforms DistMult and RESCAL on WN18, and both approaches are about en par with RESCAL on FB15k. Except for node2vec on FB15k, the other data mining approaches fail at producing sensible results.

While the results are not overwhelming, they show that the similarity of entities, as RDF2vec models it, is at least a useful signal for implementing a link prediction approach.

7.5.3 Discussion

As already discussed above, the notion of similarity, which is conveyed by RDF2-vec, mixes *similarity* and *relatedness*. This can be observed, e.g., when querying for the 10 closest concepts to *Angela Merkel* (the chancellor, i.e., head of government in Germany) in DBpedia in the different spaces, as shown in Table 7.6. The approach shows a few interesting effects:

[16]The code for the experiments can be found at `https://github.com/janothan/kbc_rdf2vec`

Dataset	Metric	RDF2vec (AVG) (DM)	RDF2vec (ANN) (DM)	RDF2vec$_{OA}$ (AVG) (DM)	RDF2vec$_{OA}$ (ANN) (DM)	TransE (LP)	TransR (LP)	RotatE (LP)	DistMult (LP)	RESCAL (LP)	ComplEx (LP)	node2vec (AVG) DM	DeepWalk (AVG) DM	KGlove (AVG) (DM)
WN18	Mean Rank Raw	147	353	64	77	263	232	-	-	1180	-	17	6112	8247
	Mean Rank Filtered	135	342	53	66	251	219	309	-	1163	-	10	6106	8243
	HITS@10 Raw	64.4	49.7	65.9	66.6	75.4	78.3	-	-	37.2	-	12.3	5.7	1.7
	HITS@10 Filtered	71.3	55.4	73.0	75.08	89.2	91.7	**95.9**	57.7	52.8	94.7	14.2	6.0	1.7
FB15K	Mean Rank Raw	399	349	168	443	243	226	-	-	828	-	192	2985	2123
	Mean Rank Filtered	347	303	120	90	125	78	40	-	683	-	138	2939	2077
	HITS@10 Raw	35.3	34.3	47.9	30.9	34.9	43.8	-	-	28.4	-	44.7	7.2	11.1
	HITS@10 Filtered	40.5	41.8	56.5	37.4	47.1	65.5	88.4	**94.2**	44.1	84.0	53.3	7.7	11.1

Table 7.5: Results of the link prediction tasks on WN18 and FB15K. Results for TransE and RESCAL from [44], results for RotatE from [516], results for DistMult from [600], results for TransR from [317]. *DM* denotes approaches originally developed for node representation in data mining, *LP* denotes approaches originally developed for link prediction.

RDF2vec	TransE-L1	TransE-L2	TransR
Joachim Gauck	Gerhard Schröder	Gerhard Schröder	Sigmar Gabriel
Norbert Lammert	James Buchanan	Helmut Kohl	Frank-Walter Steinmeier
Stanislaw Tillich	Neil Kinnock	Konrad Adenauer	Philipp Rösler
Andreas Voßkuhle	Nicolas Sarkozy	Helmut Schmidt	Gerhard Schröder
Berlin	Joachim Gauck	Werner Faymann	Joachim Gauck
German language	Jacques Chirac	Alfred Gusenbauer	Christian Wulff
Germany	Jürgen Trittin	Kurt Georg Kiesinger	Guido Westerwelle
federalState	Sigmar Gabriel	Philipp Scheidemann	Helmut Kohl
Social Democratic Party	Guido Westerwelle	Ludwig Erhard	Jürgen Trittin
deputy	Christian Wulff	Wilhelm Marx	Jens Böhrnsen

RotatE	DistMult	RESCAL	ComplEx
Pontine raphe nucleus	Gerhard Schröder	Gerhard Schröder	Gerhard Schröder
Jonathan W. Bailey	Milan Truban	Kurt Georg Kiesinger	Diána Mészáros
Zokwang Trading	Maud Cuney Hare	Helmut Kohl	Francis M. Bator
Steven Hill	Tristan Matthiae	Annemarie Huber-Hotz	William B. Bridges
Chad Kreuter	Gerda Hasselfeldt	Wang Zhaoguo	Mette Vestergaard
Fred Hibbard	Faustino Sainz Muñoz	Franz Vranitzky	Ivan Rosenqvist
Mallory Ervin	Joachim Gauck	Bogdan Klich	Edward Clouston
Paulinho Kobayashi	Carsten Linnemann	İrsen Küçük	Antonio Capuzzi
Fullmetal Alchemist and the Broken Angel	Norbert Blüm	Helmut Schmidt	Steven J. McAuliffe
Archbishop Dorotheus of Athens	Neil Hood	Mao Zedong	Jenkin Coles

KGloVe	RDF2vec OA	node2vec	DeepWalk
Aurora Memorial National Park	Joachim Gauck	Sigmar Gabriel	Manuela Schwesig
Lithuanian Wikipedia	Norbert Lammert	Guido Westerwelle	Irwin Fridovich
Baltic states	Stanislaw Tillich	Christian Wulff	Holstein Kiel Dominik Schmidt
The Monarch (production team)	Andreas Voßkuhle	Jürgen Trittin	Ella Germein
Leeds Ladies F.C. Lauryn Colman	Berlin	Wolfgang Schäuble	Goyang Citizen FC Do Sang-Jin
Steven Marković	German language	Joachim Gauck	Sean Cashman
Funk This (George Porter Jr. album)	Germany	Philipp Rösler	Chia Chiao
A Perfect Match (Ella Fitzgeral album)	Christian Wulff	Joachim Sauer	Albrix Niigata Goson Sakai
Salty liquorice	Gerhard Schröder	Franz Müntefering	Roz Kelly
WMMU-FM	federalState	Frank-Walter Steinmeier	Alberto Penny

Table 7.6: Closest concepts to *Angela Merkel* in the different embedding approaches used.

- While most of the approaches (except for RotatE, KGlove and DeepWalk) provide a clean list of people, RDF2vec brings up a larger variety of results, containing also *Germany* and *Berlin* (and also a few results which are not instances, but relations; however, those could be filtered out easily in downstream applications if necessary). This demonstrates the property of RDF2vec of mixing *similarity* and *relatedness*. The people in the RDF2vec result set are all related to Angela Merkel: Joachim Gauck was president during her chancellorship, Norbert Lammert was the head of parliament, Stanislaw Tillich was a leading board member in the same party as Merkel, and Andreas Voßkuhle was the head of the highest court during her chancellorship.

- The approaches at hand have different foci in determining similarity. For example, TransE-L1 outputs mostly German politicians (Schröder, Gauck, Trittin, Gabriel, Westerwelle, Wulff) and former presidents of other countries (Buchanan as a former US president, Sarkozy and Chirac as former French presidents) TransE-L2 outputs a list containing many former German chancellors (Schröder, Kohl, Adenauer, Schmidt, Kiesinger, Erhardt), TransR mostly lists German political party leaders (Gabriel, Steinmeier, Rösler, Schröder, Wulff, Westerwelle, Kohl, Trittin). Likewise, node2vec produces a list of German politicians, with the exception of Merkel's husband Joachim Sauer.[17] In all of those cases, the persons share some property with the query entity *Angela Merkel* (profession, role, nationality, etc.), but the similarity is usually affected only by one of those properties. In other words: one notion of similarity *dominates* the others.

- In contrast, the persons in the output list of RDF2vec are *related to the query entity* in *different* respects. In particular, they played different roles during Angela Merkel's chancellorship (Gauck was the German president, Lammert was the chairman of the parliament, and Voßkuhle was the chairman of the federal court). Here, there is no *dominant* property, instead, similarity (or rather: relatedness) is encoded along with various properties. RDF2vec$_{oa}$ brings up a result which is slightly closer to the politicians lists of the other approaches, while the result list of KGlove looks more like a random list of entities. A similar observation can be made for DeepWalk, which, with the exception of the first result (which is a German politician)

[17] The remaining approaches – RotatE, DistMult, RESCAL, ComplEx, KGlove, DeepWalk – produce lists of (mostly) persons which, in their majority, share no close link to the query concept Angela Merkel.

does not produce any results seemingly related to the query concept at hand.

With that observation in mind, we can come up with an initial set of recommendations for choosing embedding approaches:

- Approaches for data mining (RDF2vec, KGlove, node2vec, and DeepWalk) work well when dealing with sets of *homogeneous entities*. Here, the problem of confusing related entities (like Merkel and Berlin) is negligible, because all entities are of the same kind anyways. In those cases, RDF2vec captures the finer distinctions between the entities better than embeddings for link prediction, and it encodes a larger variety of semantic relations.
- From the approaches for data mining, those which respect the order (RDF2vec$_{oa}$ and node2vec) work better than those which do not (classic RDF2vec, KGlove, and DeepWalk).[18]
- For problems where *heterogeneous sets of entities* are involved, embeddings for link prediction often do a better job in telling different entities apart.

Link prediction is a problem of the latter kind: in embedding spaces where different types are properly separated, link prediction mistakes are much rarer. Given an embedding space where entities of the same type are always closer than entities of a different type, a link prediction approach will always rank all "compatible" entities higher than all incompatible ones. Consider the following example in FB15k:

$$instrument(GilScottHeron, ?)$$

Here, music instruments are expected in the object position. Nonetheless, approaches tailored towards capturing node similarity, e.g., classic RDF2vec, will suggest among plausible candidates such as *electric guitar* and *acoustic guitar*, also *guitarist* and *Jimmy Page* (who is a well-known guitarist). While *electric guitar*, *guitarist*, and *Jimmy Page* are semantically related, not all of them are sensible predictions here, and the fact that RDF2vec reflects that semantic relatedness is a drawback in link prediction.

The same argument underlies an observation made by Zouaq and Martel [618]: the authors found that RDF2vec is particularly well suited for distinguishing fine-grained entity classes (as opposed to coarse-grained entity classification). For fine-grained classification (e.g., distinguishing guitar players from

[18] As discussed above, this comments holds for the *default* configuration of node2vec and DeepWalk used in this chapter.

singers), all entities to be classified are already of the same coarse class (e.g., musician), and RDF2vec is very well suited for capturing the finer differences. However, for coarse classifications, misclassifications by mistaking relatedness for similarity become more salient.

From the observations made in the link prediction task, we can come up with another recommendation:

- For relations which come with rather clean data quality, link prediction approaches work well. However, for more noisy data, RDF2vec has a higher tendency of creating useful embedding vectors.

For the moment, this is a hypothesis, which should be hardened, e.g., by performing controlled experiments on artificially noised link prediction tasks.

7.6 Conclusion

In this chapter, we have compared two use cases and families of knowledge graph embeddings which have, up to today, not undergone any thorough direct comparison: approaches developed for data mining, such as RDF2vec, and approaches developed for link prediction, such as TransE and its descendants.

We have argued that the two approaches actually do something similar, albeit being designed with different goals in mind. To support this argument, we have run two sets of experiments which examined how well the different approaches work if applied in the respective other setup. We show that, to a certain extent, embedding approaches designed for link prediction can be applied in data mining and vice versa, however, there are differences in the outcome.

From the experiments, we have also seen that proximity in the embedding spaces works differently for the two families of approaches: in RDF2vec, proximity encodes both similarity and relatedness, while TransE and its descendants rather encode similarity alone. On the other hand, for entities that are of the same type, RDF2vec covers finer-grained similarities better. Moreover, RDF2vec seems to work more stably in cases where the knowledge graphs are rather noisy and weakly adherent to their schema.

These findings give rise both to a recommendation and some future work. First, in use cases where relatedness plays a role next to similarity, or in use cases where all entities are of the same type, approaches like RDF2vec may yield better results. On the other hand, for cases with mixed entity types where it is important to separate the types, link prediction embeddings might yield better results.

Chapter 8

KGvec2go – Knowledge Graph Embeddings as a Service

In this chapter, we present KGvec2go, a Web API for accessing and consuming graph embeddings in a lightweight fashion in downstream applications. Currently, we serve pre-trained embeddings for four knowledge graphs. We introduce the service and its usage, and we show further that the trained models have semantic value by evaluating them on multiple semantic benchmarks. The evaluation also reveals that the combination of multiple models can lead to a better outcome than the best individual model.

Over the course of this dissertation, the service has been continuously improved. The download section[1] of the service, for instance, currently hosts more than 20 embedding models for very large knowledge graphs.

The work presented in this chapter has been published before as: Portisch, Jan; Hladik, Michael; Paulheim, Heiko. KGvec2go – Knowledge Graph Embeddings as a Service. In: Language Resources and Evaluation Conference (LREC). 2020. [404]

8.1 Introduction

A *knowledge graph* (KG) stores factual information in the form of triples. Today, many such graphs exist for various domains, are publicly available, and are being interlinked. As of 2019, the *linked open data cloud* [465] counts more than

[1] see `http://kgvec2go.org/download.html`

1,000 datasets with multiple billions of unique triples.[2] Knowledge graphs are typically consumed using factual queries for downstream tasks such as question answering. Recently, knowledge graph embedding models are explored as a new way of knowledge graph exploitation. *Knowledge graph embeddings* (KGEs) represent nodes and (depending on the approach) also edges as continuous vectors. One such approach is *RDF2vec* [440]. It has been used and evaluated for machine learning, entity and document modeling, and for recommender systems [442]. RDF2vec vectors trained on a large knowledge graph have also been used as a background knowledge source for ontology matching [409].

While it has been shown that KGEs are helpful in many applications, embeddings on larger knowledge graphs can be expensive to train and to use for downstream applications. `kgvec2go.org`, therefore, allows to easily access and consume concept embeddings through simple Web APIs. Since most downstream applications only require embedding vectors for a small subset of all concepts, computing a complete embedding model or downloading a complete pre-computed one is often not desirable.

With *KGvec2go*, rather than having to download the complete embedding model, a Web query can be used to obtain only the desired concept in vector representation or even a derived statistic such as the similarity between two concepts. This facilitates downstream applications on less powerful devices, such as smartphones, as well as the application of knowledge graph embeddings in machine learning scenarios where the data scientists do not want to train the models themselves or do not have the means to perform the computations.

Models for four knowledge graphs were learned, namely: *DBpedia* [300], *WebIsALOD* [198], *Wiktionary* [470], and *WordNet* [149].
The dataset presented here allows comparing the performance of different knowledge graph embeddings on different application tasks. It further allows combining embeddings from different knowledge graphs in downstream applications. We evaluated the embeddings on three semantic gold standards and also explored the combination of embeddings.
This chapter is structured as follows: In the next section, related work will be presented. Section 8.2 outlines the approach, Section 8.3 presents the datasets for which an embedding has been trained, Section 8.4 introduces the Web API that is provided to consume the learned embedding models, and Section 8.5 evaluates the models on three semantic gold standards. The chapter closes with a summary.

[2] `https://lod-cloud.net/`

8.2 Approach

For this work, the RDF2vec approach has been re-implemented in Java and Python with a more efficient walk generation process. The implementation of the walk generator is publicly available on GitHub[3].
For the sentence generation, duplicate free random walks with *depth = 8* have been generated whereat edges within the sentences are also counted. For *WordNet* and *Wiktionary*, 500 walks have been calculated per entity. For *WebIsALOD* and *DBpedia*, 100 walks have been created in order to account for the comparatively large size of the knowledge graphs.
The models were trained with the following configuration: *skip-gram vectors*, *window size = 5*, *number of iterations = 5*, *negative sampling for optimization*, and *negative samples = 25*. Apart from walk-generation adaptations due to the size of the knowledge graphs, the configuration parameters to train the models have been held constant, and no dataset-specific optimizations have been performed in order to allow for comparability.
In addition, a Web API is provided to access the data models in a lightweight way. This allows for easy access to embedding models and to bring powerful embedding models to devices with restrictions in their *central processing unit* (CPU) and *random-access memory* (RAM), such as smartphones. The APIs are introduced in Section 8.4 The server has been implemented in Python using *flask*[4] and *gensim* [431] and can be run using *Apache HTTP Server*. Its code is publicly available on GitHub.[5]

8.3 The Datasets

For this work, four datasets have been embedded, which are quickly introduced in the following.

8.3.1 DBnary/Wiktionary

Wiktionary is "[a] collaborative project run by the Wikimedia Foundation to produce a free and complete dictionary in every language"[6]. The project is organized similarly to Wikipedia: Everybody can contribute and edit the dictionary.

[3] `https://github.com/janothan/kgvec2go-walks/`

[4] `https://flask.palletsprojects.com/en/1.1.x/`

[5] `https://github.com/janothan/kgvec2go-server/`

[6] `https://web.archive.org/web/20190806080601/https://en.wiktionary.org/wiki/Wiktionary`

The content is reviewed in a community process. Like Wikipedia, Wiktionary is available in many languages. *DBnary* [470] is an RDF version of Wiktionary that is publicly available[7]. The DBnary dataset makes use of an extended LEMON[8] model [335] to describe the data. For this work, a recent download from July 2019 of the English Wiktionary has been used.

8.3.2 DBpedia

DBpedia is a well-known linked dataset created by extracting structured knowledge from Wikipedia and other Wikimedia projects. The data is publicly available. For this work, the 2016-10 download has been used.[9] Compared to the other knowledge graphs exploited here, DBpedia contains mainly instances such as the industrial rock band *Nine Inch Nails* (which cannot be found in WordNet or Wiktionary). Therefore, DBpedia is with its instance data complementary to the other, lemma-focused knowledge graphs.

8.3.3 WebIsALOD

The *WebIsA* database [467] is a dataset which consists of hypernymy relations extracted from the *Common Crawl*[10], a downloadable copy of the Web. The extraction was performed in an automatic manner through Hearst-like lexico-syntactic patterns. For example, from the sentence "[...] added that the country has favorable economic agreements with major economic powers, including the European Union.", the fact `isA(european_union, major_economic_power)` is extracted[11].

WebIsALOD [198] is the Linked Open Data endpoint which allows querying the data in SPARQL.[12] In addition to the endpoint, machine learning was used to assign confidence scores to the extracted triples. The dataset of the endpoint is filtered, i.e., it contains a subset of the original WebIsA database, to ensure higher data quality. The knowledge graph contains instances (like DBpedia) as well as more abstract concepts that can also be found in a dictionary.

[7] `http://kaiko.getalp.org/about-dbnary/download/`

[8] LEMON stands for “Lexicon Model for Ontologies”.

[9] `https://wiki.dbpedia.org/downloads-2016-10`

[10] `https://commoncrawl.org/`

[11] This is a real example, see: `http://webisa.webdatacommons.org/417880315`

[12] `http://webisa.webdatacommons.org/`

8.3.4 WordNet

WordNet [149] is a well-known and heavily used database of English words that are grouped in sets which represent one particular meaning, so-called *synsets*. The resource is strictly authored. WordNet is publicly available, included in many natural language processing frameworks, and often used in research. An RDF version of the framework is also available for download and was used for this work.[13]

8.4 API

kgvec2go.org offers a simple Web API to retrieve: (i) individual vectors for concepts in different datasets, (ii) the cosine similarity between concepts directly, and (iii) the top *n* most related concepts for any given concept. Alternatively, the full models can be downloaded from the Web site directly.[14] The API is accessed through HTTP GET calls and will provide answers in the form of a *JavaScript Object Notation* (JSON) string. This allows for simple usage on any device that has Internet access. In addition, natural words can be used to access the data rather than long URIs that follow their own idiosyncratic pattern as is common for RDF2vec embedded models. In the following, we will quickly describe the services that are offered. For a full description of the services as well as a graphical user interface to explore the embeddings, we refer to the Web page kgvec2go.org.

8.4.1 Get Vector

kgvec2go.org allows to download an individual vector, i.e. a 200 dimensional floating point number array representation of a concept on a particular dataset. The HTTP GET call follows the pattern below: /rest/get-vector/<data_set>/
/<concept_name>
where data_set refers to the dataset that shall be used (i.e. one of alod, dbpedia, wiktionary, wordnet) and concept_name to the natural language identifier of the concept (e.g. *bed*). This call can be used in machine learning scenarios, for instance, where a numerical representation of a concept is required.
For datasets that learn an embedding based on the *part of speech* (POS) of the term, such as WordNet, multiple vectors are returned for one keyword if the latter is available in multiple POS such as *laugh*, which occurs as a noun and as a verb.

[13] http://wordnet-rdf.princeton.edu/about/
[14] http://www.kgvec2go.org/download.html

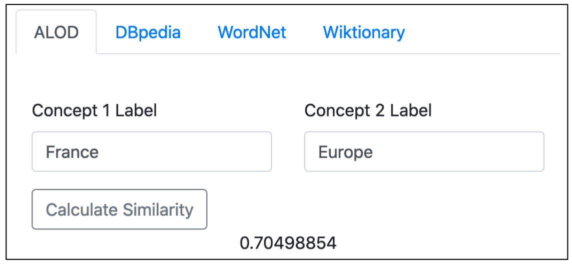


Figure 8.1: UI to query the similarity of two concepts online. Depicted is the similarity between *France* and *Europe* using the WebIsALOD embeddings.

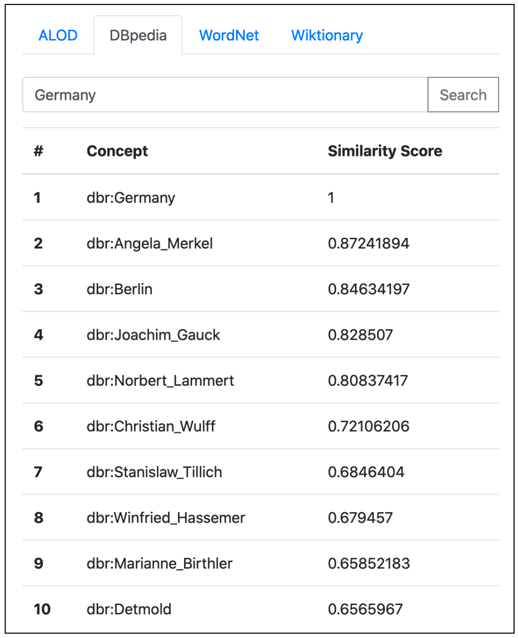
8.4.2 Get Similarity

Given two concepts, `kgvec2go.org` allows to query a specified dataset for the similarity score $s \in [-1.0, 1.0]$ where 1.0 refers to perfect similarity. The HTTP GET call follows the pattern below: `/rest/get-similarity/<data_set>/<concept_name_1>/<concept_name_2>`
where `data_set` refers to the set that shall be used, and the two concept names refer to the concept labels for which the similarity shall be calculated. This call can be used wherever the similarity or relatedness of two concepts needs to be judged, such as in recommender systems or matching tasks. A Web UI is available to try out this call in a Web browser.[15] A screenshot is shown in Figure 8.1 for the terms *France* and *Europe* for the model learned on WebIsALOD.

8.4.3 Get Closest Concepts

The API is also capable of determining the closest n concepts given a concept and a dataset. The given concept is mapped to the vector space and compared with all other vectors. Therefore, the call is expensive on large datasets and should rather be used to explore the dataset. The HTTP GET call follows the pattern below: `/rest/closest-concepts/<data_set>/<top_n>/<concept_name>`
where `data_set` refers to the set that shall be used, `top_n` refers to the number of closest concepts that shall be obtained, and `concept_name` refers to the written representation of the concept. For datasets that learn an embedding based on the part of speech of the term, such as WordNet, all closest concepts are determined for all POS of the term, and their scores are summarized. This allows to calculate the n closest concepts for a single term, such as *sleep*, that occurs in

[15] `http://www.kgvec2go.org/query.html`



#	Concept	Similarity Score
1	dbr:Germany	1
2	dbr:Angela_Merkel	0.87241894
3	dbr:Berlin	0.84634197
4	dbr:Joachim_Gauck	0.828507
5	dbr:Norbert_Lammert	0.80837417
6	dbr:Christian_Wulff	0.72106206
7	dbr:Stanislaw_Tillich	0.6846404
8	dbr:Winfried_Hassemer	0.679457
9	dbr:Marianne_Birthler	0.65852183
10	dbr:Detmold	0.6565967

Figure 8.2: UI to Query the Dataset Online. Shown is the result for query term *Germany* on dataset DBpedia. Note that the underlying DBpedia version for the training is that of 2016. In that year, Angela Merkel was the Chancellor of Germany, Berlin the capital of the country, Joachim Gauck the President of Germany, and Norbert Lammert the President of the Bundestag.

multiple POS (in this case, as a noun and as a verb). A Web UI is available to try out this call in a Web browser.[16] A screenshot is shown in Figure 8.2 for the term *Germany* on the trained DBpedia model.

8.5 Evaluation

8.5.1 Evaluation Gold Standards

In order to test whether there is semantic value in the trained vectors, we evaluate them on three datasets: *WordSim-353* [154], *SimLex-999* [208], and *MEN* [53]. The principle of evaluation is the same for all gold standards used: The system is presented with two words and has to determine their relatedness or similarity; then, the rank correlation (also known as *Spearman's Rho*) with the scores

[16] `http://www.kgvec2go.org/query.html`

in the gold standards is calculated. Higher correlations between the gold standards' scores and the system's scores are regarded as better. Pairs with an out-of-vocabulary term are handled here by returning a similarity of 0. As the goal of this dataset are comparable general-purpose embeddings, it is important to note that the embeddings were not specifically trained to perform well on the given tasks. On similarity tasks, for instance, the results would likely improve when antonymy relations were dropped. With other configuration settings, it is also possible to improve the results further on the given evaluation sets; this has, for instance, been done in [417] where better relatedness/similarity results on WebIsALOD could be achieved with other RDF2vec configurations.

8.5.2 Evaluation Mode

The learned models were evaluated on their own on each of the evaluation datasets. In addition, a combination of all datasets was evaluated. Therefore, the individual similarity scores were added. Hence, $s_{combined}(c_1, c_2) = s_{DBpedia}(c_1, c_2) + s_{WebIsALOD}(c_1, c_2) + s_{Wiktionary}(c_1, c_2) + s_{WordNet}(c_1, c_2)$ where $s_{combined}$ is the final similarity score assigned to the concept pair c_1 and c_2 and $s_{dataset}$ describes the individual score of a model trained on a single *dataset* for the same concept pair. This can be done without normalization because (i) all scores are in the same value range ($[-1, 1]$), (ii) out of vocabulary terms receive a score of 0 (so they do not influence the final results), and (iii) because Spearman's rank correlation is used which is independent of the absolute values – only the rank is considered.

8.5.3 Evaluation Results

The rank correlations on the three gold standards are summarized in Table 8.1. It can be seen that the results vary depending on the gold standard used. The Wiktionary dataset performs best when it comes to relatedness. The WebIsALOD dataset performs similarly well on WS-353 and performs best on MEN. On the SimLex-999 gold standard, WordNet outperforms the other datasets. The performance of DBpedia is significantly worse, which is due to many out-of-vocabulary terms: This particular dataset is focused on instance data rather than lexical forms such as *angry*. The evaluation performed here is, therefore, not optimal for the dataset. This can also be observed in the example results depicted in Table 8.2: While DBpedia and WebIsALOD work well for entities such as *Germany*, Wiktionary performs better for general words such as *loud*.

Interestingly, the combined evaluation mode outlined in Subsection 8.5.2 is able to outperform the best individual results on WS-353 ($\rho = 0.678$ vs. $\rho =$

0.571) as well as on MEN ($\rho = 0.230$ vs. $\rho = 0.207$). On SimLex, the combination of all similarity scores is very close to the best individual score (WordNet). This shows that it can be beneficial to combine several embedding spaces on different datasets.
It is important to note that the vectors were not trained for the specific task at hand. Nonetheless, the combined embeddings perform well on WS-353, albeit top-notch systems for each dataset cannot be outperformed. By the lower performance on SimLex-999 and MEN, it can be seen that relatedness is better represented in the embedding spaces than actual similarity. This is an intuitive result given that there was no training objective towards similarity.
When looking at the different properties of the knowledge graphs, it can be reasoned that the level of authoring is not important for the performance of the tasks at hand: WebIsALOD embeddings, which are derived from an automatically created knowledge graph, easily outperform WordNet embeddings, which are derived from a highly authored knowledge base, on WS-353 and MEN.

8.5.4 Further Remarks

It is also possible to find typical analogies in the data. In this case, two concepts are presented to the model together with a third one, for which the system shall determine an analogous concept. In the following examples, the underlined concept is the best concept that the system found given the three non-underlined concepts.

For example, on Wiktionary:

- *girl* is to *boy* like *man* is to <u>*woman*</u>
- *big* is to *small* like *fake* is to <u>*original*</u>
- *beautiful* is to *attractive* like *quick* is to <u>*rapid*</u>

Similar results can be found on the instance level. For example, on DBpedia:

- *Germany* is to *Angela Merkel* like *France* is to <u>*François Hollande*</u>[17]

8.6 Conclusion

In this chapter, we presented `KGvec2go`, a resource consisting of trained embedding models on four knowledge graphs. The models were evaluated on three

[17] Note that François Hollande is indeed the president of France as of 2016.

	WS-353	**SimLex-999**	**MEN**
Wiktionary	0.5708	0.2265	0.1513
DBpedia	0.1430	-0.0097	0.0804
WebIsALOD	0.5598	0.1509	0.2066
WordNet	0.4074	0.2870	0.1086
Combined	0.6784	0.2815	0.2304

Table 8.1: Rank Correlation of the Datasets with Three Gold Standards

	Wiktionary	DBpedia	WebIsALOD	WordNet
1	Germany	Germany	europe	Germany
2	snazziness	Angela Merkel	uk	FRG
3	West Germany	Berlin	france	skillet
4	these islands	Joachim Gauck	canada	Federal Republic of Germany
5	cobbler	Norbert Lammert	japan	Deutschland
6	German Empire	Christian Wulff	italy	High German
7	derisive	Stanislaw Tillich	australia	German
8	who shot John	Winfried Hassemer	usa	Pietism
9	glute	Marianne Birthler	england	Bavaria
10	Okla.	Detmold	asia	ingrained
1	loud	Loud	cons fan	loud (s)
2	silent	Loli	scream	secondly
3	noiseless	Cometa (HVDC)	weird noise	loud (r)
4	rackety	Looc	of noise	aright
5	noisy	Loob	history of 20th century	loud (a)
6	unsilent	Python Server Pages	collective sigh of relief	fruticulose
7	piercing	Louk	thwack	red-handed
8	quiet	Juan Llort	undesired signal	deep down
9	clamorous	Lojo	grinning	every bit
10	blasting	Lone	complaint of office worker	rhymeless

Table 8.2: Example results for the search terms *Germany* (upper part) and *loud* (lower part). WordNet returns *loud* multiple times with different part-of-speech tags. On DBpedia, results for *Loud* are given as there is no vector for *loud*.

different gold standards. It could be shown that the trained vectors carry semantic meaning and that a combination of different knowledge graph embeddings can be beneficial in some tasks. Furthermore, a lightweight API was presented, which allows consuming the models in a computationally cheap, memory-efficient, and easy way through Web APIs. We are confident that our work eases the usage of knowledge graph embeddings in real-world applications.

Chapter 9

RDF2vec Light

In this chapter, a new, lightweight, RDF2vec-based, approach for knowledge graph embeddings is presented. It is evaluated on three machine learning and retrieval tasks, and the performance is compared with the classic RDF2vec approach. It is shown that the new approach requires only a fraction of the computing power compared to the original approach while maintaining similar performance. Moreover, it is shown that RDF2vec Light does not lose performance when reducing the dimensionality of the vector space.

As an additional contribution, the first version of the jRDF2vec framework is introduced in this chapter (Section 9.5). The framework is used (and extended) for all RDF2vec training operations in this dissertation. Over time, it also gained third-party usage.

Parts of the work presented in this chapter have been published before as: Portisch, Jan; Hladik, Michael; Paulheim, Heiko. RDF2Vec Light - A Lightweight Approach for Knowledge Graph Embeddings. International Semantic Web Conference (ISWC) 2020, Posters and Demonstrations Track. 2020. [405]

9.1 Approach

In the original RDF2vec approach, vectors are trained for each node in a knowledge graph. Since large-scale knowledge graphs are very diverse, this also means that for a specific task at hand, the vast majority of those embeddings vectors are not required. As an example, one task in the aforementioned evaluation framework is to predict the quality of living in a dataset of cities. For that task, embeddings for bands, artists, and songs are rather irrelevant. Knowing the exact embedding vector for a band like *Nine Inch Nails* will not have an impact on the

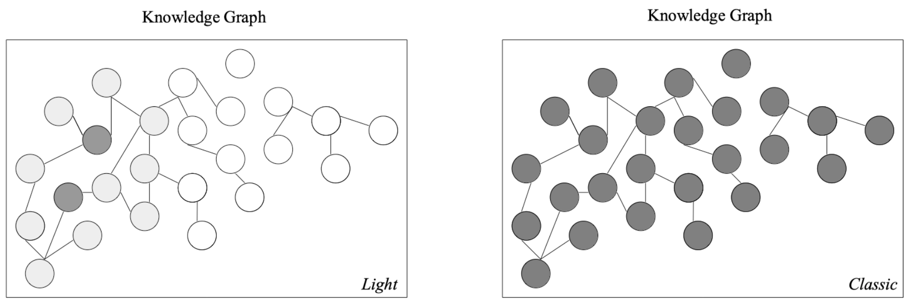


Figure 9.1: Exemplary comparison of the approaches: *RDF2vec Light* (on the left) creates walks that only involve entities of interest (dark gray). As a result, the walks only include the gray entities; the nodes that appeared in the walks, i.e. the context of the entities of interest, are colored in light gray. *RDF2vec Classic* (on the right) generates walks for all nodes.

predictive power in this task. At the same time, the original RDF2vec approach in principle uses every statement in the knowledge graph. Along the same lines as above, the statement that *Trent Reznor* is the writer of the song *Closer* will not be helpful for predicting the quality of living in different cities.

The underlying idea of *RDF2Vec Light* embeddings is to generate only local walks for entities of interest given a predefined task. For example, when the rating average of music albums on a Web site shall be regressed, walks would only be generated that involve the entities in focus (music albums). Thereby, the context of the entities can be captured. This is depicted in Figure 9.1. In order to better capture an entity's context, the walk generation algorithm has been optimized and is further explored in Subsection 9.2.

After the walk generation has been completed, the training of vectors can be performed like in the original approach. The lightweight approach requires only a fraction of the computing capabilities compared to the full training and walk generation and can be performed on consumer hardware. On smaller tasks, RDF2vec Light can be trained during the application runtime rather than being pre-trained in advance.

9.2 Walk Generation Algorithm

A walk or sentence in *RDF2vec Light* (like in *RDF2vec Classic*) consists not just of a sequence of nodes but also contains the predicates. A valid sentence of depth 1, for example, would be: *dbr:Cambridge* → *dbo:country* → *dbr:United_Kingdom*. Similarly, an example for a sentence of depth 2 would be: *dbr:University_of-*

_Cambridge → dbo:city → dbr:Cambridge → dbo:country → dbr:United_Kingdom. The walk generation ignores datatype properties. The original RDF2vec approach generates walks for each vertex $v \in V$ where a sentence for a specific vertex v_s always starts with v_s. This walk generation pattern is insufficient for local walks of only few vertices because the context is not fully reflected: There is a bias towards facts where v_s is the subject of a statement – whereas facts, where v_s appears as an object, would only occur, if another entity of interest has a path that passes v_s.[1] Therefore, the walk generation has been adapted: Rather than performing random walks where the entity of interest is always at the start of a sentence, it is randomly decided for each depth-iteration whether to go backward, i.e., to one of the node's predecessors, or forwards, i.e., to the node's successors (line 11 of Algorithm 1). The probability of continuing the walk in the backward or forward direction is proportional to the number of available options to do so (lines 12 – 15 of Algorithm 1). For example, when there are 9 options to continue the walk in forward direction and one option to continue the walk in backward direction, the walk will be continued in backward direction with a probability of $1/(1+9) = 10\%$. Predecessors are added at the beginning of the walk (line 13 of Algorithm 1) and successors at the end of the walk (line 16 of Algorithm 1). Consequently, the entity of interest can be at the beginning, at the end, or in the middle of a walk which better captures the context of the entity. This generation process is described in Algorithm 1.

9.3 Evaluation

In order to evaluate the approach presented in this chapter, the classification, regression, document similarity, and entity relatedness experiments of Ristoski et al. [442] have been repeated. They are quickly introduced in the following subsections together with the experimental setup. The results are presented in Section 9.4.

9.3.1 Classification and Regression Tasks

For the classification/regression evaluation, five default machine learning datasets for the Semantic Web are used [443]: (i) *Cities*, (ii) *Metacritic Movies*, (iii) *Metacritic Albums*, (iv) *AAUP*, and (v) *Forbes*. They consist of links in the form

[1] Note that this bias occurs only if walks are generated for a subset of V – the traditional RDF2vec approach is, consequently, balanced.

Algorithm 1 Walk Generation Algorithm for *RDF2vec Light*

procedure GENERATELIGHTWALKS(V_I: vertices of interest, d: walk depth, n: number of walks)
  $W_G \leftarrow \emptyset$
  for vertex v $\in V_I$ **do**
    for 1 to n **do**
      initialize w
      add v to w
      $pred \leftarrow$ getIngoingEdges(v)
      $succ \leftarrow$ getOutgoingEdges(v)
      while w.length() < d **do**
        $cand \leftarrow pred \cup succ$
        $elem \leftarrow$ pickRandomElementFrom($cand$)
        if $elem \in pred$ **then**
          add $elem$ at the beginning of w
          $pred \leftarrow$ getIngoingEdges($elem$)
        else
          add $elem$ at the end of w
          $succ \leftarrow$ getOutgoingEdges($elem$)
        end if
      end while
      add w to W_G
    end for
  end for
  return W_G
end procedure

of URIs to DBpedia[2] and target variables. Task (i) contains a list of cities and a score describing the quality of living as the target variable; task (ii) consists of movies together with their average rating as the target variable from metacritic.com; similarly, task (iii) consists of album links and their average rating from the same Web site; task (iv) contains a list of universities and the average salary paid there as target variable; lastly, task (v) contains companies and their market value as target variable from Forbes as of 2015. For the classification tasks, discretization has been applied. The classification tasks are evaluated us-

[2]The URIs refer to DBpedia 2015-10. Here, DBpedia 2016-10 is used, which leads in some cases to missing URIs due to changes in the knowledge graph. Such instances are ignored in the evaluation.

ing accuracy, the regression tasks are evaluated using *root mean squared error* (RMSE).

9.3.2 Document Similarity Task

For the document similarity evaluation, an adapted version of the *LP50* [299] dataset is used where documents are represented as a set of DBpedia links. It consists of pairwise annotated similarity ratings of documents on a Likert scale ranging from 1 to 5. The similarity here is obtained by determining the maximal similarity in a pairwise comparison of entities within the documents to be compared. The dataset is evaluated using Pearson's and Spearman's Rho. In order to obtain a single score, their harmonic mean is additionally calculated.

9.3.3 Entity Relatedness Task

For the entity relatedness task, the *KORE50* [511] dataset is used. It consists of 420 pairwise entity relation scores where each entity is represented as a DBpedia URI. The dataset is evaluated using Kendall's Tau.

9.3.4 Experimental Setup

Six classic and six light embedding spaces have been trained each with the following parameters held constant: $window\ size = 5$, $negative\ samples = 25$. The parameters that were changed are the generation mode (CBOW and SG) as well as the dimension of the embedding space (50, 100, 200). All walks have been generated with 500 walks per entity and a depth of 4. For the evaluation, the DBpedia knowledge graph as of 2016-10[3] has been embedded.

For the classification and regression tasks, we follow the same setup as in the original RDF2vec paper [443]: For the classification tasks, four classifiers have been evaluated: *Naïve Bayes*, *C4.5* (decision tree algorithm), *KNN* with $k = 3$, and support vector machine *(SVM)* with $C \in \{10^{-3}, 10^{-2}, 0.1, 1, 10, 10^2, 10^3\}$ where the best C is chosen. A 10-fold cross validation has been used to calculate the performance statistics. For the regression tasks, three approaches have been evaluated: *linear regression*, *KNN*, and *M5rules*.

The datasets have been evaluated using the *Evaluation Framework for Node Embedding Techniques* [389], a publicly available[4] Python framework implemented to evaluate embeddings on the datasets described above. Table 9.1 depicts some characteristics of those datasets.

[3] `https://wiki.dbpedia.org/downloads-2016-10`

[4] `https://github.com/mariaangelapellegrino/Evaluation-Framework`

Table 9.1: Characteristics of the Test Datasets

Dataset	# Entities	Avg. Degree
AAUP	960	83.3775
Cities	212	1264.7122
Forbes	1585	36.1759
Metacritic Albums	1600	19.0762
Metacritic Movies	2000	17.5003
KORE	414	474.5984
LP50	407	2087.5274

In the results tables (9.2, 9.3, 9.4, 9.5), the *strategy* refers to the configuration with which the embeddings have been obtained. The structure can be read as follows:
<mode>_<number_of_walks_per_entity>_<walk_depth>_<training_mode>_<dimension> where *mode* is either *Light* or *Classic*. For example, *Light_500_4_CBOW_100* refers to RDF2vec Light embeddings with 500 walks per entity, a walk depth of 4, CBOW configuration, and an embedding space dimensionality of 100.

9.4 Results

The classification results are depicted in Table 9.2. Here, the skip-gram configurations outperform the CBOW ones. In three out of five cases, the classic embeddings obtain the overall best result – however, the corresponding light configurations achieve similarly high results. On the *Metacritic Movies* and *Metacritic Albums* datasets, the light configuration consistently outperforms the classic embeddings. On all datasets, SVMs are the best classifier. When using CBOW configurations, the light approach consistently outperforms the classic one. It is noteworthy that higher dimensionalities do not always perform better. The configurations with a dimensionality of 50 achieve the best results on two datasets. The classic embeddings have, to our knowledge, not yet been evaluated with a dimensionality of 50 before and yield relatively good performance.

The regression results are depicted in Table 9.3. They are similar to the classification ones: Again, the best configuration is SG. The classic approach scores best on three datasets compared to the light approach, which achieves the overall best scores on two datasets. The results obtained from a dimensionality of 50 are similar to the vectors with 100 elements – on Forbes, the overall best score is even achieved with a vector of size 50, albeit the scores are close.

For both classification and regression, we can observe that the difference between the best classic and the best light model is small, with the *Cities* dataset being an exception. Looking at the characteristics of the datasets depicted in Table 9.1, we can observe that the Cities dataset also has a much higher average degree of the entities at hand than the other classification and regression datasets. In particular, with a degree above 1,000, it is impossible that every statement about the entities is actually used for computing the embeddings if only 500 walks are created per entity. On the other hand, since in the classic approach, the 500 random walks are not only started from those 212 entities, but also from all its neighboring entities, a much larger fraction – if not all of those statements – are used for computing the embeddings. This results in more information about the entities at hand being captured in the walks created by the classic approach than the light approach.

In order to obtain a better understanding of the situations in which the light approach yields good results and those in which the classic approach is drastically better, we analyzed the graphs spanned by the walks that *RDF2vec Light* creates, i.e., the graphs which correspond to the union of all triples contained in a walk. Those graphs are depicted in Figure 9.2. It can be observed that very dense graphs correspond to setups where the light variant of RDF2vec yields the best results. Table 9.6 provides some key figures about those graphs. It can be seen in Table 9.6 that a high ratio between unique nodes and nodes of interest is a good indicator of the performance of the embeddings, i.e., cases in which the resulting spanned graph contains many more nodes than those of interest.

The document similarity results are depicted in Table 9.4. Here, the overall best harmonic mean can be obtained with the classic configuration. The gap between classic and light is larger for the skip-gram configurations, whereas for CBOW the light configurations again outperform the classic ones. The lower dimensionality of 50 does not sacrifice significant performance for the light embeddings. The high average degree of the *LP50* dataset indicates (see Table 9.1), that some relevant information is likely lost when using *RDF2vec Light*.

The entity relatedness results are shown in Table 9.5. As before, the skip-gram configurations perform better than the CBOW ones. Here, the distance from light to classic is negligible for CBOW but significant for SG. The light embeddings struggle when it comes to the concept *Chuck Norris* – an indicator that more background knowledge is required for this particular entity. Generally, it can be concluded for this task that more context is required to capture the latent features. This could be achieved, for example, by generating walks with a higher depth than the one used here.

In general, for the document and entity relatedness tasks, the result quality of *RDF2vec Light* is lower for skip-gram models than for the classic RDF2vec.

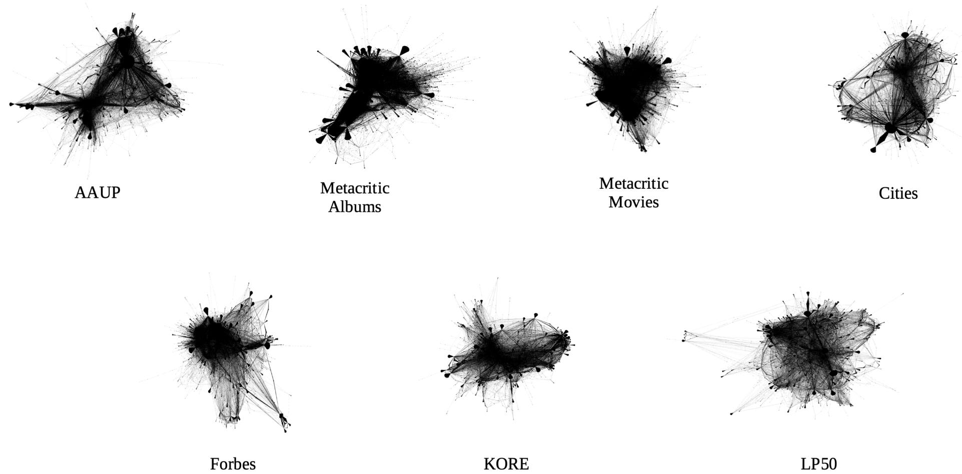


Figure 9.2: Depiction of the graphs which were assembled using the generated walks. The graphs are rendered using a force layout in Gephi.

One reason, looking at the dataset characteristics in Table 9.1, is again the high average degree of the entities in the dataset, especially in the LP50 dataset.

The experiments indicate that *RDF2vec Light* can handle a reduction of dimensionality well when comparing the 50-dimensional vector results with the 100-dimensional ones. This is intuitive since there are likely much fewer latent features available in the training data compared to the classic approach. The aforementioned ability reinforces the *light* aspects of the approach, making it even more suitable for environments where computing power and memory are limited. It could furthermore be seen that the average degree of the dataset is an indicator of the performance of the light embeddings.

All fully trained DBpedia embedding models created in the scope of this chapter are publicly available through KGvec2go [404].[5,6] It is important to note that the numbers are not fully comparable to the performance numbers quoted in [442] because in the latter publication, the DBpedia version of 2015 is used and all URIs are found while in this evaluation setting, not all URIs could be found due to changes in the underlying knowledge graph.

[5] `http://kgvec2go.org/download`

[6] Please note that we blackened all references to URLs, GitHub repositories, and libraries for the double-blind submission.

	Cities				Metacritic Movies				Metacritic Albums				AAUP				Forbes			
Strategy	**NB**	**KNN**	**SVM**	**C4.5**	**NB**	**KNN**	**SVM**	**C4.5**	**NB**	**KNN**	**SVM**	**C4.5**	**NB**	**KNN**	**SVM**	**C4.5**	**NB**	**KNN**	**SVM**	**C4.5**
Light_500_4_CBOW_50	64.91	67.73	**52.56**	57.25	67.28	65.73	**73.31**	60.77	68.34	67.24	**72.44**	59.74	58.0	52.66	**61.93**	50.08	52.72	54.96	**60.79**	50.41
Classic_500_4_CBOW_50	53.56	59.45	49.36	53.30	53.74	52.68	55.25	51.71	52.78	49.57	51.7	50.42	49.69	52.77	55.33	51.03	47.88	52.59	57.28	49.07
Light_500_4_CBOW_100	66.4	68.26	**71.78**	56.79	66.72	64.09	**73.5**	60.92	68.11	66.22	**72.43**	59.02	56.52	52.44	**63.21**	51.84	51.1	55.03	**60.81**	49.97
Classic_500_4_CBOW_100	54.22	57.70	49.36	48.32	53.83	52.48	58.21	51.79	55.48	51.24	57.44	51.25	51.26	54.78	55.00	51.48	48.84	50.81	57.36	49.99
Light_500_4_CBOW_200	65.01	68.15	**71.61**	60.5	67.33	63.93	**73.93**	61.33	68.03	66.04	**73.34**	59.71	55.96	51.86	**61.74**	50.27	50.25	55.11	**59.11**	50.26
Classic_500_4_CBOW_200	55.34	48.91	49.36	50.56	54.75	52.79	58.79	52.12	56.74	53.35	59.18	52.44	52.57	56.08	56.83	53.44	48.67	48.6	57.57	48.48
Light_500_4_SG_50	70.49	70.92	75.9	55.28	72.33	66.75	**74.15**	61.08	73.97	72.17	**76.49**	65.04	62.73	51.39	65.46	50.21	56.67	56.51	**61.56**	51.45
Classic_500_4_SG_50	78.13	74.37	<u>**80.57**</u>	57.49	69.21	65.28	72.81	59.90	70.51	73.10	76.42	62.65	61.64	59.64	<u>**68.04**</u>	54.04	55.76	54.62	61.08	51.65
Light_500_4_SG_100	70.68	67.62	73.99	54.79	72.87	67.29	<u>**74.89**</u>	60.88	74.27	72.52	<u>**76.98**</u>	60.83	63.52	54.24	64.54	52.57	56.71	56.29	**61.38**	50.31
Classic_500_4_SG_100	76.89	75.90	**79.01**	53.45	69.78	67.69	72.72	58.55	72.50	74.08	76.51	62.04	60.07	59.26	**64.72**	52.24	54.03	55.02	60.42	50.82
Light_500_4_SG_200	73.23	67.04	73.81	59.60	72.80	66.70	**74.58**	61.45	73.74	70.76	**76.35**	61.93	60.84	55.65	62.83	52.22	52.96	56.96	60.26	50.80
Classic_500_4_SG_200	75.01	75.38	**77.06**	49.88	71.19	58.86	73.85	59.04	73.07	75.4	75.66	63.98	57.29	57.29	**66.74**	51.04	53.23	54.47	<u>**61.82**</u>	50.21

Table 9.2: Classification Results: The quoted numbers refer to accuracy scores. The best score of each comparison group is printed in bold. The overall best score for a test set is additionally underlined.

	Cities			Metacritic Movies			Metacritic Albums			AAUP			Forbes		
Strategy	**LR**	**KNN**	**M5**	**LR**	**KNN**	**M5**	**LR**	**KNN**	**M5**	**LR**	**KNN**	**M5**	**LR**	**KNN**	**M5**
Light_500_4_CBOW_50	19.23	**17.09**	22.41	**19.75**	22.37	28.27	**12.52**	13.37	18.01	**68.35**	79.32	103.70	**34.62**	36.24	49.97
Classic_500_4_CBOW_50	16.95	18.34	23.74	22.77	25.83	32.0	14.06	16.2	19.82	70.59	77.43	94.42	36.64	36.78	53.51
Light_500_4_CBOW_100	21.16	**17.06**	23.96	**19.9**	22.72	28.58	**12.35**	13.55	17.96	**65.85**	79.44	98.78	**34.96**	36.04	49.03
Classic_500_4_CBOW_100	22.15	20.66	25.18	22.94	25.5	31.78	14.17	15.96	19.59	73.33	79.02	95.22	42.32	37.88	50.66
Light_500_4_CBOW_200	54.86	**17.25**	22.88	**19.60**	22.83	28.00	**12.50**	13.47	17.95	**67.65**	79.02	94.76	**35.97**	37.13	49.38
Classic_500_4_CBOW_200	99.73	21.35	25.9	23.54	25.83	31.7	14.24	15.47	19.45	80.29	80.28	94.4	45.76	38.19	53.58
Light_500_4_SG_50	19.39	16.07	26.69	**19.34**	21.71	28.68	12.0	12.8	16.85	67.66	79.91	104.59	**34.58**	36.27	49.83
Classic_500_4_SG_50	12.95	<u>**12.29**</u>	20.85	19.89	22.77	28.52	**11.80**	12.31	16.91	<u>**64.85**</u>	72.06	94.68	34.89	37.13	50.46
Light_500_4_SG_100	20.89	17.09	23.74	<u>**19.21**</u>	21.57	28.52	11.89	12.58	17.36	66.59	77.66	99.31	<u>**34.48**</u>	36.97	49.32
Classic_500_4_SG_100	15.26	**13.15**	24.05	19.61	22.02	29.23	<u>**11.57**</u>	12.12	17.21	**65.50**	77.14	98.46	35.26	36.5	50.04
Light_500_4_SG_200	44.38	16.57	23.71	**19.45**	21.82	28.1	12.16	12.79	17.22	70.13	78.39	98.50	36.73	**35.89**	47.88
Classic_500_4_SG_200	28.34	**13.49**	25.05	19.71	23.71	29.15	11.92	**11.82**	17.1	**67.96**	84.9	101.43	36.93	37.34	50.02

Table 9.3: Regression results: The quoted numbers refer to the RMSE. The best score of each comparison group is printed in bold. The overall best score for a test set is additionally underlined.

Strategy / Score	**Pearson**	**Spearman**	**Harmonic Mean**
Light_500_4_CBOW_50	0.4171	0.3612	**0.3871**
Classic_500_4_CBOW_50	0.2908	0.1815	0.2235
Light_500_4_CBOW_100	0.4059	0.3469	**0.3741**
Classic_500_4_CBOW_100	0.3233	0.1774	0.2291
Light_500_4_CBOW_200	0.4114	0.3546	**0.3809**
Classic_500_4_CBOW_200	0.3433	0.1814	0.2374
Light_500_4_SG_50	0.4561	0.2738	0.3421
Classic_500_4_SG_50	0.5238	0.3793	**0.440**
Light_500_4_SG_100	0.4449	0.2635	0.3310
Classic_500_4_SG_100	0.5679	0.3737	<u>**0.4507**</u>
Light_500_4_SG_200	0.4464	0.2589	0.3278
Classic_500_4_SG_200	0.5613	0.3580	**0.4371**

Table 9.4: Document Similarity Scores on LP50. The best scores of each group are printed in bold. The overall best score is additionally underlined.

Strategy	IT Companies	Hollywood Celebrities	Television Series	Video Games	Chuck Norris	All
Light_500_4_CBOW_50	0.2463	0.3621	**0.3537**	**0.4168**	0.1263	**0.3343**
Classic_500_4_CBOW_50	**0.2989**	**0.3832**	0.2105	0.2842	**0.3789**	0.2982
Light_500_4_CBOW_100	0.2211	**0.3558**	**0.3221**	**0.4126**	0.1053	**0.3179**
Classic_500_4_CBOW_100	**0.3137**	0.3052	0.2232	0.3789	**0.2526**	0.3028
Light_500_4_CBOW_200	0.2147	**0.3389**	**0.3958**	**0.4800**	0.1474	**0.3474**
Classic_500_4_CBOW_200	**0.3579**	0.2842	0.2947	0.2968	<u>**0.5053**</u>	0.3178
Light_500_4_SG_50	**0.4947**	**0.4926**	0.4274	0.5158	0.3579	0.4767
Classic_500_4_SG_50	0.4926	0.4568	<u>**0.5179**</u>	<u>**0.5873**</u>	**0.3684**	**0.5068**
Light_500_4_SG_100	0.4421	0.4505	0.3663	0.4695	0.3474	0.4281
Classic_500_4_SG_100	<u>**0.5621**</u>	**0.4905**	**0.5032**	**0.5032**	**0.4737**	**0.5288**
Light_500_4_SG_200	0.4632	0.3895	0.3495	0.4484	0.2421	0.4045
Classic_500_4_SG_200	**0.5579**	<u>**0.5305**</u>	**0.4779**	**0.5789**	<u>**0.5053**</u>	<u>**0.5348**</u>

Table 9.5: Results on the Entity Relatedness Task Using Cosine Similarity. The best value of each comparison group is highlighted in bold. The overall best value is additionally underlined. The values represent Kendall's Tau.

	Nodes	Avg. Degree	Modularity	Nodes / Nodes of Interest
AAUP	76,665	1.895	0.636	79.86
Cities	100,942	3.085	0.722	476.14
Forbes	58,431	3.997	0.609	36.86
Metacritic Albums	34,086	6.163	0.423	21.30
Metacritic Movies	44,840	2.889	0.453	22.42
KORE	56,227	3.618	0.67	135.81
LP50	70,738	1,565	0.731	176.26

Table 9.6: The generated walks can be re-assembled to one graph per dataset. Those graphs are statistically described here.

9.5 Implementation

9.5.1 jRDF2vec Implementation and Public API

The *RDF2vec* method, as well as the *RDF2vec Light* extension, have been implemented in Java and Python. The code is publicly available[7] on GitHub under the name *jRDF2vec* using an MIT-license[8]. The implementation relies on the *gensim* framework [431] for training and *flask* for inter-process communication management. *jRDF2vec* allows training RDF2vec embeddings in classic and light mode with a focus on scalability for large knowledge graphs. Multiple walk generation modes are available (e.g., the classic walk generation algorithm or the algorithm presented in this chapter), and the walk generation can be distributed to multiple threads through a thread pool. *jRDF2vec* can handle various RDF formats such as n-triples, RDF/XML, Turtle, or *Header, Dictionary, Triples* (HDT) [150].
In addition, an easy-to-use REST API has been implemented and is provided on `http://www.kgvec2go.org`.[9] An example of how to call the API is also provided.[10]

The code to re-assemble graphs from walks is implemented in Java. The program reads generated walks, builds the graph, and outputs it in the *Pajek Net* graph format that can be read by graph analysis desktop programs such as *Pajek* or *Gephi*. The code is publicly available on GitHub as well.[11]

---

[7] `https://github.com/dwslab/jRDF2Vec`

[8] The *MIT License* is a permissive license which originated from the *Massachusetts Institute of Technology* (MIT). For more information see: `https://choosealicense.com/licenses/mit/`

[9] The code for the server is also publicly available and can also be run locally: `https://github.com/janothan/kgvec2go-server/`

[10] `https://github.com/janothan/kgvec2go-server/blob/master/examples/KGvec2go_rdf2vec_light.ipynb`

[11] `https://github.com/janothan/WalksToPajekNetFile`

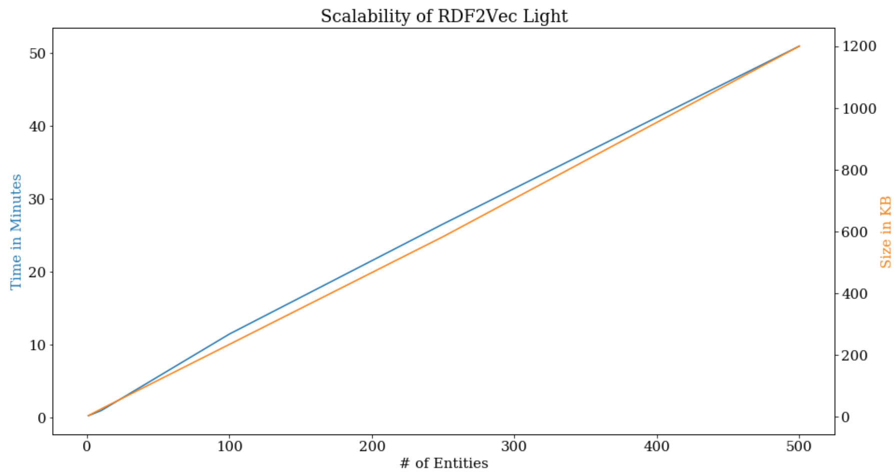


Figure 9.3: *RDF2vec Light* scalability: Depicted is the generation and training time in minutes for an increasing number of entities together with the storage requirements for the resulting vectors. In the given setting, the classic approach does not finish within five days.

9.5.2 Performance

The main advantage of *RDF2vec Light* is its increased performance compared to the traditional approach. It can be run on very small machines such as consumer laptops (and even mobile devices) while achieving similar performance. Figure 9.3 shows the training time and storage size required for an increasing number of entities $|e| \in \{1, 10, 250, 500\}$ in a low-performance setting (4 threads) using HDT disk access for the walk generation.[12] The URIs for the entities of interest have been randomly drawn in an additive manner so that the performance calculation for 500 entities contains all the URIs that were used before for smaller configurations. The experiment has been performed on the DBpedia knowledge graph with $walks = 200$, $depth = 4$, $mode = sg$, and $dimension = 100$. A full embedding run on the whole knowledge graph with the setting outlined above takes more than five days. As visible from the figure, the walk generation and training time scale between 0 and 500 entities of interest almost linearly. The resulting vector file size scales exactly linearly.[13]

[12] Note that faster run times can be achieved with the implementation by switching from HDT to the memory-based implementation (higher RAM requirements) or by increasing the number of threads (higher CPU requirements).

[13] Note that the implementation includes only the vectors of interest in the resulting vector file.

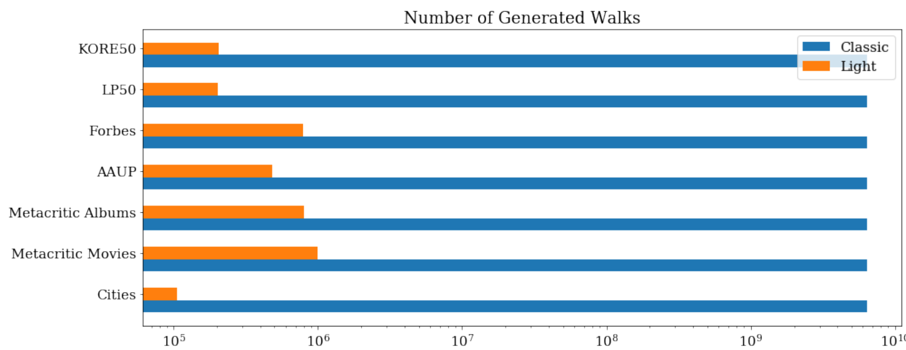


Figure 9.4: Comparison of the number of generated walks of both approaches on a logarithmic scale.

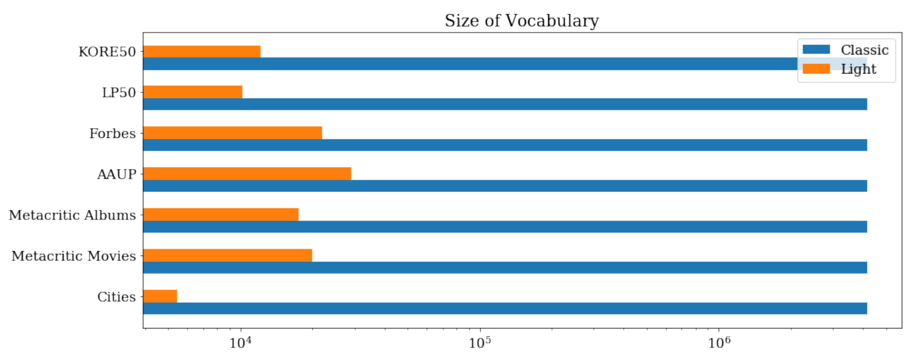


Figure 9.5: Comparison of the vocabulary size of both approaches on a logarithmic scale.

The walk generation scales with $O(|e|)$ and can be distributed to multiple subprocesses. Thereby, the implementation can also be run on larger computation servers efficiently. In terms of the number of walks generated, the lightweight approach generates roughly four orders of magnitude fewer walks compared to the classic approach, as visible in Figure 9.4. In terms of the vocabulary size, the classic model works with two orders of magnitudes more concepts in the training step, which is depicted in Figure 9.5.[14] Consequently, the memory requirements of the light approach are significantly lower in the training phase.

[14] Note that the *jRDF2vec* implementation persists only the vectors of interest so that the storage requirements for the light approach are significantly lower than the vocabulary size in Figure 9.5 implies.

9.6 Conclusion

In this chapter, we presented *RDF2vec Light*, an approach for learning latent representations of knowledge graph entities that requires only a fraction of the computing power compared to other embedding approaches. Rather than embedding the whole knowledge graph, *RDF2vec Light* trains vectors for only few entities of interest and their context. For this approach, the walk generation algorithm has been adapted to better represent the context of the entities.

Multiple experiments showed similar performance results compared to the classic RDF2vec method. While our approach outperforms the classic model when using CBOW, the skip-gram performance is more balanced. The experiments indicate that *RDF2vec Light* can handle a reduction of dimensionality well. It could further be shown that a low average degree of the entities of interest indicates relatively good performance for *RDF2vec Light*. The full implementation is available online.

On the relatedness and similarity tasks, improvements could be achieved by adding more context to the entities of interest, for example, by increasing the depth, the number of walks, or by extending the model to also generate walks for neighbors. Given that the average degree of the entities of interest is known, walk generation parameters could also be set dynamically. The approach presented here reduces the training data for the neural model. Possible extensions to the model could include also changing the actual training algorithm: Currently, in the training phase, embeddings are trained for all components of the sentences rather than just on the entities of interest. A tweak in the learning approach could increase the runtime performance further.

In subsequent chapters, we extend the downstream application scenarios – in particular, we exploit the semantic features for ontology matching. A concrete application of RDF2vec Light can be found in Chapter 18.

Chapter 10

Order-Aware RDF2vec

In this chapter, a small but very effective adaption of the classic RDF2vec algorithm is proposed and evaluated: While the classic RDF2vec algorithm ignores the order of concepts in the context window, the variation presented in this chapter – named $RDF2vec_{oa}$ – is *order-aware* (OA), i.e., considers the positions within the $word2vec$ context window.

The work presented in this chapter has been published before as: Portisch, Jan; Paulheim, Heiko. Putting RDF2vec in Order. In: Proceedings of the International Semantic Web Conference - Posters and Demos, ISWC 2021. 2021. [412]

10.1 Introduction

RDF2vec [442] is a representation learning approach for entities in a knowledge graph. The basic idea is to first create *sequences* from a knowledge graph by starting random walks from each node. These sequences are then fed into the *word2vec* algorithm [345, 344] for creating word embeddings, with each entity or property in the graph being treated as a "word". As a result, a fixed-size feature vector is obtained for each entity.

Word2vec is a well-known neural language model to train latent representations (i.e., fixed-size vectors) of words based on a text corpus. Its objective is either to predict a word w given its context words (known as continuous bag-of-words or CBOW), or vice versa (known as skip-gram or SG).

Given the context *k* of a word w, where *k* is a set of preceding and succeeding words of w, the learning objective of word2vec is to predict w. This is known as the *continuous bag of words* model. The *skip-gram* model is trained the other

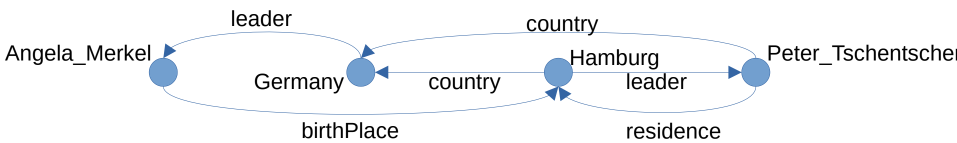


Figure 10.1: Example Knowledge Graph

way around: Given w, k has to be predicted. Within this training process, the size of k is also known as *window* or *window size*.

One shortfall of the original word2vec approach is its insensitivity to the relative positions of words. It is, for instance, irrelevant whether a word is preceding or succeeding w, and the actual distance to w is not considered. This property of word2vec is ideal to cope with the fact that in many languages, the same sentence can be expressed with different word orderings (cf. *Yesterday morning, Tom ate bread* vs. *Tom ate bread yesterday morning*). In contrast, walks extracted from knowledge graphs, the semantics of the underlying nodes differ depending on the position of an entity in the walk, as the following examples illustrate.

Fig. 10.1 depicts a small excerpt of a knowledge graph. Among others, the following walks could be extracted from the graph:

```
Hamburg -> country -> Germany            -> leader     -> Angela_Merkel
Germany -> leader  -> Angela_Merkel      -> birthPlace -> Hamburg
Hamburg -> leader  -> Peter_Tschentscher -> residence  -> Hamburg
```

If an RDF2vec model is trained for the entities in the center (i.e., `Germany`, `Angela_Merkel`, and `Peter_Tschentscher`), all of the sequences share exactly two entities in their context (`Hamburg` and `leader`), i.e., they will be projected equally close in the vector space. However, a model respecting positions would particularly differentiate the different meanings of `leader` (i.e., whether someone/thing *has* or *is* a leader), and the different *roles* of involved entities (i.e., `Hamburg` as a place of birth or a residence of a person, or being located in a country). Therefore, it would map the two politicians closer to each other than to `Germany`.

Ling et al. [319] present an extension to the word2vec algorithm, known as *structured word2vec*, which incorporates the positional information of words. This is achieved by using multiple encoders (CBOW) respectively decoders (SG), depending on the position of the context words. An illustration for SG can be found in Figure 10.2, where it is visible that the classic component uses only one output matrix O, which maps the embeddings to the output, while the structured approach uses one output matrix per position in the window (e.g., O_{+1} for the subsequent word to w_0).

In this chapter, we present $RDF2vec_{oa}$, an *order-aware* variant of RDF2vec obtained by changing the training component from word2vec to structured word-2vec, and show promising preliminary results.

10.2 Related Work

RDF2vec was one of the first approaches to adopt statistical language modeling techniques to knowledge graphs. Similar approaches, such as *node2vec* [173] and *DeepWalk* [391], were proposed for unlabeled graphs, while knowledge graphs are labeled by nature, i.e., they contain different types of edges.

Other language modeling techniques that have been adapted for knowledge graphs include GloVe [390], which yielded *KGlove* [85], and BERT [102], which yielded *KG-BERT* [603].

Variants of RDF2vec include the use of different heuristics for biasing the walks [84]; [542] evaluate multiple heuristics for biasing the walks, or alternative walk strategies. Very few authors tried to change the training objective of RDF2vec. Besides word2vec, the GloVe [390] algorithm has also been used [85].

10.3 Experiments and Preliminary Results

We use jRDF2vec[1] [405] to generate random walks and Ling et al.'s structured word2vec implementation[2] to train an embedding based on the walks.

For the embeddings, we use the DBpedia 2016-04 dataset. We generated 500 random walks for each node in the graph with a depth of 4 (node hops).

[1] `https://github.com/dwslab/jRDF2Vec`

[2] `https://github.com/wlin12/wang2vec`

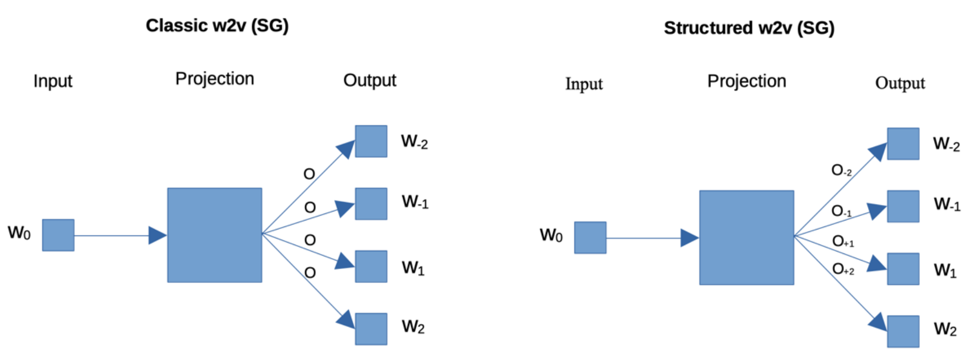


Figure 10.2: Classic word2vec vs. Structured word2vec

word2vec and structured word2vec were trained using the same set of walks and the same training parameters: *SG*, $window = 5$, and $size \in \{100, 200\}$.

We evaluate both the classic and the position-aware RDF2vec approach on a variety of different tasks and datasets. For our evaluation, we use the *GEval* framework [388]. We follow the setup proposed in [443] and [388]. Those works use data mining tasks with an external ground truth. Different feature extraction methods – which include the generation of embedding vectors – can then be compared using a fixed set of learning methods. Overall, we evaluate our new embedding approach on six tasks using 20 datasets altogether. The evaluation is conducted on six different downstream tasks – classification and regression, clustering, determining semantic analogies, and computing entity relatedness and document similarity, the latter based on entities mentioned in the documents.

The results are presented in Table 10.1. When comparing the classic to the order-aware embeddings, it is visible that the performances are very similar on most tasks such as classification. A first observation is that we cannot observe significant performance drops on any of the tasks when switching from classic to order-aware RDF2vec embeddings. However, significant performance increases can be observed on clustering tasks and on semantic analogy tasks, which are the tasks where entities of different classes are involved (whereas the classification and regression tasks deal with entities of the same class, e.g., cities or countries). The order-aware RDF2vec configuration with 100 dimensions achieved on seven datasets the overall best results and outperforms its classic configuration with the same dimension on ten datasets, partly with significantly better outcomes. On the other hand, in most cases where the classic variant performs better, it does so by a smaller margin. Thus, in general, the order-aware variant can be used safely without performance drops, and in some cases, with significant performance gains.

10.4 Conclusion

In this chapter, we presented a position-aware variant of RDF2vec together with first, very promising evaluation results.

The approach is evaluated in more depth and combined with other RDF2vec configurations and flavors in Chapters 7, 11, 12, and 13.

Table 10.1: Results of RDF2vec$_{classic}$ (c-100, c-200) and RDF2vec$_{oa}$ (oa-100, oa-200) trained with 100 and 200 dimensions respectively. The best value in each dimension group is printed in bold, the overall best value is additionally underlined.

Task	Metric	Dataset	c-100	oa-100	c-200	oa-200
Classification	ACC	AAUP	<u>**0.693**</u>	0.679	**0.692**	0.683
	ACC	Cities	**0.793**	**0.793**	0.798	<u>**0.807**</u>
	ACC	Forbes	**0.629**	0.607	<u>**0.635**</u>	0.630
	ACC	Metacritic Albums	0.783	<u>**0.799**</u>	0.788	**0.792**
	ACC	Metacritic Movies	**0.757**	0.736	<u>**0.763**</u>	0.748
Clustering	ACC	Cities/Countries (2k)	0.755	**0.939**	0.758	<u>**0.946**</u>
	ACC	Cities/Countries	<u>**0.786**</u>	0.785	0.7624	**0.766**
	ACC	Cities/Albums/Movies /AAUP/Forbes	<u>**0.932**</u>	0.931	0.861	**0.929**
	ACC	Teams	0.969	<u>**0.971**</u>	0.892	**0.945**
Regression	RMSE	AAUP	65.151	<u>**62.624**</u>	66.301	**65.077**
	RMSE	Cities	12.726	<u>**11.220**</u>	14.855	**13.484**
	RMSE	Forbes	<u>**34.290**</u>	34.340	36.460	**35.967**
	RMSE	Metacritic Albums	11.366	<u>**11.215**</u>	**11.528**	11.651
	RMSE	Metacritic Movies	**19.091**	19.530	<u>**19.078**</u>	19.432
Semantic Analogies	ACC	Capital-Countries	0.852	<u>**0.990**</u>	0.872	**0.949**
	ACC	Capital-Countries (all)	0.832	<u>**0.933**</u>	**0.901**	0.896
	ACC	Currency-Country	0.417	**0.520**	<u>**0.537**</u>	0.441
	ACC	City-State	0.5577	**0.607**	0.555	<u>**0.627**</u>
Entity Relatedness	Harmonic Mean	-	**0.726**	0.716	**0.747**	<u>**0.747**</u>
Document Similarity	Kendall Tau	-	<u>**0.405**</u>	0.373	**0.350**	0.325

Chapter 11

RDF2vec Walk Strategies

In previous chapters, RDF2vec has already been introduced: It is a knowledge graph embedding mechanism which first extracts sequences from knowledge graphs by performing random walks, then feeds those into the word embedding algorithm word2vec for computing vector representations for entities. In this chapter, we introduce two new flavors of walk extraction coined *e-walks* and *p-walks*, which put an emphasis on the structure or the neighborhood of an entity respectively and thereby allow for creating embeddings which focus on similarity or relatedness. By combining the walk strategies with order-aware and classic RDF2vec, as well as CBOW and skip-gram word2vec embeddings, we conduct a preliminary evaluation with a total of 12 RDF2vec variants.

The work presented in this chapter has been published before as: Portisch, Jan; Paulheim, Heiko. Walk this Way! Entity Walks and Property Walks for RDF2vec. In: The Semantic Web: ESWC 2022 Satellite Events. 2022. [to appear] [416]

11.1 Introduction

RDF2vec [440] is an approach for embedding entities of a knowledge graph in a continuous vector space. It extracts sequences of entities from knowledge graphs, which are then fed into a word2vec encoder [345]. Such embeddings have been shown to be useful in downstream tasks which require numeric representations of entities and rely on a distance metric between entities that captures entity similarity and/or relatedness [399].

Different variants for walk extraction in RDF2vec have been proposed in the past, including the inclusion of weights in the random component [84] and the

#	RDF2vec	p-RDF2vec	e-RDF2vec
1	Ludwigshafen	Arnsberg	Ludwigshafen
2	Peter Kurz	Frankfurt	Timeline of Mannheim
3	Timeline of Mannheim	Tehran	Peter Kurz
4	Karlsruhe	Bochum	Adler Mannheim
5	Adler Mannheim	Bremen	Peter Kurze

Table 11.1: Five nearest neighbors to *Mannheim* in RDF2vec (classic), p-RDF2vec, and e-RDF2vec trained on DBpedia (SG)

use of other walk strategies such as community hops and walklets [509]. Moreover, it has been shown recently that using an order-aware variant instead of classic word2vec improves the resulting embeddings [412].

RDF2vec mixes the notion of similarity and relatedness. This can be seen, for example, in Table 11.1: The closest concepts in the vector space for *Mannheim* are comprised of the city timeline, a person, the local ice hockey team, and close cities. All of these are *related* to the city in a sense that they have a semantic relation to Mannheim (Peter Kurz, for instance, is Lord mayor of Mannheim). However, these concepts are not *similar* to the city since a person and a city do not have much in common.

In this chapter, we present two new variants of RDF2vec: *p-RDF2vec* emphasizes structural properties of entities, i.e., their attributes, and consequently has a higher exposure towards similarity. *e-RDF2vec* emphasizes the neighboring entities, i.e., the context of entities, and consequently has a higher exposure towards relatedness.

11.2 New Walk Flavors

In the following, we define a knowledge graph $\mathcal{G}$ as a labeled directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{R} \times \mathcal{V}$ for a set of relations $\mathcal{R}$. Vertices are subsequently also referred to as *entities* and edges as *predicates*.

Classic RDF2vec creates sequences of random walks. A random walk of length n (for an even number n) for w_0 has the form

$$w = (w_{-\frac{n}{2}}, w_{-\frac{n}{2}+1}, ..., w_0, ..., w_{\frac{n}{2}-1}, w_{\frac{n}{2}}) \tag{11.1}$$

where $w_i \in \mathcal{V}$ if i is even, and $w_i \in \mathcal{R}$ if i is odd. For better readability, we stylize $w_i \in \mathcal{V}$ as e_i and $w_i \in \mathcal{R}$ as p_i:

$$w = (e_{-\frac{n}{2}}, p_{-\frac{n}{2}+1}, ..., e_0, ..., p_{\frac{n}{2}-1}, e_{\frac{n}{2}}) \tag{11.2}$$

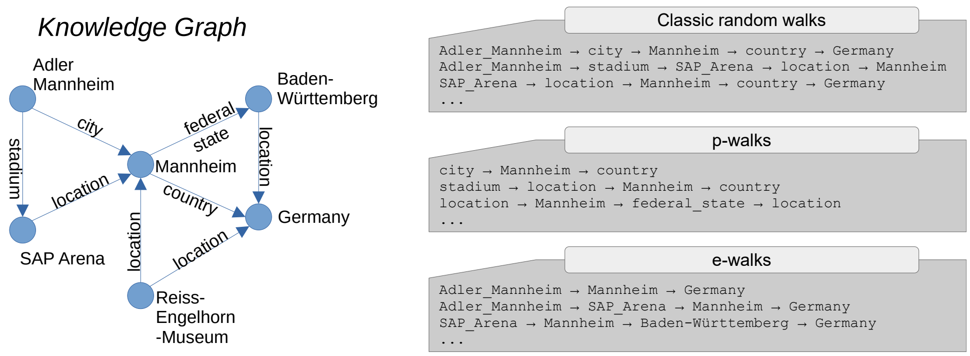


Figure 11.1: Illustration of Different Walk Types

In the case of loops, it is possible that a walk contains an entity or edge more than once.

From the definition of random walks, we derive two other types of random walks (see Fig. 11.1): A *p-walk* w_p is a subsequence of a walk w which consists of only the focus entity e_0 and the predicates in the walk, i.e.,

$$w_p = (p_{-\frac{n}{2}+1}, p_{-\frac{n}{2}+3}, ..., e_0, ..., p_{\frac{n}{2}-3}, p_{\frac{n}{2}-1}) \tag{11.3}$$

In contrast, an *e-walk* consists only of the entities in the walk, i.e.,

$$w_e = (e_{-\frac{n}{2}}, e_{-\frac{n}{2}+2}, ..., e_0, ..., e_{\frac{n}{2}-2}, e_{\frac{n}{2}}) \tag{11.4}$$

In other words: p-walks capture the *structure* around an entity, while e-walks capture the *context*. Thus, we hypothesize that embeddings computed from p-walks capture (structural) *similarity*, while those computed from e-walks capture contextual similarity, which can also be understood as *relatedness*.

11.3 Evaluation

We evaluate embeddings obtained using three different walk extraction strategies, i.e., classic walks, p-walks, and e-walks, and training with classic word2vec as well as order-aware word2vec, using both the CBOW and skip-gram variants.

This, in total, yields 12 different configurations for RDF2vec.[1] All embedding models are publicly available to download via KGvec2go [404].[2]

For evaluation, we use the framework proposed in [388], which consists of different tasks (classification, regression, clustering, analogy reasoning, entity relatedness, document similarity). We use a recent DBpedia release[3]. The results are depicted in Table 11.2. We can make a few interesting observations:

1. In 12/20 cases, the best results are achieved with classic walks. p-walks yield the best results in 3/20 cases, e-walks do so in 5/20 cases.

2. For entity relatedness, e-walks yield the best results, showing that those walks actually capture relatedness best.

3. For document similarity, p-walks outperform the other approaches. One explanation could be that structural similarity of entities (e.g., politicians vs. athletes) is more important for that task.

4. Semantic analogies are known to require both, relatedness and similarity.[4] Therefore, one may expect both p-walks and e-walks to perform poorly, which is indeed verified by our experiments.

5. As observed in [412], the order-aware variants almost always outperform the non-order-aware ones, for all kinds of walks, except for the semantic analogy problems. This effect is even slightly stronger for p-walks and e-walks than for classic RDF2vec.

6. Generally, skip-gram (and its order-aware variant) are more likely to yield better results than CBOW.

Table 11.1 shows the five closest concepts for classic RDF2vec and the extensions presented in this chapter. It can be seen that classic and e-RDF2vec have exposure towards relatedness while p-RDF2vec results in similar entities (i.e., only cities) being retrieved.

[1] We generated 500 walks per node with a depth of 4, i.e., we perform 4 node hops. All embeddings are trained with a dimensionality of 200. The experiments were performed with jRDF2vec (`https://github.com/dwslab/jRDF2Vec`), which implements all the different variants used in this chapter.

[2] `http://kgvec2go.org/download.html`

[3] `https://www.dbpedia.org/blog/snapshot-2021-09-release/`

[4] For solving an analogy task like *Paris is to France like Berlin is to X*, *X* must be similar to *France*, as well as related to *Berlin*.

11.4 Conclusion

In this work, we have shown that p-walks and e-walks are interesting alternatives, which, in particular in combination with the order-aware variant of RDF2vec, can outperform classic RDF2vec embeddings. Moreover, we have seen that using p-walks and e-walks can help create embeddings whose distance function reflects similarity and relatedness respectively.

At the same time, the evaluation is still not very conclusive. Therefore, we compiled collections of real and synthetic test cases, which allow us to make clear statements about which techniques are promising for which kind of problem. The test case creation, together with an evaluation, is presented in the following chapter.

Table 11.2: Result of the 12 RDF2vec variants on 20 tasks. The best score for each task is printed in bold. The suffix $_{oa}$ marks the order-aware variant of RDF2vec.

			Classic RDF2vec				e-RDF2vec				p-RDF2vec			
Task	Metric	Dataset	sg	sg_{oa}	cbow	$cbow_{oa}$	sg	sg_{oa}	cbow	$cbow_{oa}$	sg	sg_{oa}	cbow	$cbow_{oa}$
Classification	ACC	AAUP	0.706	0.713	0.643	0.690	0.696	**0.717**	0.703	0.690	0.564	0.623	0.551	0.612
		Cities	**0.818**	0.803	0.725	0.723	0.770	0.743	0.750	0.702	0.606	0.677	0.501	0.707
		Forbes	**0.623**	0.605	0.575	0.600	0.608	0.605	0.612	0.600	0.581	0.610	0.560	0.578
		Metacritic Albums	0.586	0.585	0.536	0.532	0.596	0.583	0.564	0.584	0.634	0.632	0.569	**0.667**
		Metacritic Movies	0.726	0.716	0.549	0.626	0.724	**0.732**	0.686	0.676	0.610	0.660	0.535	0.663
Clustering	ACC	Cities and Countries (2k)	0.789	0.900	0.520	**0.917**	0.726	0.726	0.668	0.660	0.605	0.520	0.637	0.733
		Cities and Countries	0.587	0.760	0.783	0.720	0.749	0.766	**0.820**	0.745	0.687	0.782	0.787	0.728
		Cities, Albums Movies, AAUP, Forbes	0.829	**0.854**	0.547	0.652	0.759	0.828	0.557	0.719	0.598	0.798	0.663	0.748
		Teams	0.909	0.931	0.940	0.925	0.889	0.926	0.916	0.931	**0.941**	0.938	0.940	0.580
Regression	RMSE	AAUP	65.985	**63.814**	77.250	66.473	67.337	65.429	70.482	69.292	80.318	72.610	96.248	77.895
		Cities	15.375	**12.782**	18.963	19.287	17.017	16.913	17.290	20.798	20.322	17.214	24.743	20.334
		Forbes	36.545	**36.050**	39.204	37.067	38.589	38.558	39.867	36.313	37.146	36.374	37.947	38.952
		Metacritic Albums	15.288	15.903	15.812	15.705	15.573	15.785	15.574	**14.640**	15.178	14.869	15.000	16.679
		Metacritic Movies	**20.215**	20.420	24.238	23.362	20.436	20.258	23.348	22.518	23.235	22.402	23.979	22.071
Semantic Analogies	ACC	capital country entities	**0.957**	0.864	0.810	0.789	0.794	0.747	0.660	0.397	0.008	0.091	0.000	0.036
		all capital country entities	**0.905**	0.857	0.594	0.758	0.657	0.591	0.359	0.592	0.014	0.073	0.002	0.052
		currency entities	**0.574**	0.535	0.338	0.447	0.309	0.193	0.198	0.297	0.006	0.076	0.002	0.085
		city state entities	**0.609**	0.578	0.507	0.442	0.459	0.484	0.250	0.361	0.009	0.048	0.000	0.036
Entity Relatedness	Kendall Tau		0.747	0.716	0.611	0.547	**0.832**	0.800	0.726	0.779	0.432	0.768	0.568	0.737
Document Similarity	Harmonic Mean		0.237	0.230	0.283	0.209	0.275	0.250	0.170	0.111	0.193	**0.382**	0.296	0.256

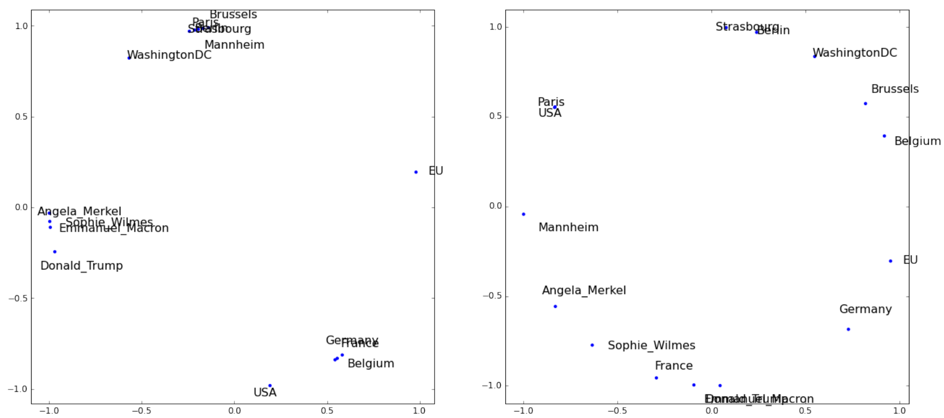
Chapter 12

A DL Benchmark for Knowledge Graph Embedding Evaluation

Knowledge graph embeddings have gained a lot of uptake during the time of this dissertation and have been heavily used in link prediction and other downstream prediction tasks. Most approaches are evaluated on a single task or a single group of tasks to determine their overall performance. The evaluation is then assessed in terms of how well the embedding approach performs on the task at hand, but it is hardly evaluated (and often not even deeply understood) what information the embedding approaches are *actually* learning to represent.

To fill this gap, we present the *Description Logic Class Constructors* (DLCC) benchmark, a resource to analyze embedding approaches in terms of which kinds of classes they can represent. Two gold standards are presented, one based on the real-world knowledge graph DBpedia and one synthetic gold standard. In addition, an evaluation framework is provided, which implements an experiment protocol so that researchers can directly use the gold standard. To demonstrate the use of DLCC, we compare multiple embedding approaches using the gold standards. We find that many *description logic* (DL) constructors on DBpedia are actually learned by recognizing different correlated patterns than those defined in the gold standard and that specific DL constructors, such as cardinality constraints, are particularly hard to be learned for most embedding approaches.

The work presented in this chapter has been submitted for publication as: Portisch, Jan; Paulheim, Heiko. The DLCC Node Classification Benchmark for Analyzing Knowledge Graph Embeddings. International Semantic Web Conference (ISWC 2022). 2022. [to appear] [414]



(a) Good class separation (b) Bad class separation

Figure 12.1: Two Example Embeddings. The left-hand side embedding shows a good class separation of persons, countries, and cities, whereas the right-hand side one does not.

12.1 Introduction

Knowledge graph embeddings are projections of entities and relations to continuous vector spaces. They have been proposed for various purposes and are typically evaluated on task-specific gold standards such as FB15k and WN18 [44] for link prediction, kgbench for node classification [40], or [443] for machine learning tasks such as classification, regression, or clustering. The benchmarks frequently come with their own evaluation protocol.

Independent of the original benchmark task, knowledge graph embeddings are generally versatile so that they can be used for multiple tasks [399]. While the performance of embeddings in downstream tasks is often superior to other entity representation techniques, most, if not all, embedding approaches have in common that it is not ultimately clear *what* is learned. For example, both for link prediction and for node classification, it is required that classes can be separated (e.g., persons, countries, and cities are clustered in the embedding space) [399], but so far, it has not been systematically evaluated which embedding methods can learn which kinds of class separations. Figure 12.1 shows an example of two embedding spaces with different qualities of class separation.

In this chapter, we present the DLCC (for *Description Logic Class Constructors*) dataset and an evaluation framework that help to better analyze and understand embedding approaches for specific DL constructors. There are four contributions of this chapter: (1) A framework for the DLCC gold standard creation is presented, (2) two concrete gold standards are provided – a real graph-based gold standard and one based on synthetic knowledge graphs, (3) an evaluation framework is provided to easily evaluate and compare the class separation capabilities of embeddings, and (4) a preliminary analysis for different state of the art embedding approaches is provided.

12.2 Related Work

In the area of link prediction (or knowledge base completion), the two well-known evaluation datasets, FB15k and WN18 [44], are both based on real datasets: FB15k is based on the Freebase dataset, and WN18 is based on WordNet [149]. They were presented in the context of link prediction: Given a triple in the form *(**h**ead, **r**elation, **t**ail)*, two prediction tasks *(head, relation, ?)* and *(?, relation, tail)* are created. The evaluation is performed by calculating the mean rank/HITS@10 for a list of proposals. Since it has been remarked that those datasets contain too many simple inferences due to inverse relations, the more challenging variants FB15k-237 [534] and WN18RR [100] have been proposed. More recently, evaluation sets based on larger knowledge graphs, such as YAGO3-10 [100] and DBpedia50k/DBpedia500k [477] have been introduced.

Bloem et al. [40] introduce *kgbench*, a node classification benchmark for knowledge graphs, which, like DLCC, comes with datasets in different sizes and predefined train/test splits. Unlike DLCC, kgbench is based on real-world datasets. Therefore, it is suitable to evaluate and compare the quality of different embedding approaches on real-world tasks but does not provide any insights into what these embedding approaches are capable of representing.

Alshagari et al. [17] present a framework for ontological concepts covering three aspects: (i) categorization, (ii) hierarchy, and (iii) logic validation. The framework can be used for language models and for knowledge graph embeddings. The work presented in this chapter differs in that it goes beyond explicit DBpedia types. The evaluation of this chapter is, therefore, of an analytical rather than a descriptive nature. Moreover, the task sets of DLCC are significantly larger and more comprehensive.

Ristoski et al. [443] provide a collection of benchmarking datasets for machine learning, including classification, clustering, and regression tasks. Later, the GEval framework [388, 389] was introduced to provide a standardized evaluation protocol for this dataset. The evaluation datasets are based on DBpedia. Internally, the embeddings are processed by different downstream classification, regression, or clustering algorithms. The evaluation framework presented in this chapter is similar to GEval in that it also evaluates multiple classifiers given a concept vector input.

Melo and Paulheim provide a method for synthesizing benchmark datasets for link and entity type prediction, which are used in conjunction with a fixed ontology. [339] Their goal is to mimic the characteristic of existing knowledge graphs in terms of distributions and patterns.

12.3 Covered DL Constructors

The aim of this chapter is to provide a benchmark for analyzing which kinds of constructs in a knowledge graph can be recognized by different embedding methods. To that end, we define class labels using different DL constructors. Later on, we apply classification algorithms to analyze how well the differently labeled classes can be separated using different embedding algorithms.

Ingoing and Outgoing Relations All entities that have a particular outgoing or ingoing relation (e.g., *everything that has a location*).

$$\exists r.\top \tag{12.1}$$

$$\exists r^{-1}.\top \tag{12.2}$$

$$\exists r.\top \sqcup \exists r^{-1}.\top \tag{12.3}$$

where r is bound to a particular relation.[1]

Relations to Particular Individuals All entities that have a relation (in any direction) to a particular individual (e.g., *everything that is related to Mannheim*).

$$\exists R.\{e\} \sqcup \exists R^{-1}.\{e\} \tag{12.4}$$

where R is *not* bound to a particular relation. Those relations can also span two (or more[2] hops):

$$\exists R_1.(\exists R_2.\{e\}) \sqcup \exists R_1^{-1}.(\exists R_2^{-1}\{e\}) \tag{12.5}$$

Particular Relations to Particular Individuals All entities that have a particular relation to a particular individual (e.g., *movies directed by Steven Spielberg*).

$$\exists r.\{e\} \tag{12.6}$$

[1] We use r to denote a particular relation, whereas R denotes *any* relation.

[2] For reasons of scalability, we restrict the provided gold standard to two hops.

Qualified Restrictions All entities that have a particular relation to an individual of a given type (e.g., *all people married to soccer players*).

$$\exists r.T \tag{12.7}$$

$$\exists r^{-1}.T \tag{12.8}$$

If types are modeled as a normal relation in the graph (i.e., `rdf:type` is yet another relation), we can reformulate 12.7 to

$$\exists r.(\exists \texttt{rdf:type}.T) \tag{12.9}$$

In that case, it behaves equally to a chained variant of Equation 12.6.

Cardinality Restrictions of Relations All entities that have at least or at most n relations of a particular kind (e.g., *people who have at least two citizenships*). Here, we depict only the *lower bound* variant because the corresponding decision problem is between the two variants (entities that fall below the bound, i.e., adhere to the upper bound, are in the negative example set).[3]

$$\geq 2r.\top \tag{12.10}$$

$$\geq 2r^{-1}.\top \tag{12.11}$$

Qualified Cardinality Restrictions Qualified cardinality restrictions combine qualified restrictions with cardinalities (e.g., people who have published at least two science fiction novels).

$$\geq 2r.T \tag{12.12}$$

$$\geq 2r^{-1}.T \tag{12.13}$$

Table 12.1 sums up the DL constructors for which test cases were built.

12.4 Approach

For the twelve test cases in Table 12.1, we create positive examples (i.e., those which fall into the respective class) and those which do not (under closed-world semantics). For example, for tc01, we would generate a set of positive instances for which $\exists r.\top$ holds and a set of negative instances for which $\nexists r.\top$ holds. We then evaluate how well these two classes can be separated, given the embedding

[3]The fact that most KGs follow the open-world assumption is ignored here.

Table 12.1: Overview of the Test Cases

Test Case	DL Expression
tc01	$\exists r.\top$
tc02	$\exists r^{-1}.\top$
tc03	$\exists r.\top \sqcup \exists r^{-1}.\top$
tc04	$\exists R.\{e\} \sqcup \exists R^{-1}.\{e\}$
tc05	$\exists R_1.(\exists R_2.\{e\}) \sqcup \exists R_1^{-1}.(\exists R_2^{-1}\{e\})$
tc06	$\exists r.\{e\}$
tc07	$\exists r.T$
tc08	$\exists r^{-1}.T$
tc09	$\geq 2r.\top$
tc10	$\geq 2r^{-1}.\top$
tc11	$\geq 2r.T$
tc12	$\geq 2r^{-1}.T$

vectors of the positive and negative instances. For that, we split the examples into a training and testing partition; we train binary classifiers on the training subset of the examples and evaluate their performance on the test subset.

The approach is visualized in Figure 12.2: A gold standard generator generates a set of positive and negative URIs, as well as a fixed train/test split. The approach presented in this chapter allows for generating custom gold standards – however, a contribution of this chapter is also to provide a pre-calculated gold standard. This pre-calculated gold standard can be used to guarantee reproducibility. Officially published gold standards are versioned to allow for future improvements. In this chapter, we present version `v1` of the gold standard.

A user provides embeddings in a simple textual format and provides them together with the training data as input to the evaluator. The evaluator trains multiple classifiers and evaluates them on the selected gold standard using the provided vectors as classification input. The program then calculates multiple statistics in the form of CSV files that can be further analyzed in a spreadsheet program or through data analysis frameworks such as pandas[4]. These analyses help the user to understand how well the provided vectors are performing on a particular DL constructor.

[4] `https://pandas.pydata.org/`

12.4.1 Gold Standard Generator

The gold standard generator is publicly available[5]. It is implemented as a Java maven project. The generator can generate either a DBpedia benchmark (see Subsection 12.5.1) or a synthetic one (see Subsection 12.5.2). Any DBpedia version can be used, the user merely needs to provide a SPARQL endpoint. A comprehensive set of unit tests ensures a high code quality. The generator automatically generates a fixed train-test split for the evaluation framework or any other downstream application. The split is configurable; for the pre-generated gold standards, an 80-20 split is used. The resulting gold standard is balanced – i.e., the number of positives equals the number of negatives – and the train and test partitions are stratified. Hence, any classifier which achieves an accuracy significantly above 50% is capable of learning the test case's problem type from the vectors to some extent.

It is important to note that the generator only needs to be run by users who want to build their own gold standards. The typical user would merely download[6] the official gold standard files online. We recommend using the pre-calculated gold standards to ensure comparability across publications.

12.4.2 Evaluation Framework

The evaluator is publicly available[7] as well together with usage examples. It is implemented in Python and can be easily used in a Jupyter notebook. A comprehensive set of unit tests ensures a high code quality.

The standard user can directly download the gold standard and use the evaluation framework. To test class separability, the evaluation framework currently runs six machine learning classifiers[8] (1) decision trees, (2) naïve Bayes, (3) KNN, (4) SVM, (5) random forest, and (6) a *multilayer perceptron network* (MLP). The framework uses the default configurations of the sklearn library[9].

After training and evaluation, the framework persists multiple CSV files per test case as well as higher-level aggregate CSV files. Examples of such CSV files are a file listing the accuracy per classifier and per test case or a file listing the accuracy of the best classifier per test case. In the case of DBpedia test cases

[5] `https://github.com/janothan/DL-TC-Generator`

[6] *digital object identifier* (DOI): `10.5281/zenodo.6509715`; GitHub link for the latest version. `https://github.com/janothan/DL-TC-Generator/tree/master/results`

[7] `https://github.com/janothan/dl-evaluation-framework`

[8] The evaluation framework is not restricted to the set of classifiers listed here. New classifiers can be easily added if desired.

[9] `https://scikit-learn.org/stable/index.html`

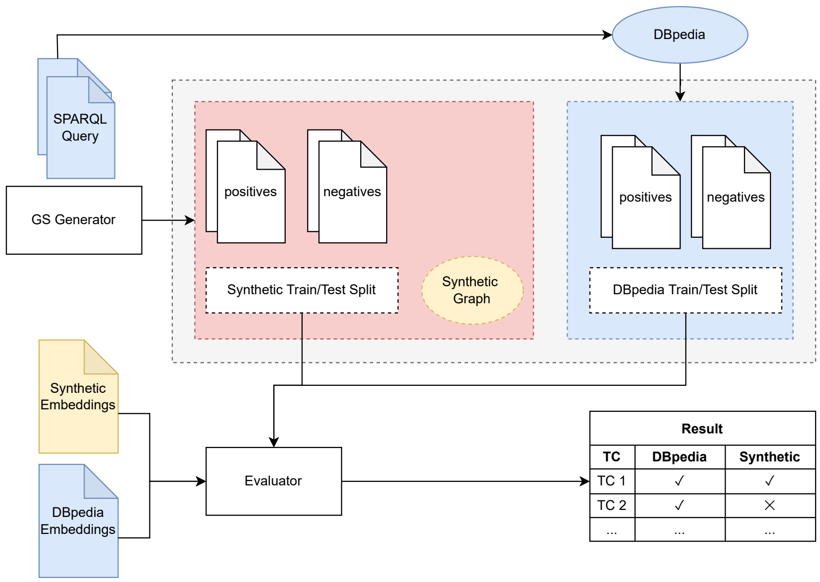


Figure 12.2: Overview of the Approach

where multiple domains are available per test case, the results can be analyzed on the level of each domain separately or in an aggregated manner on the level of the test case.

12.5 Benchmarks

We currently provide two benchmarks, while the framework described above allows for generating customized benchmarks.

12.5.1 DBpedia Benchmark

We use the DBpedia knowledge graph to create test cases.[10] We created SPARQL queries for each test case (see Table 12.1) to generate positives, negatives, and hard negatives. The latter are created by variations such as softening the constraints in the class constructor or switching subject and object in the constraint. For example, for qualified relations, a positive example would be a person playing in a team which is a basketball team. A simple negative example would be

[10] We used DBpedia version 2021-09. The generator can be configured to use any DBpedia SPARQL endpoint if desired.

any person not playing in a basketball team, whereas a hard negative example would be any person playing in a team that is not a basketball team.

Query examples for every test case in the people domain are provided in Tables 12.2 and 12.3. The framework uses slightly more involved queries to vary the size of the result set and to randomize results better.

In total, we used six different domains: people (P), books (B), cities (C), music albums (A), movies (M), and species (S). This setup yields more than 200 hand-written SPARQL queries, which are used to obtain positives, negatives, and hard negatives; they are available online[11] and can be easily extended, e.g., to add an additional domain. For each test case, we created differently sized (50, 500, 5000) balanced test sets.[12]

12.5.2 Synthetic Benchmark

The previous benchmark is realistic and well suited to compare approaches on differently typed DL constructors.

However, the following aspects have to be considered: (1) DBpedia is a large knowledge graph; not every embedding approach can be used to learn an embedding for it (or not every researcher has the computational means to do so, respectively). (2) Depending on the DL constructor and the domain, not enough examples can be found on DBpedia. (3) It cannot be precluded that patterns correlate; therefore, the fact that an embedding approach can learn a particular class can only be an indicator that it *might* learn the underlying constructor pattern, but the results are not conclusive. Correlating properties, type biases for entities, etc., may lead to surprising results in some domains (see Section 12.6.3).

Therefore, we complement the DBpedia-based gold standard with a synthetic benchmark. The idea is to generate a graph that contains the DL constructors (positive and negative) of interest. The graph can be constructed to resemble the DBpedia graph statistically but can be significantly smaller (and contain a sufficient number of positives and negatives), and, by construction, side effects and correlations which exist in DBpedia can be mitigated to a large extent.

The configurable parameters are `numClasses`, `numProperties`, `numInstances`, `branchingFactor`, `maxTriplesPerNode`, and `numNodesInterest` (all parameters are integers). The overall process is depicted in Algorithm 2: First, a class tree with `numClasses` classes is constructed in a way that each class has

[11] `https://github.com/janothan/DL-TC-Generator/tree/master/src/main/resources/queries`

[12] The desired size classes can be configured in the framework.

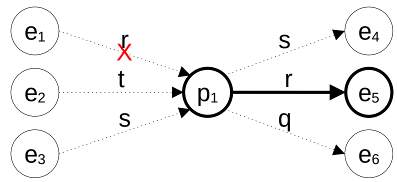


Figure 12.3: Illustration of the instance generation, using the class constructor $\exists r.T$. First, the pattern is instantiated for the positive example p_1 with the edge (p_1, r, e_5). Then, random edges are inserted (dashed lines). The edge (e_1, r, p_1) is removed, because it would turn e_1 into an additional positive example.

at most `branchingFactor` children. Then, `numproperties` properties are generated. Each property is assigned to a range and domain from the class tree, whereby the first property has the root node as domain and range type so that every node can be involved in at least one triple statement. A skew can be introduced so that domain and range refer with a higher probability to a more general class than to a specific one. Lastly, we generate instances and assign them to a class as type, which is depicted in Algorithm 2.

Once the ontology is created, `numNodesInterest` positives and negatives are generated (adhering to domain/range restrictions). Each class constructor is first initialized explicitly for the positive examples. Then, for each entity e in the graph (i.e., positive and negative examples), $rand(n) \in [1, maxTriples\text{-}PerNode]$ random triples are generated, which have e as a subject and adhere to the domain and range definitions, whereby it is checked that no additional positives are created, and no negatives are turned into positives accidentally (see Figure 12.3).

For version `v1` of the gold standard, `numClasses = 760`, `numProperties = 1355`, `numInstances = 10,000`, `branchingFactor = 5`, `maxTriplesPerNode = 11`, and `numNodesInterest = 1000` were chosen. The parameters were chosen to form graphs which are smaller than DBpedia but resemble the DBpedia graph statistically. Therefore, the statistical properties of the DBpedia ontology calculated by Heist et al. [191] were used.

12.6 Exemplary Analysis

In order to demonstrate the use of the DLCC benchmark, we compare two flavors of RDF2vec [442], two flavors of TransE [44], as well as TransR [317] and ComplEx [538] embeddings with respect to their capability of separating the classes in the different datasets.

Algorithm 2 Ontology Creation

```
procedure GENERATECLASSTREE(numClasses, branchingFactor)
    clsURIs ← GENERATEURIS(numClasses)
    root ← RANDOMDRAW(clsURIs)
    i ← 0
    workList ← NEWLIST( )
    result ← NEWTREE( )
    currentURI ← root
    for clsURI in clsURIs do
        if clsURI = root then
            CONTINUE
        end if
        if i = branchingFactor then
            currentURI ← workList.removeFirst()
            i ← 0
        end if
        result.addLeaf(currentURI, clsURI)
        i ← i + 1
        workList.add(clsURI)
    end for
    return result
end procedure

procedure GENERATEPROPERTIES(numProperties, classTree)
    properties ← GENERATEURIS(numProperties)
    for property in properties do
        property.addDomain( DRAWDOMAINRANGE(classTree, 0.25) )
        property.addRange( DRAWDOMAINRANGE(classTree, 0.25) )
    end for
    return properties
end procedure

procedure DRAWDOMAINRANGE(classTree, p)
    result ← classTree.randomClass()
    while Random.nextDouble > p ∧ ¬(classTree.getChildren(result) == ∅) do
        result ← randomDraw(classTree.getChildren(result))
    end while
end procedure

procedure GENERATEINSTANCES(numInstances, classTree)
    instances ← GENERATEURIS(numInstances)
    for instance in instances do
        instance.type(classTree.randomClass())
    end for
    return instances
end procedure
```

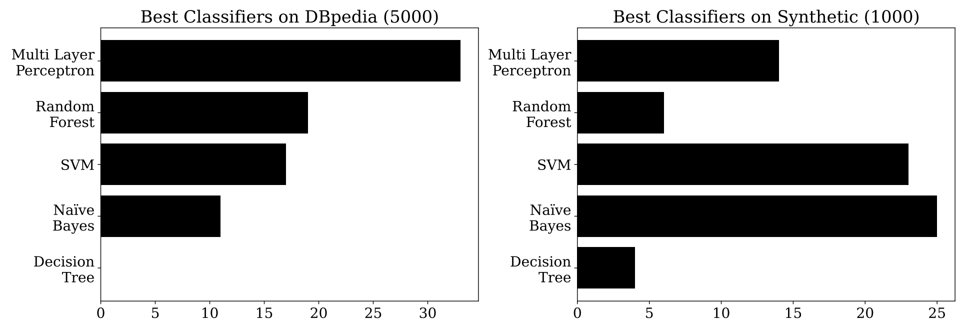


Figure 12.4: Best Classifiers on the DBpedia and Synthetic Gold Standards. It is important to note that the total number of test cases varies between the two gold standards – therefore, two separate plots were drawn.

12.6.1 Configurations

For DBpedia, we use version 2021-09. We train RDF2vec in the variants SG and SG_{oa} [412]. The embedding files are available via KGvec2go [404].[13] For the DBpedia embeddings, we used 500 random, duplicate free walks per entity, with a depth of 4, a window of 5, 5 epochs, and a dimension of 200. We used the same parameters for the synthetic gold standard with the exception of $dimension = 100$ and $walks = 100$ to account for the smaller gold standard size. The embeddings were trained using the jRDF2vec[14] framework [405].

For TransE, we use the variants using the L1 and L2 norm [44]. TransE, TransR, and ComplEx were trained using the DGL-KE framework[15] [614], using the respective default parameters, with 200 dimensions for DBpedia and 100 for the synthetic datasets, as for RDF2vec. The models are publicly available.[16]

12.6.2 Results and Interpretation

The results on the DBpedia gold standard (class size 5,000) and the synthetic gold standard (class size 1,000) are depicted in Tables 12.4 and 12.5. For each model and test case, six classifiers were trained (192 classifiers in total). The tables present the results of the best classifiers. We performed significance tests (approximated one-sided binomial test) for each test case and approach with

[13] `http://data.dws.informatik.uni-mannheim.de/kgvec2go/dbpedia/2021-09/`

[14] `https://github.com/dwslab/jRDF2Vec`

[15] `https://github.com/awslabs/dgl-ke`

[16] `http://data.dws.informatik.uni-mannheim.de/kgvec2go/dbpedia/2021-09/non-rdf2vec/`

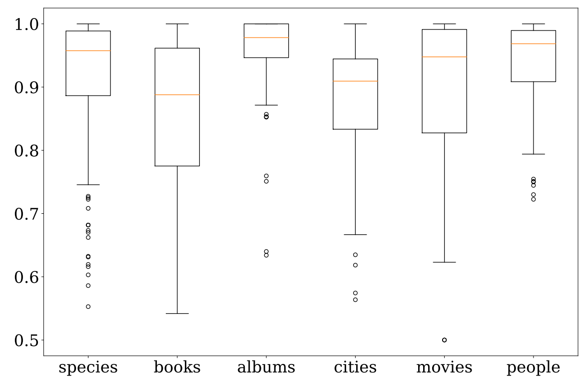


Figure 12.5: Domain Complexity of the DBpedia Gold Standard (Size Class 5000)

$\alpha = 0.05$ to determine whether the accuracy is significantly higher than 0.5 (random guessing). Since multiple classifiers were trained for each test case, we applied a Bonferroni correction of α to account for the multiple testing problem. [452] On the DBpedia gold standard, all results are significant; on the synthetic gold standard, more insignificant results are observed, particularly for TransR and ComplEx.

Figure 12.4 shows the aggregated number of the best classifiers for each embedding on each test case. It is visible that on DBpedia, MLPs work best, followed by random forests and SVMs. On the synthetic gold standard, naïve Bayes works best most of the time, followed by SVMs and MLPs. The differences can partly be explained by the different size classes of the training sets (MLPs and random forests typically work better on more data).

Figure 12.5 depicts the complexity per domain of the DBpedia gold standard in a box-and-whisker plot. The complexity was determined by using the accuracy of the best classifier of each embedding model without hard test cases (since not every domain has an equal amount of hard test cases). We observe that all domain test cases are similarly hard to solve, whereby the albums, people, and species domain are a bit simpler to solve than the books and cities domain.

In general, we can observe that the results on the DBpedia gold standard are much higher than on the synthetic gold standard. While on the DBpedia gold standard, all but five tasks can be solved with an accuracy above 0.9 (although the cases with hard variants are actually harder than the non-hard ones, and all the five problems with a best accuracy below 0.9 are hard cases), the synthetic gold standard has quite a few tasks (tc07–tc12) which are obviously

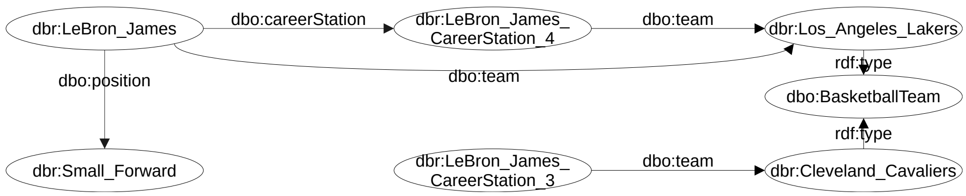


Figure 12.6: Excerpt of DBpedia

much harder. For example, it is hardly possible for any of the approaches to learn classes whose definitions involve cardinalities.

Furthermore, we can observe that it seems easier to predict patterns involving outgoing edges than those involving ingoing edges (cf. tc02 vs. tc01, tc08 vs. tc07, tc10 vs. tc09, tc12 vs. tc11), at least for the DBpedia case. Even though the tasks are very related, this can be explained by the learning process, which often emphasizes outgoing directions: In RDF2vec, random walks are performed in forward direction; similarly, TransE is directed in its training process.

For constructors involving a particular entity (tc04 and tc05), we can observe that RDF2vec is clearly better than embedding approaches for link prediction, at least on the synthetic gold dataset. Those tasks refer to *entity relatedness*, for which RDF2vec has been shown to be more adequate [412, 416]. The picture is more diverse for the other cases.

12.6.3 DBpedia Gold Standard vs. Synthetic Gold Standard

The results reveal strong differences between the gold standards. Many class constructors that are easily learnable on the DBpedia gold standard are hard on the synthetic one. Moreover, the previously reported superiority of RDF2vec$_{oa}$ over standard RDF2vec [399, 412] cannot be observed on the synthetic data.

Figure 12.6 shows an excerpt of DBpedia, which we will use to illustrate these deviations. The instance `dbr:LeBron_James` is a positive example for task tc07 in Table 12.2. At the same time, 95.6% of all entities in DBpedia fulfilling the positive query for positive examples also fall in the class $\exists$`dbo:position`.$\top$ (which is a tc01 problem), but only 13.6% of all entities fulfilling the query for trivial negatives. Hence, on a balanced dataset, this class can be learned with an accuracy of 0.91 by any approach that can learn classes of type tc01. As a comparison to the synthetic dataset shows, the results on the DBpedia test set for tc07 actually overestimate the capability of many embedding approaches to learn classes constructed with a tc07 class constructor. Such correlations are quite frequent in DBpedia but vastly absent in the synthetic dataset.

The example can also explain the advantage of RDF2vec$_{oa}$ on DBpedia. Unlike standard RDF2vec, this approach would distinguish the appearance of node `dbo:team` as a direct edge of `dbr:LeBron_James` as well as an indirect edge connected to `dbr:LeBron_James_CareerStation_`N, where the former denotes the current team, whereas the latter also denote all previous teams. Those subtle semantic differences of distinctive usages of the same property in various contexts also do not exist in the synthetic gold standard. Hence, the order-aware variant of RDF2vec does not have an advantage here.

12.7 Conclusion

In this chapter, we presented DLCC, a resource to analyze embedding approaches in terms of which kinds of classes they are able to represent. DLCC comes with an evaluation framework to easily evaluate embeddings using a reproducible protocol. All DLCC components, i.e., the gold standard, the generation framework, and the evaluation framework, are publicly available. Significant efforts were made to comply with the principles of FAIR (*Findable, Accessible, Interoperable, Reusable*) [574].[17]

We have shown that many patterns using DL class constructors on DBpedia are actually learned by recognizing patterns with other constructors correlating with the pattern to be learned, thus yielding misleading results. This effect is less prominent in the synthetic gold standard. We showed that certain DL constructors, such as cardinality constraints, are particularly hard to learn.

In the next chapter, we perform an extensive evaluation using, among other gold standards, DLCC.

[17] Dataset DOI: `10.5281/zenodo.6509715`; uploaded and indexed via zenodo; published with a permissive license; re-usable; metadata is provided.

TC	Query Positive	Query Negative	Query Negative (hard)

tc01	SELECT DISTINCT(?x)
	WHERE {
	?x a dbo:Person .
	?x dbo:child ?y . }
		SELECT DISTINCT(?x)
		WHERE {
		?x a dbo:Person .
		FILTER(NOT EXISTS {
		?x dbo:child ?z})}
			SELECT DISTINCT(?x) WHERE
			{
			?x a dbo:Person .
			?y dbo:child ?x.
			FILTER(NOT EXISTS {
			?x dbo:child ?z})}

tc02	Analogous to tc01 (inverse case).		

tc03	SELECT DISTINCT(?x)
	WHERE {
	{ ?x a dbo:Person .
	?x dbo:child ?y}
	UNION
	{ ?x a dbo:Person .
	?y dbo:child ?x}}
		SELECT COUNT(?x) WHERE {
		?x a dbo:Person .
		FILTER(NOT EXISTS{
		?x dbo:child ?y}
		AND NOT EXISTS {
		?z dbo:child ?x})}
			–

tc04	SELECT DISTINCT(?x)
	WHERE {
	{ ?x a dbo:Person .
	?x ?y
	dbr:New_York_City}
	UNION
	{ ?x a dbo:Person .
	dbr:New_York_City ?y
	?x}}
		SELECT DISTINCT(?x)
		WHERE {
		?x a dbo:Person .
		FILTER(NOT EXISTS{
		?x ?y
		dbr:New_York_City}
		AND NOT EXISTS {
		dbr:New_York_City ?y
		?x})}
			SELECT DISTINCT(?x) WHERE
			{{
			?x a dbo:Person .
			?x ?y1 ?z .
			?z ?y2
			dbr:New_York_City }
			UNION {
			?x a dbo:Person .
			?z ?y1 ?x .
			dbr:New_York_City ?y2
			?z }
			FILTER(NOT EXISTS
			{?x ?r
			dbr:New_York_City}
			AND NOT EXISTS
			{dbr:New_York_City ?s
			?x})}

tc05	Analogous to tc04 (inverse case).		

tc06	SELECT DISTINCT(?x)
	WHERE {
	?x a dbo:Person .
	?x dbo:birthPlace
	dbr:New_York_City }
		SELECT DISTINCT(?x)
		WHERE {
		?x a dbo:Person .
		FILTER(NOT EXISTS{
		?x dbo:birthPlace
		dbr:New_York_City
		})}
			SELECT DISTINCT(?x) ?r
			WHERE {{
			?x a dbo:Person .
			?x dbo:birthPlace ?y
			.
			dbr:New_York_City ?r
			?x .
			FILTER(?y!=dbr:New_York_City)}
			UNION {
			?x a dbo:Person .
			?x dbo:birthPlace ?y
			.
			?x ?r
			dbr:New_York_City .
			FILTER(?y!=dbr:New_York_City)}}

Table 12.2: Exemplary SPARQL Queries of Test Cases 01-06 for Class `Person` (Table 1 of 2)

TC	Query Positive	Query Negative	Query Negative (hard)

tc07	SELECT DISTINCT(?x) WHERE {
	?x a dbo:Person .
	?x dbo:team ?y .
	?y a dbo:BasketballTeam }
		SELECT DISTINCT(?x) WHERE {
		?x a dbo:Person .
		FILTER(NOT EXISTS{
		?x dbo:team ?y .
		?y a dbo:BasketballTeam})}
			SELECT DISTINCT(?x) WHERE {
			?x a dbo:Person .
			?x dbo:team ?z1 .
			?x ?r ?z2 .
			?z2 a dbo:BaseballTeam
			FILTER(NOT EXISTS{
			?x dbo:team ?y .
			?y a dbo:BasketballTeam })}

tc08	Analogous to tc07 (inverse case).		

tc09	SELECT DISTINCT(?x) WHERE {
	?x a dbo:Person .
	?x dbo:award ?y1.
	?x dbo:award ?y2.
	FILTER(?y1!=?y2)}
		SELECT DISTINCT(?x) WHERE {
		?x a dbo:Person .
		FILTER(NOT EXISTS{
		?x dbo:award ?y1.
		?x dbo:award ?y2.
		FILTER(?y1!=?y2)})}
			SELECT DISTINCT(?x) WHERE {
			?x a dbo:Person .
			?x dbo:award ?y .
			FILTER(NOT EXISTS{
			?x dbo:award ?z.
			FILTER(?y!=?z)})}

tc10	Analogous to tc09 (inverse case).		

tc11	SELECT DISTINCT(?x) WHERE {
	?x a dbo:Person .
	?x dbo:recordLabel ?y1 .
	?y1 a dbo:RecordLabel .
	?x dbo:recordLabel ?y2 .
	?y2 a dbo:RecordLabel .
	FILTER(?y1!=?y2)}
		SELECT DISTINCT(?x) WHERE {
		?x a dbo:Person .
		FILTER(NOT EXISTS{
		?x dbo:recordLabel ?y1 .
		?y1 a dbo:RecordLabel .
		?x dbo:recordLabel ?y2 .
		?y2 a dbo:RecordLabel .
		FILTER(?y1!=?y2)})}
			SELECT DISTINCT(?x) WHERE {
			?x a dbo:Person .
			?x dbo:recordLabel ?y1 .
			?y1 a dbo:RecordLabel .
			FILTER(NOT EXISTS{
			?x dbo:recordLabel ?y2 .
			?y2 a dbo:RecordLabel .
			FILTER(?y1!=?y2)})}

tc12	Analogous to tc11 (inverse case).		

Table 12.3: Exemplary SPARQL Queries of Test Cases 07-12 for Class Person (Table 2 of 2)

Table 12.4: Results on the DBpedia Gold Standard. The best result for each test case is printed in bold. Listed are the results of the best classifier for each task and model.

TC	RDF2vec	RDF2vec$_{oa}$	TransE-L1	TransE-L2	TransR	ComplEx
tc01	0.915	0.937	0.842	**0.947**	0.858	0.862
tc01 hard	0.681	0.891	0.799	**0.916**	0.744	0.651
tc02	0.953	0.961	0.852	**0.970**	0.832	0.853
tc02 hard	0.637	0.780	0.780	**0.849**	0.693	0.608
tc03	0.949	**0.958**	0.821	0.933	0.856	0.874
tc04	0.960	0.968	0.934	0.986	0.973	**0.990**
tc04 hard	0.963	**0.984**	0.814	0.912	0.855	0.935
tc05	0.986	**0.992**	0.867	0.948	0.881	0.905
tc06	0.957	0.963	0.929	0.985	0.976	**0.991**
tc06 hard	0.863	0.936	0.823	0.779	**0.964**	0.933
tc07	0.938	0.955	0.930	**0.987**	0.978	0.966
tc08	0.961	**0.966**	0.898	0.964	0.870	0.888
tc09	0.902	0.901	0.884	**0.938**	0.879	0.883
tc09 hard	0.785	0.793	0.749	**0.848**	0.758	0.776
tc10	0.947	0.958	0.957	**0.984**	0.898	0.931
tc10 hard	0.740	0.737	**0.775**	0.774	0.656	0.739
tc11	0.932	0.897	0.917	**0.960**	0.930	0.946
tc11 hard	0.725	0.737	0.712	**0.806**	0.753	0.723
tc12	0.955	0.938	0.961	**0.984**	0.879	0.894
tc12 hard	0.714	0.717	0.762	**0.765**	0.659	0.710

Table 12.5: Results on the Synthetic Gold Standard. The best result for each test case is printed in bold; statistically insignificant results are printed in italics. Listed are the results of the best classifier for each task and model.

TC	RDF2vec	RDF2vec$_{oa}$	TransE-L1	TransE-L2	TransR	ComplEx
tc01	**0.882**	0.867	0.767	0.752	0.712	0.789
tc02	**0.742**	0.737	0.677	0.677	*0.531*	*0.549*
tc03	0.797	**0.812**	*0.531*	0.581	*0.554*	*0.536*
tc04	**1.000**	0.998	0.790	0.898	0.685	*0.553*
tc05	**0.892**	0.819	0.691	0.774	0.631	0.726
tc06	0.978	0.963	0.898	0.978	0.888	**1.000**
tc07	0.583	0.583	*0.540*	0.615	**0.673**	*0.518*
tc08	0.563	0.585	0.585	**0.613**	*0.540*	*0.523*
tc09	0.610	**0.628**	0.588	*0.543*	*0.525*	*0.545*
tc10	**0.638**	0.623	0.588	0.573	*0.518*	*0.510*
tc11	**0.633**	0.580	0.583	0.590	0.573	0.590
tc12	**0.644**	0.614	0.618	*0.550*	*0.513*	*0.540*

Chapter 13

Comprehensive Evaluation of RDF2vec and its Variants

In Part III, we have made multiple separate contributions in the area of knowledge graph embeddings. In this chapter, we will combine approaches previously presented and carry out an in-depth analysis. Therefore, we reiterate the description logic constructors introduced in the previous chapter and develop hypotheses for each RDF2vec variant and constructor. We then conduct a systematic evaluation of 12 RDF2vec variants together with seven benchmark models and discuss the implications of the results.

The work presented in this chapter is to be submitted for publication as: Portisch, Jan; Paulheim, Heiko. RDF2vec Variants and DL Classes. Semantic Web Journal. 2022. [to be submitted] [415]

13.1 Introduction

In this dissertation, multiple RDF2vec extensions were presented. Generally, three kinds of extensions can be distinguished: (1) Changes in the walk generation algorithm, (2) changes in the embedding algorithm, and (3) other changes. The extensions are presented in the following paragraphs.

Walk Generation Extensions Entity walks and property walks were presented in Chapter 11. Those change the walk generation algorithm in terms of what graph elements are included. We found evidence that e-RDF2vec spaces are rather focused on relatedness, while there is indication that p-RDF2vec spaces cover fine-grained similarity better.

Embedding Algorithm Extensions The classic RDF2vec configuration is based on word2vec. RDF2vec$_{oa}$ (see Chapter 10) uses an order-aware variant of the original word2vec algorithm. The approach has shown to be *consistently* better than the classic RDF2vec configuration in various publications. This was already demonstrated in Chapters 7 and 10.

Other Extensions RDF2vec always generates embedding vectors for an entire knowledge graph. This process can be very expensive for large knowledge graphs and may be even unfeasible for very large knowledge graphs. At the same time, most tasks do not require an embedding for every concept in a knowledge graph. In many cases, the set of required embeddings can be determined ex ante – e.g., *entities of type city* when the task is to regress the score for the quality of living. In such instances, RDF2vec Light (see Chapter 9) can be used. The approach applies the walk generation algorithm only to the predefined entities and thereby reduces the required time for walk generation and training significantly. Experiments showed that the performance is comparable to the more expensive classic variant – particularly in instances where the set of entities is homogeneous.

For brevity, we will not reintroduce the approaches but refer the reader to the corresponding chapters of this dissertation. In Chapter 12, an extensive description logic gold standard was introduced, named DLCC. We will not reiterate the details of the gold standard construction but limit ourselves to a deeper look into the constructors and to hypothesizing which variations may be capable of learning which constructor.

The rest of this chapter is structured as follows: Section 13.2 introduces the RDF2vec configurations together with the configurations of the benchmark models that are evaluated in this chapter. Subsequently, the DL constructors of DLCC are reiterated, whereby we introduce hypotheses in Section 13.3. The central part of this chapter is the evaluation which is carried out in Section 13.4. Therein, results are presented, interpreted, and the hypotheses are validated. We conclude this chapter with Section 13.5.

13.2 RDF2vec Variant Configurations and Benchmark Models

RDF2vec The walk generation processes and the embedding models are independent components of RDF2vec which can be freely combined. In this chapter, we evaluate the following walk generation algorithms:

1. classic walks
2. entity walks
3. predicate walks

We combine these with the following language models:

1. classic word2vec (CBOW and SG)
2. order-aware word2vec (CBOW$_{oa}$ and SG$_{oa}$)

This leads to the following combinations:

1. $RDF2vec$ (original: classic word2vec with classic walks)
2. $RDF2vec_{oa}$ (order-aware word2vec with classic walks)
3. $p-RDF2vec$ (predicate walks with word2vec)
4. $p-RDF2vec_{oa}$ (predicate walks with order-aware word2vec)
5. $e-RDF2vec$ (entity walks with classic word2vec)
6. $e-RDF2vec_{oa}$ (entity walks with order-aware word2vec)

Since all of the above combinations are available in SG and CBOW, this chapter evaluates 12 variants of RDF2vec in total.

We trained 12 RDF2vec embeddings using the configurations listed above. For the DBpedia benchmarks, we use version 2021-09. We generated 500 walks per entity, with a depth of 4, a window of 5, 5 epochs, and a dimension of 200. We used the same parameters for the synthetic gold standard with the exception of $dimension = 100$ and $walks = 100$ to account for the smaller gold standard size. The embeddings were trained using the jRDF2vec[1] framework [405]. The embedding files are publicly available[2] via KGvec2go [404] and can also be used for other downstream tasks.

Benchmark Models We trained DBpedia embeddings using seven benchmark models:

- TransE [44] with L1 norm
- TransE [44] with L2 norm

[1] `https://github.com/dwslab/jRDF2Vec`

[2] `http://data.dws.informatik.uni-mannheim.de/kgvec2go/dbpedia/2021-09/`

- TransR [317]
- ComplEx [538]
- DistMult [538]
- RESCAL [370]
- RotatE [516]

The above-mentioned benchmark models were trained utilizing the DGL-KE framework[3] [614], using the respective default parameters, with 200 dimensions for DBpedia and 100 for the synthetic datasets, as for RDF2vec. The models are publicly available and can also be used for other downstream tasks.[4]

13.3 DL Constructors and Hypotheses

In Section 7.5.1 of Chapter 7, an existing machine learning gold standard was introduced for the evaluation of knowledge graph embeddings. The gold standard is task-oriented, i.e., it gives indication which embedding configuration is suitable for a specific task – however, the gold standard is not suitable to perform a more profound analysis such as *what* is or can be learned. We, therefore, introduced a new gold standard in Chapter 12. Our aim is to provide a benchmark for analyzing which kinds of constructs in a knowledge graph can be recognized by different embedding methods. To that end, in this section, we define class labels using different DL constructors and argue which variants of RDF2vec are capable of learning them. The subsequent equations are identical to those presented in Chapter 12 and are repeated in a shortened format for the convenience of the reader.

13.3.1 Hypotheses in Detail

Ingoing and Outgoing Relations All entities that have a particular outgoing or ingoing relation.

$$\exists r.\top \tag{13.1}$$

$$\exists r^{-1}.\top \tag{13.2}$$

$$\exists r.\top \sqcup \exists r^{-1}.\top \tag{13.3}$$

where r is bound to a particular relation.

[3] `https://github.com/awslabs/dgl-ke`

[4] `http://data.dws.informatik.uni-mannheim.de/kgvec2go/dbpedia/2021-09/non-rdf2vec/`

Hypothesis 1a (13.1) and (13.2) can be learned by $RDF2vec_{oa}$ and by $p-RDF$-$2vec_{oa}$. Non-oa variants cannot properly learn them because they cannot distinguish the two.

Hypothesis 1b (13.3) can be learned by $RDF2vec$, $RDF2vec_{oa}$, $p-RDF2vec$, and $p-RDF2vec_{oa}$.

Use case An exemplary use case would be entity classification. If a relation has a particular domain or range, an embedding vector capturing that information could be used to infer the corresponding class.

Relations to Particular Individuals All entities that have a relation (in any direction) to a particular individual.

$$\exists R.\{e\} \sqcup \exists R^{-1}.\{e\} \tag{13.4}$$

where R is *not* bound to a particular relation. Those relations can also span two (or more[5]) hops:

$$\exists R_1.(\exists R_2.\{e\}) \sqcup \exists R_1^{-1}.(\exists R_2^{-1}\{e\}) \tag{13.5}$$

Hypothesis 2a (13.4) can be learned by $RDF2vec$, $RDF2vec_{oa}$, $e-RDF2vec$, and $e-RDF2vec_{oa}$. Sub-hypothesis: It is possible that the non-oa variants learn it a bit better. However, the non-oa variants will not be able to tell closely related entities (one hop away) from less related ones (more than two hops away).[6]

Hypothesis 2b (13.5) can be learned by $RDF2vec$, $RDF2vec_{oa}$, $e-RDF2vec$, and $e-RDF2vec_{oa}$, as long as the walk length allows for capturing those relations. Sub-hypothesis: It is possible that the non-oa variants learn it a bit better.

Use case An exemplary use case would be capturing entity relatedness. Two entities sharing many connections to a third entity are typically related. This can also be useful in query expansion for information retrieval. The distinction between closely and vaguely related entities (sharing an entity one or two hops away) may be crucial if queries should not be expanded too much.

[5]For reasons of scalability, we restrict the provided gold standard to two hops.

[6]Depending on the entity at hand, the second set might grow very large. For example, in DBpedia, half of the entities are reachable from *New York City* within two hops.

Particular Relations to Particular Individuals All entities that have a particular relation to a particular individual.

$$\exists r.\{e\} \tag{13.6}$$

Hypothesis 3 (13.6) can only be learned properly by $RDF2vec_{oa}$. Non-oa variants cannot distinguish between the two.[7]

Use case An exemplary use case would be capturing entity similarity. For example, two movies that have the same director and some overlapping cast can be considered similar. This can be used, e.g., in recommender systems or other predictive modeling tasks.

Qualified Restrictions All entities that have a particular relation to an individual of a given type.

$$\exists r.T \tag{13.7}$$

$$\exists r^{-1}.T \tag{13.8}$$

If types are included in the graph, then `rdf:type` becomes yet another restriction, and we can reformulate (13.7) to

$$\exists r.(\exists \texttt{rdf:type}.T) \tag{13.9}$$

Therefore, it behaves equally to a chained variant of (13.6), and, given a long enough walk length, should have similar constraints. However, if the related entity has strong domain and range signals, it may be learned just by observing the ingoing and outgoing relations of that entity. In that case, $p - RDF2vec_{oa}$ could also be capable of learning that class to a certain extent.

Hypothesis 4a (13.7) can only be learned properly by $RDF2vec_{oa}$, and, to a certain extent, by $p - RDF2vec_{oa}$.

The second case (13.8) is trickier. Here, the relation to the entity at hand and the type information of the related entity can only appear in two *different* walks, but never together. Hence, we assume:

Hypothesis 4b (13.8) cannot be learned by any $RDF2vec$ variant.

[7] For example: distinguishing people influenced by Leibniz vs. people who influenced Leibniz.

Use case Qualified restrictions are often useful for fine-grained entity classification and thereby capture some aspects of entity similarity. For example, for distinguishing a basketball and a baseball team, it is not sufficient that both have a coach and players, but that those are of the class `BasketballPlayer` or `BaseballPlayer`. If the similarity aspects become rather fine-grained, they may also be used in predictive modeling tasks.

Cardinality Restrictions of Relations All entities that have at least or at most n relations of a particular kind. Here we depict only the *at least* variant because the corresponding decision problem is between the two variants.[8]

$$\geq 2r.\top \tag{13.10}$$

$$\geq 2r^{-1}.\top \tag{13.11}$$

Since RDF2vec is based on single walks, it cannot *directly* learn cardinalities. However, if a relation appears with a higher cardinality, it is occurring in the walks including the corresponding instance more often, making it a stronger signal for the word2vec algorithm.

Hypothesis 5 (13.10) and (13.11) can be learned, at least to a certain extent, by $RDF2vec_{oa}$ and $p-RDF2vec_{oa}$. Non-oa variants cannot distinguish the two cases.[9]

Use case Cardinalities often capture entity similarity aspects not expressed in other restrictions. For example, when comparing two authors in a knowledge graph of publications, both will have published papers (which makes them indistinguishable when only looking at qualified restrictions), but there is still a difference if one has published two and the other has published two hundred papers.

Qualified Cardinality Restrictions Qualified cardinality restrictions combine qualified restrictions with cardinalities.

$$\geq 2r.T \tag{13.12}$$

$$\geq 2r^{-1}.T \tag{13.13}$$

[8] The fact that most knowledge graphs follow the open-world assumption is ignored here.

[9] For example: distinguishing someone who has been influenced by more than two people vs. someone who has influenced more than two people.

Since this is a combination of qualified restrictions and cardinality restrictions, we hypothesize that it can be captured by RDF2vec variants that can handle both of them:

Hypothesis 6a (13.12) can be learned to a certain extent by $RDF2vec_{oa}$.

Hypothesis 6b (13.13) cannot be learned by any variant of $RDF2vec$.

Use case Just like qualified restrictions and cardinality restrictions, these restrictions capture finer-grained aspects of entity similarity and are thus useable both for fine-grained entity classification and for predictive modeling tasks.

13.3.2 Overview of Hypotheses

Table 13.1 summarizes our hypotheses together with the test cases that were developed (see Section 13.3). For CBOW vs. SG, we have no particular hypothesis.

13.4 Evaluation

13.4.1 Results on the ML Gold Standard

The results for the ML gold standard (introduced in Section 7.5.1 of Chapter 7) are provided in Tables 13.2 (classification and clustering), 13.3 (regression and semantic analogies), and 13.4 (entity relatedness and document similarity).

Classification On the classification task, it can be observed that the order-aware RDF2vec variants lead – with few exceptions – to generally better or the same results. It is further observable that the SG configuration outperforms the CBOW configuration. Within the RDF2vec group, the classic and the entity variant achieve the best results. Concerning the benchmark models, the overall best results are achieved using TransE with L2; RDF2vec SG configurations are close to the best scores. Within the RDF2vec family, the classic configurations perform overall better than the other configurations.

Clustering Concerning the benchmark models, the overall best results are obtained using TransE with L2. Concerning the RDF2vec configurations, the results are rather inconclusive.

	Test Case	DL Expression	$RDF2vec$	$RDF2vec_{oa}$	$p-RDF2vec$	$p-RDF2vec_{oa}$	$e-RDF2vec$	$e-RDF2vec_{oa}$
H1a	tc01	$r.\top$		✓		✓		
H1a'	tc02	$r^{-1}.\top$		✓		✓		
H1b	tc03	$\exists r.\top \sqcup \exists r^{-1}.\top$	✓	✓	✓	✓		
H2a	tc04	$\exists R.\{e\} \sqcup \exists R^{-1}.\{e\}$	✓	✓			✓	✓
H2b	tc05	$\exists R_1.(\exists R_2.\{e\}) \sqcup \exists R_1^{-1}.(\exists R_2^{-1}\{e\})$	✓	✓			✓	✓
H3	tc06	$r.\{e\}$		✓				
H4a	tc07	$\exists r.T$		✓		(✓)		
H4b	tc08	$\exists r^{-1}.T$						
H5	tc09	$\geq 2r.\top$		(✓)		(✓)		
H5'	tc10	$\geq 2r^{-1}.\top$		(✓)		(✓)		
H6a	tc11	$\geq 2r.T$		(✓)				
H6b	tc12	$\geq 2r^{-1}.T$						

Table 13.1: Overview of Hypotheses and Test Cases

Regression Again, on the regression tasks, improvements can be observed for the order-aware variants, which outperform non-order-aware variants. Again, TransE with L2 regularization achieves the best results in most cases with RDF-2vec SG_{oa} being the runner-up.

Semantic Analogies On the semantic analogies task, the classic RDF2vec variant with SG configuration performs best in three out of four cases. Improvements by the order-aware variants cannot be observed on this task. RESCAL and RotatE perform comparatively poorly on this task.

Entity Relatedness and Document Similarity On the entity relatedness task, the e-RDF2vec variants perform comparatively well, with e-RDF2vec SG being the best model. This is intuitive since the e-RDF2vec variant can be expected to pick up the notion of entity relatedness best. On the document similarity task, it can be observed that the p-RDF2vec variant outperforms the other RDF2vec configurations. Again, this finding is intuitive since the configuration is expected to pick up fine-grained entity similarity best.

13.4.2 Results on DLCC

As outlined in the previous chapter, the DLCC benchmarks are balanced. That means that a performance significantly above 50% indicates that the model learns the constructor to some extent. It is important to highlight that Tables 13.5 and 13.6 state the best results out of six classifiers. In order to determine whether the stated result for an embedding configuration for a particular test case is significant, we performed an approximated one-sided binomial significance test with $\alpha = 0.05$. Since multiple classifiers were trained for each test case, we applied the conservative Bonferroni correction [452] of α to account for the multiple testing problem.

DBpedia Benchmark The results on the DLCC DBpedia benchmark (class size 5,000) are reported in Table 13.5. For each model, six classifiers were trained resulting in more than 2,000 classification results. At first sight, it is quickly observable that all models can learn all tasks comparatively well; all results are statistically significant. It is, furthermore, visible that the hard test cases are indeed harder.

On the DBpedia gold standard, it can be seen that s-RDF2vec is rather suitable for similarity-based constructors (tc1, tc2, tc3, tc6) while e-RDF2vec is doing better on relatedness-oriented constructors (tc04, tc05).

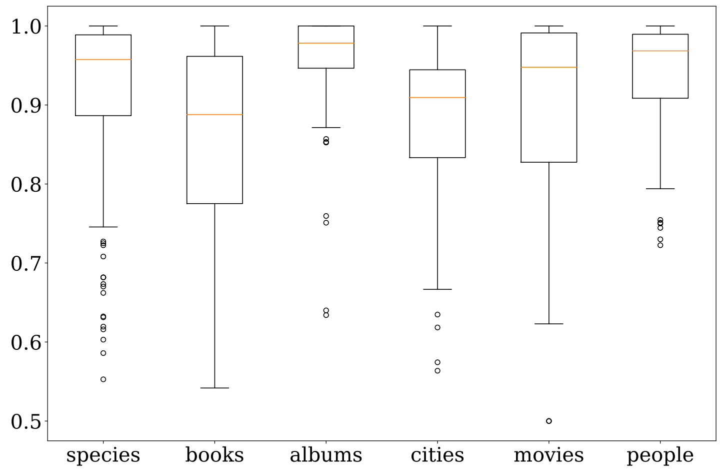


Figure 13.1: Domain Complexity of the DBpedia Gold Standard (Size Class 5000)

Moreover, we can observe that it seems easier to predict patterns involving outgoing edges than those involving ingoing edges (cf. tc02 vs. tc01, tc08 vs. tc07, tc10 vs. tc09, tc12 vs. tc11). Even though the tasks are very related, this can be explained by the learning process, which often emphasizes outgoing directions: In RDF2vec, random walks are performed in forward direction; similarly, TransE is directed in its training process. On the DBpedia benchmark, it is observable that the TransE-L2 configuration performs, overall, best scoring the first place in 9 out of 20 instances.

Figure 13.1[10] depicts the complexity per domain of the DBpedia gold standard in a box-and-whisker plot. The complexity was determined by using the accuracy of the best classifier of each embedding model without hard test cases (since not every domain has an equal amount of hard test cases). We observe that all domain test cases are similarly hard to solve whereby the albums, people, and species domain are a bit simpler to solve than the books and cities domain.

Synthetic Benchmark The results on the synthetic benchmark (class size 1,000) are reported in Table 13.6. Again, for each model, six classifiers were trained, whereby only the best performing classifiers' results are discussed. RDF2vec configurations are performing very well on this gold standard, being the best

[10] Figure 13.1 is calculated according to the same method as Figure 12.5 of the previous chapter. While the overall trend observed in the previous chapter can be confirmed, it is important to note that there are differences in the figures due to significantly more classifiers being used in this chapter.

performing embedding model in 10 out of 12 cases. In terms of the best RDF2vec configuration, the classic CBOW variant achieves the best results in five cases.

The intuition that s-RDF2vec is doing better on similarity-based constructors while e-RDF2vec is doing better on relatedness-oriented constructors can again be observed: This time, e-RDF2vec is not able to learn tc02 and tc03 which is intuitive since the approach does not learn the notion of predicate types. On tc04 and tc05, on the other hand, the e-RDF2vec approach performs very well (much better than s-RDF2vec).

The best benchmark model is RESCAL. RotatE produces more often insignificant results than significant results – the model outperforms pure guessing in only a third of the cases.

The overall most complicating test case is tc07. Similarly, more than half of the models are not significantly able to learn tc08. This is remarkable since the constructors can be almost perfectly predicted on the corresponding DBpedia gold standards. Hence, we can reason that handling qualified restrictions is a very intricate task. The second hardest group of tasks is those involving cardinalities (tc10-tc12).

DBpedia Benchmark vs. Synthetic Benchmark The comparison of the DBpedia and the synthetic benchmark is particularly intriguing. We can see that the synthetic benchmark is much harder to solve since the results are drastically lower in most cases. While there are no insignificant results on the DBpedia gold standard, there are many for the synthetic one – particularly when it comes to the benchmark models. Any class constructors that are easily learnable on the DBpedia gold standard are hard on the synthetic one. Moreover, the previously reported superiority of RDF2vec$_{oa}$ over standard RDF2vec [399, 412] cannot be observed on the synthetic data.

These observations are consistent with the ones made in the previous chapter. The variations are due to the existence of correlations in the DBpedia test cases and the lack of subtle usage differences in the synthetic test cases. For a detailed explanation, we refer the reader to the reasoning in Subsection 12.6.3 for details.

The comparison between DBpedia and synthetic test cases reveals that most models are not *actually* learning the description logic constructor but instead are picking up cross-correlations very well.

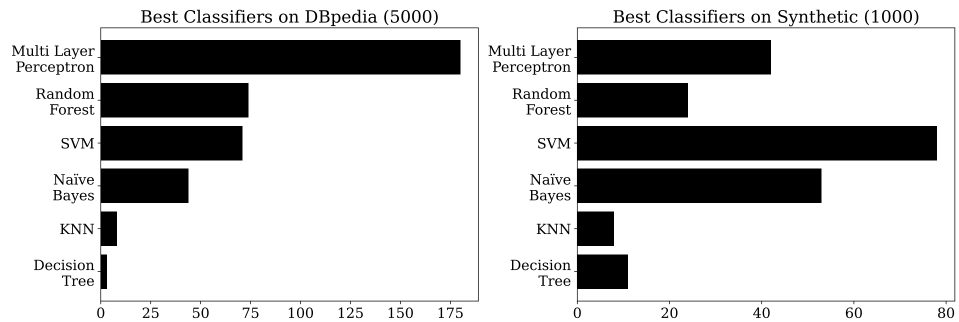


Figure 13.2: Best DLCC Classifiers on DBpedia and Synthetic. It is important to note that the total number of test cases varies between the two gold standards – therefore, two separate plots were drawn.

Figure 13.2[11] shows the aggregated number of the best classifiers for each embedding on each test case. It is visible that on DBpedia, MLPs work best, followed by random forests and SVMs. On the synthetic gold standard, SVMs work best most of the time followed by naïve Bayes and MLPs. The differences can partly be explained by the different size classes of the training sets (MLPs and random forests typically work better on more data).

13.4.3 Discussion of the Hypotheses

In this section, the hypotheses stated in Section 13.3 are verified and discussed. We treat the hypotheses as non-exclusive. That is, we accept the hypotheses if there is significance that the stated configurations can indeed learn the description logic constructor; in cases where we hypothesize that the constructor can be learned by neither configuration, we reject the hypothesis if a single approach can learn the constructor. However, we do not want to mislead the reader: We underestimated which other configurations are also capable of learning constructors. We, therefore, encourage the reader to not just check which hypotheses are accepted but also to follow the reasoning. Hence, we use the hypotheses as structured discussion points for a deeper analysis.

[11] Figure 13.2 is calculated following the same method as Figure 12.4 of the previous chapter. Note, however, that there are differences since the figure in this chapter involves significantly more classifiers due to the large number of configurations. The overall trend, which can be seen in Figure 12.4, can be confirmed here.

Hypothesis 1 The hypothesis can be accepted. However, it has to be acknowledged that – with the exception of e-RDF2vec – all RDF2vec configurations perform rather well.

Hypothesis 1a/1a' In fact, out of all RDF2vec configurations, RDF2vec$_{oa}$ and p-RDF2vec$_{oa}$ are performing best on tc01 and tc02 for DBpedia. On the synthetic gold standard, this can similarly be observed, albeit the improvement of the order-aware configuration does not account for all RDF2vec variants. The previously discussed directionality bias in the training likely leads to better results on tc01 compared to tc02.

Hypothesis 1b Particularly on tc03 (synthetic), it is visible that e-RDF2vec cannot really learn the constructor: None of the configurations performs significantly better than random guessing. As expected, once the directionality restriction is lifted, the results generally improve.

Hypothesis 2 The hypothesis can be accepted. Again, however, it has to be noted that even the p-RDF2vec configuration performs well on tc04 and tc05. While performing worse than the other configurations, p-RDF2vec is still able, to a small extent, to learn the constructor, as witnessed by the results on the synthetic gold standard. The sub-hypotheses, stating that non-order-aware variants perform better than order-aware variants, can be rejected. On DBpedia, significant increases can be observed when using the order-aware variant. Although there are multiple cases of non-oa variants slightly outperforming order-aware variants on the synthetic gold standard, there is, overall, also not enough evidence to accept this hypothesis.

Hypothesis 3 The hypothesis can be accepted. Particularly on the hard tc06 test case, the classic RDF2vec configuration with the order-aware training component performs best. It has to be admitted, though, that on the synthetic gold standard, the e-RDF2vec variant performs very well. A reason for this may be the fact that domain/range restrictions can also be found in the synthetic gold standard, which allows to reason on a likely predicate given an object entity.

Hypothesis 4 The hypothesis can only be partially accepted.

Hypothesis 4a The RDF2vec$_{oa}$ configuration is indeed the best performing configuration on tc07 for both gold standards. A look at the synthetic gold standard reveals that p-RDF2vec cannot learn this constructor.

Hypothesis 4b While we assumed that this constructor could not be learned by any configuration, there is indication that, at least to a small extent, classic and p-RDF2vec can learn to recognize the constructor. In both cases, the p-RDF2vec$_{oa}$ configuration achieves the overall best result. The improvement of the order-aware component can be explained since only this component can detect the inverse usage of the relationship.

Hypothesis 5 The hypothesis can be accepted. On DBpedia, p-RDF2vec and classic RDF2vec can learn cardinality restrictions. On the synthetic gold standard, this is only true for RDF2vec classic and CBOW p-RDF2vec configurations. From the rather low score (in the 60ies in terms of accuracy), it can be seen that learning cardinality is rather hard.

Hypothesis 6 This hypothesis can only partially be accepted since multiple configurations are capable of learning tc12. What can be concluded when comparing hypothesis 6 to hypothesis 5 is that the addition of the type restriction makes the test cases harder to solve: This can be seen when comparing the scores for tc09 versus tc11 and tc10 versus tc12. e-RDF2vec can surprisingly learn the constructors on DBpedia (even well) – but a look at the synthetic gold standard reveals that it can neither learn tc11 nor tc12 when correlations are mostly removed. This finding is intuitive since e-RDF2vec is unaware of the actual predicates within a graph (it is merely aware of their existence).

13.5 Conclusion

In this chapter, we presented an extensive evaluation of 12 RDF2vec variants and benchmark models using default benchmarks and DLCC, a newly introduced benchmark for description logic constructors.

We have shown that many patterns using DL class constructors on DBpedia are actually learned by recognizing patterns with other constructors correlating with the pattern to be learned, thus yielding misleading results. This effect is less prominent in the synthetic gold standard. We showed that certain DL constructors, particularly qualified restrictions and cardinality constraints, are particularly hard to learn.

Table 13.2: ML Results for Classification and Clustering

Approach	Classification (ACC)					Clustering (ACC)			
	AAUP	Cities	Forbes	Metacritic Albums	Metacritic Movies	Cities and Countries (2k)	Cities and Countries	Cities, Albums Movies, AAUP, Forbes	Teams
RDF2vec SG	0.706	0.818	**0.623**	0.586	0.726	0.789	0.587	0.829	0.909
RDF2vec SG_{oa}	0.713	0.803	0.605	0.585	0.716	0.9	0.76	0.854	0.931
RDF2vec CBOW	0.643	0.725	0.575	0.536	0.549	0.52	0.783	0.547	0.94
RDF2vec $CBOW_{oa}$	0.69	0.723	0.6	0.532	0.626	0.917	0.72	0.652	0.925
p-RDF2vec SG	0.564	0.606	0.581	0.634	0.61	0.605	0.687	0.598	**0.941**
p-RDF2vec SG_{oa}	0.623	0.677	0.61	0.632	0.66	0.52	0.782	0.798	0.938
p-RDF2vec CBOW	0.551	0.501	0.56	0.569	0.535	0.637	0.787	0.663	0.94
p-RDF2vec $CBOW_{oa}$	0.612	0.707	0.578	0.667	0.663	0.733	0.728	0.748	0.58
e-RDF2vec SG	0.696	0.77	0.608	0.596	0.724	0.726	0.749	0.759	0.889
e-RDF2vec SG_{oa}	**0.717**	0.743	0.605	0.583	0.732	0.726	0.766	0.828	0.926
e-RDF2vec CBOW	0.703	0.75	0.612	0.564	0.686	0.668	0.82	0.557	0.916
e-RDF2vec $CBOW_{oa}$	0.69	0.702	0.6	0.584	0.676	0.66	0.745	0.719	0.931
TransE-L1	0.639	0.716	0.572	0.624	0.645	0.933	0.93	0.901	0.835
TransE-L2	0.668	**0.827**	0.61	**0.668**	**0.76**	**0.94**	**0.939**	**0.906**	0.893
TransR	0.637	0.775	0.576	0.619	0.715	0.929	0.917	0.753	0.816
RotatE	0.628	0.653	0.542	0.582	0.573	0.821	0.641	0.76	0.688
RESCAL	0.653	0.755	0.596	0.622	0.689	0.933	0.927	0.894	0.835
DistMult	0.637	0.689	0.577	0.634	0.678	0.868	0.896	0.859	0.814
ComplEx	0.628	0.756	0.585	0.632	0.7	0.897	0.909	0.859	0.815

Table 13.3: ML Results for Regression and Semantic Analogies

Approach	Regression (RMSE)					Semantic Analogies (ACC)			
	AAUP	Cities	Forbes	Metacritic Albums	Metacritic Movies	capital country entities	all capital country entities	currency entities	city state entities
RDF2vec SG	65.985	15.375	36.545	15.288	20.215	**0.957**	0.905	**0.574**	**0.609**
RDF2vec SG_{oa}	**63.814**	12.782	36.05	15.903	20.42	0.864	0.857	0.535	0.578
RDF2vec CBOW	77.25	18.963	39.204	15.812	24.238	0.81	0.594	0.338	0.507
RDF2vec $CBOW_{oa}$	66.473	19.287	37.067	15.705	23.362	0.789	0.758	0.447	0.442
p-RDF2vec SG	80.275	20.322	37.146	15.178	23.235	0.008	0.014	0.006	0.009
p-RDF2vec SG_{oa}	72.61	17.214	36.374	14.869	22.402	0.091	0.073	0.076	0.048
p-RDF2vec CBOW	96.248	24.743	37.947	15.0	23.979	0.0	0.002	0.002	0.0
p-RDF2vec $CBOW_{oa}$	77.895	20.334	38.952	16.679	22.071	0.036	0.052	0.085	0.036
e-RDF2vec SG	67.337	17.017	38.589	15.573	20.436	0.794	0.657	0.309	0.459
e-RDF2vec SG_{oa}	65.429	16.913	38.558	15.785	20.258	0.747	0.591	0.193	0.484
e-RDF2vec CBOW	70.482	17.29	39.867	15.574	23.348	0.66	0.359	0.198	0.25
e-RDF2vec $CBOW_{oa}$	69.292	20.798	36.313	14.64	22.518	0.397	0.592	0.297	0.361
TransE-L1	82.007	16.485	37.465	14.652	22.796	0.901	0.909	0.09	0.345
TransE-L2	64.386	**12.301**	36.454	**13.689**	**19.765**	0.874	0.884	0.39	0.321
TransR	85.084	13.436	38.067	14.581	20.624	0.923	**0.925**	0.136	0.398
RotatE	83.21	20.869	38.713	14.949	23.9	0.676	0.515	0.0	0.237
RESCAL	68.589	16.383	35.875	14.608	21.562	0.395	0.372	0.0	0.161
DistMult	73.205	17.65	36.737	14.213	21.292	0.779	0.856	0.001	0.295
ComplEx	75.846	15.33	**35.689**	14.236	21.041	0.609	0.829	0.004	0.29

Table 13.4: ML Results for Entity Relatedness and Document Similarity

Approach	Entity Relatedness (Kendall Tau)	Document Similarity (Harmonic Mean)
RDF2vec SG	0.747	0.237
RDF2vec SG_{oa}	0.716	0.23
RDF2vec CBOW	0.611	0.283
RDF2vec $CBOW_{oa}$	0.547	0.209
p-RDF2vec SG	0.432	0.193
p-RDF2vec SG_{oa}	0.768	0.382
p-RDF2vec CBOW	0.568	0.296
p-RDF2vec $CBOW_{oa}$	0.737	0.256
e-RDF2vec SG	**0.832**	0.275
e-RDF2vec SG_{oa}	0.8	0.25
e-RDF2vec CBOW	0.726	0.17
e-RDF2vec $CBOW_{oa}$	0.779	0.111
TransE-L1	0.632	0.388
TransE-L2	0.537	0.398
TransR	0.589	**0.484**
RotatE	0.432	0.467
RESCAL	0.558	0.358
DistMult	0.432	0.406
ComplEx	0.589	0.387

Table 13.5: Results on the DBpedia Gold Standard. The best results are printed in bold.

TC	SG	SG_{oa}	CBOW	$CBOW_{oa}$	s-SG	s-SG_{oa}	s-CBOW	s-$CBOW_{oa}$	e-SG	e-SG_{oa}	e-$CBOW_{oa}$	e-$CBOW_{oa}$	TransE-L1	TransE-L2	TransR	DistMult	ComplEx	RESCAL	RotatE
tc01	0.915	0.937	0.778	0.870	0.907	0.933	0.780	0.924	0.845	0.860	0.840	0.840	0.842	0.947	0.858	0.874	0.862	**0.966**	0.768
tc01 hard	0.681	0.891	0.637	0.891	0.627	0.903	0.576	0.894	0.644	0.651	0.659	0.659	0.799	**0.916**	0.744	0.646	0.651	0.830	0.618
tc02	0.953	0.961	0.865	0.956	0.930	0.972	0.901	**0.974**	0.883	0.895	0.906	0.906	0.852	0.970	0.832	0.859	0.853	0.908	0.737
tc02 hard	0.637	0.780	0.618	0.774	0.628	0.828	0.583	0.838	0.623	0.628	0.607	0.607	0.780	**0.849**	0.693	0.622	0.608	0.729	0.649
tc03	0.949	**0.958**	0.846	0.905	0.913	0.956	0.800	0.938	0.883	0.900	0.886	0.886	0.821	0.933	0.856	0.894	0.874	0.943	0.780
tc04	0.960	0.968	0.705	0.872	0.877	0.908	0.659	0.873	0.965	0.969	0.915	0.915	0.934	0.986	0.973	0.984	**0.990**	0.990	0.862
tc04 hard	0.963	0.984	0.674	**0.992**	0.725	0.828	0.583	0.782	0.938	0.990	0.983	0.983	0.814	0.912	0.855	0.917	0.935	0.918	0.789
tc05	0.986	0.992	0.772	0.906	0.869	0.899	0.719	0.870	0.990	**0.995**	0.931	0.931	0.867	0.948	0.881	0.907	0.905	0.908	0.802
tc06	0.957	0.963	0.698	0.850	0.876	0.903	0.641	0.857	0.960	0.969	0.928	0.928	0.929	0.985	0.976	0.985	**0.991**	0.990	0.866
tc06 hard	0.863	0.936	0.604	0.908	0.708	0.770	0.559	0.745	0.699	0.708	0.650	0.650	0.823	0.779	**0.964**	0.882	0.933	0.964	0.819
tc07	0.938	0.955	0.742	0.785	0.895	0.924	0.726	0.863	0.946	0.946	0.859	0.859	0.930	**0.987**	0.978	0.929	0.966	0.945	0.847
tc08	0.961	0.966	0.891	0.896	0.911	**0.968**	0.841	0.951	0.904	0.914	0.925	0.925	0.898	0.964	0.870	0.856	0.888	0.875	0.831
tc09	0.902	0.901	0.773	0.858	0.819	0.858	0.726	0.832	0.874	0.884	0.840	0.840	0.884	**0.938**	0.879	0.877	0.883	0.929	0.780
tc09 hard	0.785	0.793	0.659	0.751	0.698	0.741	0.600	0.712	0.777	0.782	0.744	0.744	0.749	**0.848**	0.758	0.774	0.776	0.820	0.676
tc10	0.947	0.958	0.918	0.905	0.924	0.975	0.852	0.969	0.911	0.912	0.925	0.925	0.957	**0.984**	0.898	0.918	0.931	0.927	0.878
tc10 hard	0.740	0.737	0.716	0.711	0.610	0.679	0.569	0.652	0.715	0.718	0.729	0.729	**0.775**	0.774	0.656	0.743	0.739	0.713	0.665
tc11	0.932	0.897	0.865	0.780	0.884	**0.991**	0.808	0.954	0.928	0.972	0.921	0.921	0.917	0.960	0.930	0.889	0.946	0.954	0.838
tc11 hard	0.725	0.737	0.687	0.676	0.684	0.707	0.631	0.707	0.763	0.734	0.641	0.641	0.712	**0.806**	0.753	0.666	0.723	0.726	0.638
tc12	0.955	0.938	0.888	0.909	0.900	0.971	0.830	0.965	0.893	0.905	0.904	0.904	0.961	**0.984**	0.879	0.912	0.894	0.927	0.834
tc12 hard	0.714	0.717	0.712	0.699	0.628	0.637	0.545	0.628	0.690	0.713	0.715	0.715	0.762	**0.765**	0.659	0.714	0.710	0.701	0.652

Table 13.6: Results on the Synthetic Gold Standard. The best result for each test case is printed in bold. Listed are the results of the best classifier for each task and model.

TC	SG	SG_{oa}	CBOW	$CBOW_{oa}$	s-SG	s-SG_{oa}	s-CBOW	s-$CBOW_{oa}$	e-SG	e-SG_{oa}	e-CBOW	e-$CBOW_{oa}$	TransE-L1	TransE-L2	TransR	DistMult	ComplEx	RESCAL	RotatE
tc01	0.882	0.867	0.566	0.877	0.870	0.842	0.802	0.847	0.774	0.757	0.752	0.727	0.767	0.752	0.712	0.837	0.789	**0.895**	0.769
tc02	0.742	0.737	0.769	0.732	**0.822**	0.734	0.769	0.754	0.536	0.529	0.536	0.529	0.677	0.677	0.531	0.584	0.549	0.689	0.546
tc03	0.797	0.812	**0.927**	0.774	0.794	0.709	0.784	0.742	0.526	0.526	0.561	0.519	0.531	0.581	0.554	0.556	0.536	0.634	0.541
tc04	**1.000**	0.998	0.990	0.998	0.568	0.588	0.608	0.628	**1.000**	0.995	**1.000**	0.998	0.790	0.898	0.685	0.588	0.553	0.528	0.728
tc05	**0.892**	0.819	0.889	0.819	0.631	0.648	0.681	0.648	0.832	0.819	0.882	0.791	0.691	0.774	0.631	0.658	0.726	0.608	0.646
tc06	0.978	0.963	0.898	0.965	0.800	0.828	0.748	0.820	0.970	0.968	0.905	0.965	0.898	0.978	0.888	**1.000**	**1.000**	**1.000**	0.955
tc07	0.583	0.583	0.575	0.555	0.553	0.553	0.535	0.540	0.543	0.525	0.498	0.518	0.540	0.615	**0.673**	0.565	0.518	0.550	0.508
tc08	0.563	0.585	0.555	0.583	0.635	**0.638**	0.568	0.618	0.525	0.533	0.553	0.540	0.585	0.613	0.540	0.535	0.523	0.533	0.535
tc09	0.610	0.628	**0.648**	0.605	0.563	0.550	0.605	0.590	0.550	0.535	0.508	0.528	0.588	0.543	0.525	0.525	0.545	0.638	0.538
tc10	0.638	0.623	**0.665**	0.600	0.548	0.560	0.633	0.565	0.593	0.565	0.568	0.515	0.588	0.573	0.518	0.525	0.510	0.580	0.533
tc11	0.633	0.580	**0.668**	0.575	0.573	0.555	0.580	0.553	0.550	0.545	0.540	0.545	0.583	0.590	0.573	0.518	0.590	0.625	0.538
tc12	0.644	0.614	**0.657**	0.638	0.563	0.565	0.590	0.640	0.541	0.568	0.560	0.524	0.618	0.550	0.513	0.553	0.540	0.578	0.533

Chapter 14

RDF2vec for Ontology Matching

The previous chapters of this part introduced the topic of knowledge graph embeddings, addressed the issue of embedding accessibility, presented improvements to existing approaches, and provided rich comparative analyses of embedding spaces. While there are many applications for knowledge graph embeddings – particularly RDF2vec embeddings [438] – we will focus in this last chapter of Part III on an application that is so far rarely addressed in the literature: Ontology and knowledge graph matching.

14.1 Using RDF2vec for Matching

Generally, two options for RDF2vec-based matching can be distinguished:

1. Matching Through Internal Embedding

2. Matching Through External Embedding

The two approaches are presented conceptually in more detail in the following subsections.

14.1.1 Matching Through Internal Embedding

For this approach, the embedding is trained on the ontologies or knowledge graphs to be matched. Within the matching operation, these vector representations of the elements that are to be matched are used.

It is important to emphasize that one major challenge for this approach is that the ontologies/graphs to be matched are typically not yet aligned. Therefore, when applying an embedding operation, we obtain two separate embed-

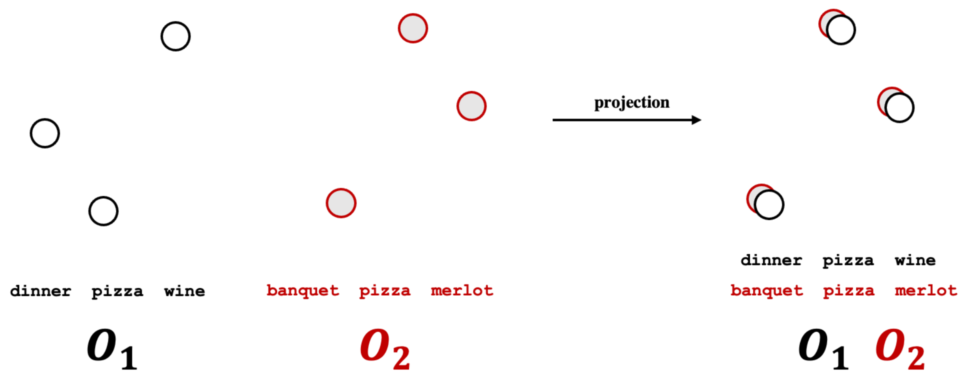


Figure 14.1: Matching Through Internal Embedding – Projection Example. The initial situation is depicted on the left side: There are two separate embedding spaces. After the projection (right side), the concept vectors are aligned.

ding spaces[1] (even if the graphs are merged). A comparison among vectors from different embedding spaces consequently does not yield any useful results.

One option to match with internal embeddings is to transform embeddings from one frame of reference into the other[2]. Once the embeddings of two ontologies are available in the same frame of reference, we can work with meaningful vector operations such as applying the cosine or Euclidean distance to determine similarity. This process is conceptually depicted in Figure 14.1. In this dissertation, two such approaches are presented:

- *RDF2vec vector projections via linear projections* were presented in Subsection 6.5.1 of Chapter 6.
- *RDF2vec vector projections via absolute orientations* are presented in Section 14.2 of this chapter.

14.1.2 Matching Through External Embedding

Matching through external embedding requires the availability of a broad *external knowledge resource* (BK). Rather than embedding the ontologies or graphs that are to be matched, the external resource is embedded. Once a vector representation is obtained for each element $e \in BK$, a three-step matching process

[1] Strictly speaking, both ontologies are embedded in the same space ($\mathbb{R}^{\Delta}$) but have different frames of reference in this space, which makes their values a priori not comparable.

[2] Alternatively, both embedding spaces could be transformed in a joint third frame of reference.

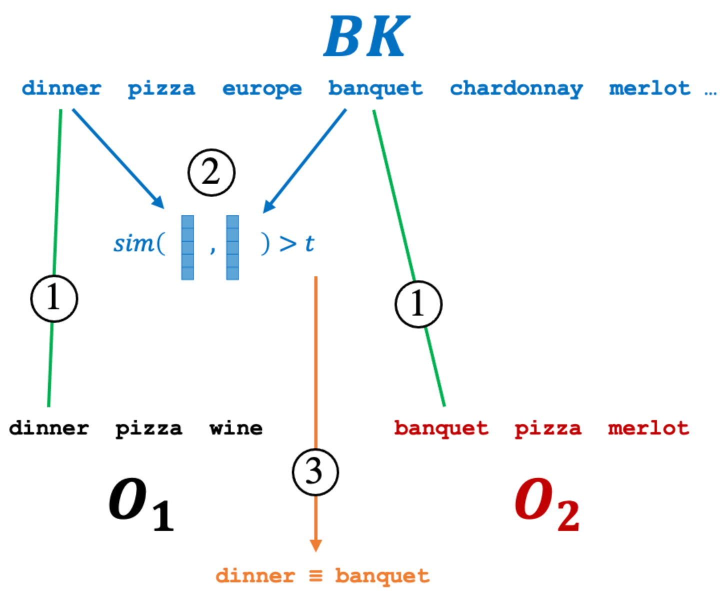


Figure 14.2: Matching Through External Embedding

can be applied as depicted in Figure 14.2 (the process numbers match up with the numbers annotated in the figure):

1. *Concept Linking* In the first step, the concepts in the ontologies need to be linked to concepts in the background knowledge source. This process has been covered in depth in Section 3.6 of Chapter 3.

2. *Similarity Calculation* Once corresponding concepts are identified in the background knowledge source, the similarity is calculated, e.g., by using the Euclidean or cosine distance.

3. *Correspondence Decision* Depending on the similarity score obtained in (2) and a pre-defined criterion, it is decided whether to add a correspondence or not.

In this dissertation, multiple such approaches are presented:

- *ALOD2vec Matcher* is presented in Chapter 17.

- In Chapter 18, multiple external background knowledge sources are compared using the strategy outlined in this section.

It is important to note that *matching through external embedding* can be combined with additional exploitation strategies for BK, such as logic-based techniques (see Section 3.5 and Figure 3.11 of Chapter 3).

14.2 Ontology Matching Through Absolute Orientation of Embedding Spaces

In the previous section, the two principles of how to match with RDF2vec were presented, namely (1) matching through internal embedding and (2) matching through external embedding. The latter process is described in more depth in the next part of this dissertation. In this section, we explore a novel structure-based mapping approach, which is based on knowledge graph embeddings: The ontologies to be matched are embedded, and a technique known as absolute orientation is used to align the two embedding spaces. Next to the approach, the section presents a first preliminary evaluation using synthetic and real-world datasets. We find in experiments with synthetic data that the approach works very well on similarly structured graphs; it handles alignment noise better than size and structural differences in the ontologies.

The work presented in this section has been published before as: Portisch, Jan; Costa, Guilherme; Stefani, Karolin; Kreplin, Katharina; Hladik, Michael; Paulheim, Heiko. Ontology Matching Through Absolute Orientation of Embedding Spaces. In: The Semantic Web: ESWC 2022 Satellite Events. 2022. [to appear] [397]

14.2.1 Introduction

In this section, we examine the use of embedding two ontologies to find an alignment between them. Given two embeddings of the ontologies, we use a set of anchor points to derive a joint embedding space via a rotation operation.

Related Work

Graph Embeddings Given be a (knowledge) graph $G = (V, E)$ where V is the set of vertices and E is the set of directed edges. Further given be a set of relations R, $E \subseteq VxRxV$. A knowledge graph embedding is a projection $E \cup R \rightarrow \mathbb{R}^d$.[3]

[3] Variations of this formulations are possible, e.g., including different dimensions for the vector spaces of E and R, and/or using complex instead of real numbers. For a broader introduction, we refer the reader to Section 2.5 of Chapter 2 and to Chapter 7.

In this section, we use the RDF2vec approach, which generates multiple random walks per vertex $v \in V$. An RDF2vec sentence resembles a walk through the graph starting at a specified vertex v. Those random walks are fed into a *word2vec* algorithm, which treats the entities and relations as words and the random walks as sentences and consequently outputs numeric vectors for entities and relations.

Absolute Orientation Multiple approaches exist for aligning embeddings. In this section, the extension by Dev et al. [101] of the *absolute orientation* approach is used. The approach showed good performance on multilingual word embeddings. The calculation of the rotation matrix is based on two vector sets $A = \{a_1, a_2, ...a_n\}$ and $B = \{b_1, b_2, ...b_n\}$ of the same size n where $a_i, b_i \in \mathbb{R}^d$. In a first step, the means $\bar{a} = \frac{1}{n}\sum_{i=1}^{n} a_i$ and $\bar{b} = \frac{1}{n}\sum_{i=1}^{n} b_i$ are calculated. Now, $\bar{a}$ and $\bar{b}$ can be used to center A and B: $\hat{A} \leftarrow (A, \bar{a})$ and $\hat{B} \leftarrow (B, \bar{b})$. Given the sum of the outer products $H = \sum_{i=1}^{n} \hat{b}_i \hat{a}_i^T$, the singular value decomposition of H can be calculated: $svd(H) = [U, S, V^T]$. The rotation is $R = UV^T$. Lastly, $\hat{B}$ can be rotated as follows: $\tilde{B} = \hat{B}R$.

Matching with Embeddings Embedding-based ontology matching approaches have gained traction recently, mostly using embeddings of the textual information contained in ontologies [408]. OntoConnect [72], for example, uses fastText within a larger neural network to match ontologies; DOME [201] exploits doc2vec; TOM [281] and F-TOM [278] use *Sentence-BERT* (SBERT). With the exception of ALOD2vec Matcher [411], knowledge graph embeddings are rarely used. The work presented in this section is different in that it does not rely on labels or an external knowledge graph. Instead, an embedding is learned for the ontologies to be matched.

14.2.2 Approach

We first train two separate embedding spaces for the two ontologies to be matched (i.e., O_1 and O_2). This is done in two independent RDF2vec training processes. In a second step, we then perform the absolute orientation operation to rotate one embedding space onto the other.

For the matching operation, we assign for each node in $e \in O_1$ the closest node $e \in O_2$ according to Euclidean distance.

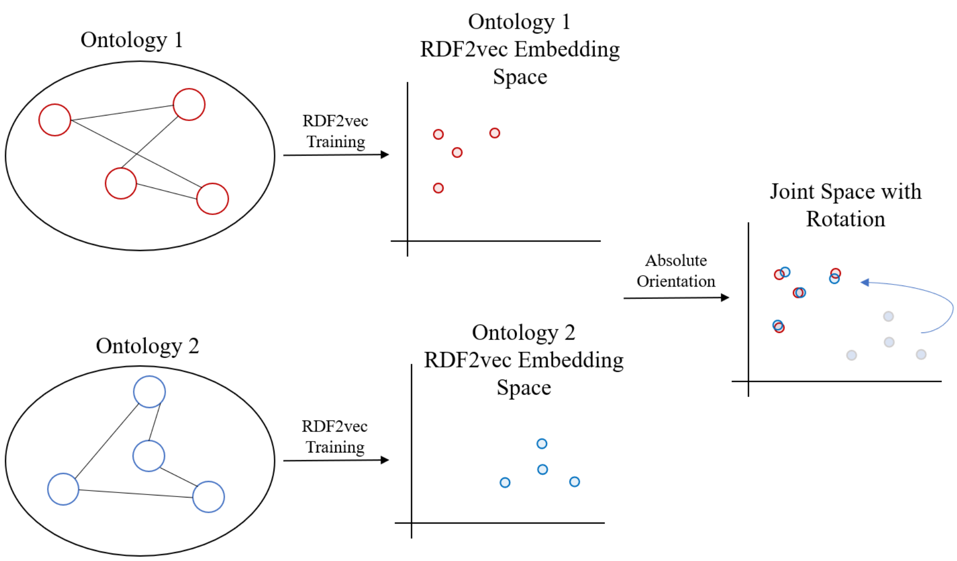


Figure 14.3: High-Level Overview of the Absolute Orientation Approach

14.2.3 Experiments

For the experiments, jRDF2vec[4] [405] was used to obtain RDF2vec embeddings. We chose the following hyper parameter values: $dimension = 100$, $window = 6$, $depth = 6$, $walks = 150$. The code together with the complete set of figures and results is available online.[5]

Synthetic Experiments

In a first step, we perform sandbox experiments on synthetic data. We generate a graph G with 2,500 nodes V. For each node $v \in V$, we draw a random d number using a Poisson distribution $f(k;\lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$ with $\lambda = 4$. We then randomly draw d nodes from $V \setminus v$ and add the edge between v and the drawn node to G. We duplicate G as G' and generate an alignment A where each $v \in V$ is mapped to its copy $v' \in V'$. We define the matching task such that G and G' shall be matched. The rotation is performed with a fraction α from A, referred to as the anchor alignment A'. In all experiments, we vary α between 0.2 and 0.8 in steps of size 0.2.

[4] see `https://github.com/dwslab/jRDF2Vec`

[5] see `https://github.com/guilhermesfc/ontology-matching-absolute-orientation`

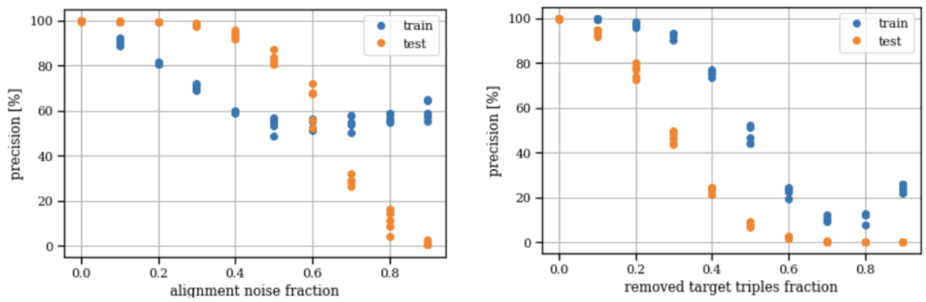


Figure 14.4: The effect of distortions. (1) alignment noise (left) and (2) size differences (right). Graphs are given for $\alpha = 0.2$.

Training Size In order to test the stability of the performed rotation, also referred to herein as training, we evaluate varying values for α. Each experiment is repeated five times to account for statistical variance. The matching precision is computed for each experiment on the training dataset A' and on the testing dataset $A \setminus A'$. The split between the training and the testing datasets is determined by α. We found that the model is able to map the entire graphs regardless of the size of the training set A' (each run achieved a precision of 100%).

Alignment Noise In order to test the stability in terms of noise in the anchor alignment A', we distort a share of the training correspondences by randomly matching other than the correct nodes. We vary this level of alignment noise between 0 (no noise introduced) and 0.9 (90% of the alignments are randomly matched) in steps of size 0.1. Figure 14.4 (left) shows the performance with $\alpha = 0.2$. We observe that the test performance declines with an increasing amount of noise. Interestingly, this relation is not linear. It is visible in Figure 14.4 (left) that the approach can handle 40% of noise before dropping significantly in terms of test performance.

Graph Heterogeneity In order to test the stability in terms of graph heterogeneity, we randomly remove triples from the target graph G' after setting up the alignment between the source graph G and the target graph G'. We vary the fraction of randomly removed triples in G' between 0 (no triples removed) and 0.9 (90% of the triples removed) in steps of size 0.1. In Figure 14.4 (right), it can be observed that with a size deviation of 30%, the performance starts to drop rapidly. Comparing the two plots in the figure, it can be seen that the approach handles noise significantly better than size and structure deviations in graphs.

Experiments on Real Data

We also test our approach on the OAEI multifarm dataset. Here, multilingual ontologies from the conference domain have to be matched. Since the absolute orientation approach does not use textual data, we only evaluate the German-English test case. This is sufficient because the other language combinations of the multifarm dataset use structurally identical graphs. With a sampling rate of 20%, our approach achieves micro scores of $P = 0.376$, $R = 0.347$, and $F_1 = 0.361$. Compared to the systems participating in the 2021 campaign [419], the recall is on par with state of the art systems; an overall lower F_1 is caused by a comparatively low precision score. While not outperforming top-notch OAEI systems in terms of F_1, the performance indicates that the approach is able to perform ontology matching and may particularly benefit from the addition of non-structure-based features.

14.2.4 Conclusion

In this subsection, we presented work on aligning graphs through a graph embedding algorithm combined with an absolute orientation rotation approach. In multiple experiments, we showed that the approach works for structurally similar ontologies. It handles alignment noise better than varying sizes and structures of graphs.

Part IV

Background Knowledge in Knowledge Graph Matching

In Chapter 3, four different exploitation strategies have been identified: (i) factual queries, (ii) structure-based approaches, (iii) statistical/neural methods, and (iv) logic-based techniques. These approaches are presented in detail in Section 3.7 and visualized in Figure 3.10.

This part focuses on the usage of general-purpose background knowledge in ontology and knowledge graph matching. More precisely, multiple exploitation approaches are presented and evaluated. Therefore, this part distinguishes explicit and latent exploitation approaches.

In Chapter 15, *Wiktionary Matcher* is presented. The matcher uses an explicit (factual query) strategy on a general-purpose multilingual dataset.

Subsequently, three concrete latent matching approaches are presented: First, in Chapter 16, a general framework for transformer-based matching is presented on which two OAEI matchers were based upon: TOM [281] and Fine-TOM [278]. Multiple experiments on cross encoders are presented in Section 16.1. Second, Section 16.2 covers *KERMIT*, a scalable knowledge graph matching system which is based on the combination of bi- and cross-encoders. KERMIT builds on the components presented in Section 16.1 and additionally uses (among multiple other components) a bi-encoder together with a logic-based post-processing step. Hence, KERMIT combines a neural approach with a logic-based approach. Third, in Chapter 17, *ALOD2vec Matcher* is presented. The matcher uses a latent (statistical/neural) strategy on a single Semantic Web dataset.

After the presentation of multiple concrete matching systems and approaches, Chapter 18 systematically compares multiple exploitation strategies on various background knowledge sources. The significance of the background knowledge source and the exploitation strategy is determined and compared.

This part closes with two industry use cases of the methods and findings developed in this dissertation in Chapter 19. The first use case (Section 19.1) focuses on matching financial industry terms to a set of predefined classes – a task relevant to the automatic parsing and categorization of financial instruments in text documents. The second use case (Section 19.2) presents a concrete schema matching prototype developed at SAP SE.

Chapter 15

Wiktionary Matcher

In this chapter, a general-purpose background knowledge matching system is presented: *Wiktionary Matcher*. The matching system uses an RDF version of Wiktionary, a multilingual, community-built dictionary.

In terms of the background knowledge classification system presented in Section 3.5, the knowledge source qualifies as *general-purpose* → *structured* → *lexical and taxonomical* → *multilingual* knowledge source[1]. The system uses *direct linking* according to the classification presented in Section 3.6 together with a *factual query* strategy (see Section 3.7).

Wiktionary Matcher is a multilingual matching system. Unlike most other multilingual systems, the matcher does not use a translation API. Instead, the translation component exploits more complex graph-like patterns in Wiktionary that may be considered a very simple case of structure-based matching[2].

The Wiktionary Matcher system participated in three consecutive OAEI campaigns: 2019 [402], 2020 [402], and 2021 [402]. It was updated and improved for each campaign. Since Wiktionary is growing in a continuous fashion (unlike WordNet), the latest Wiktionary version was used in the corresponding campaigns.

[1] Since an RDF version of the dataset is used, one could also argue for a classification as Semantic Web dataset. We decided to classify the system in Chapter 3 as lexical resource since the original dataset is not built on Semantic Web technologies and the dataset strongly exhibits the properties of a lexical resource.

[2] In the survey of Chapter 3, we classified the system as pure *factual query* system since the graph patterns are statically implemented through multiple SPARQL queries.

Parts of this chapter have been published before as:

Portisch, Jan; Hladik, Michael; Paulheim, Heiko. Wiktionary Matcher. CEUR Workshop Proceedings OM 2019 - Proceedings of the 14th International Workshop on Ontology Matching co-located with the 18th International Semantic Web Conference (ISWC 2019), OM@ISWC 2019. Auckland, New Zealand. 2019. [402]

Portisch, Jan; Paulheim, Heiko. Wiktionary Matcher Results for OAEI 2020. In: The Fifteenth International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020), OM@-ISWC 2020. Virtual Space. 2020. [410]

Portisch, Jan; Paulheim, Heiko. Wiktionary Matcher Results for OAEI 2021. In: Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), OM@ISWC 2021. Virtual Space. 2022. [413]

15.1 Presentation of the System

15.1.1 State, Purpose, General Statement

The *Wiktionary Matcher* is an element-level, label-based matcher which uses an online lexical resource, namely *Wiktionary*. The latter is "[a] collaborative project run by the Wikimedia Foundation to produce a free and complete dictionary in every language"[3]. The dictionary is organized similarly to Wikipedia: Everybody can contribute to the project, and the content is reviewed in a community process. Compared to WordNet [149], Wiktionary is significantly larger and also available in other languages than English. This matcher uses *DBnary* [470], an RDF version of Wiktionary that is publicly available[4]. The DBnary dataset makes use of an extended LEMON model to describe the data. For this matcher, recent DBnary datasets for eight Wiktionary languages[5] have been downloaded and merged into one RDF graph. Triples not required for the matching algorithm, such as glosses, were removed in order to increase the performance of the

[3] see `https://web.archive.org/web/20190806080601/https://en.wiktionary.org/wiki/Wiktionary`

[4] see `http://kaiko.getalp.org/about-dbnary/download/`

[5] Namely: Dutch, English, French, Italian, German, Portuguese, Russian, and Spanish.

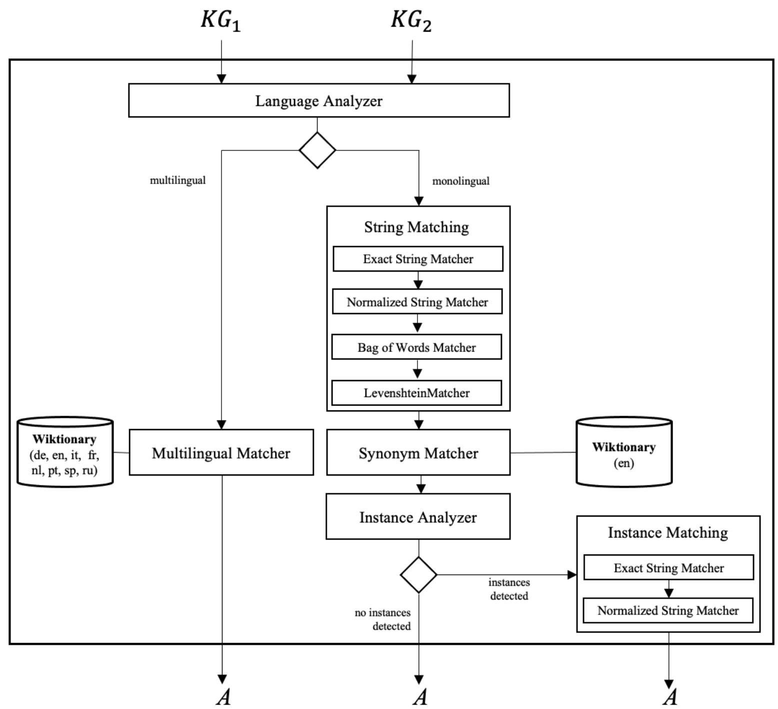


Figure 15.1: High-level overview of the *Wiktionary Matcher*. KG_1 and KG_2 represent the input ontologies and optionally instances. The final alignment is referred to as A.

matcher and to lower its memory requirements. As Wiktionary contains translations, this matcher can work on monolingual and multilingual matching tasks.

In this chapter, the latest version of the system is presented (OAEI 2021 [413]); *Wiktionary Matcher* also participated in the OAEI in 2019 [402] and in the OAEI 2020 [410]. The matcher has been implemented and packaged using the *Matching EvaLuation Toolkit* (MELT)[6], a Java framework for matcher development, tuning, evaluation, and packaging [203, 400].

15.1.2 Specific Techniques Used

This matching system was initially introduced at the OAEI 2019 [402]. An overview of the matching system is provided in Figure 15.1. The main techniques used for matching are summarized below.

[6] see Part II and `https://github.com/dwslab/melt`

Monolingual Matching For monolingual ontologies, the matching system first applies multiple string matching techniques. Afterward, the synonym matcher module links labels to concepts in Wiktionary and checks then whether the concepts are synonymous in the external dataset. This approach is conceptually similar to an upper ontology matching approach. Concerning the usage of a collaboratively built knowledge source, the approach is similar to *WikiMatch* [196], which exploits the Wikipedia search engine. *Wiktionary Matcher* adds a correspondence to the final alignment purely based on the synonymy relation independently of the actual word sense. This is done in order to avoid word sense disambiguation on the ontology side but also on the Wiktionary side: Versions for some countries do not annotate synonyms and translations for senses but rather on the level of the lemma. Hence, many synonyms are given independently of the word sense. In such cases, word sense disambiguation would have to be also performed on Wiktionary [343]. Linking labels of entities to Wiktionary is carried out as follows: The entire label is looked up in the knowledge source. If the label cannot be found, labels consisting of multiple word tokens are truncated from the right, and the process is repeated to check for sub-concepts. This allows for detecting long sub-concepts even if the full string cannot be found. Label *conference banquet* of concept *http://ekaw#Conference-_Banquet* from the *Conference* track, for example, cannot be linked to the background dataset using the full label. However, by applying right-to-left truncation, the label can be linked to two concepts, namely *conference* and *banquet*, and in the following also be matched to the correct concept *http://edas#ConferenceDinner* which is linked in the same fashion. For multi-linked concepts (such as *conference dinner*), a match is only annotated if every linked component of the label is synonymous with a component in the other label. Therefore, to state an example, *lens (http://mouse.owl#MA_0000275)* is not mapped to *crystalline_lens (http://human.owl#NCI_C12743)* due to a missing synonymous partner for *crystalline* whereas *urinary bladder neck (http://mouse.owl#MA_000 2491)* is matched to *bladder neck (http://human.owl#NCI_C12336)* because *urinary bladder* is synonymous to *bladder*.

Multilingual Matching For every matching task, the system first determines the language distributions in the ontologies. If the ontologies appear to be in different languages, the system automatically enables the multilingual matching module: Here, Wiktionary translations are exploited: A match is created if one label can be translated to the other one according to at least one Wiktionary language version – such as the Spanish label *ciudad* and the French label *ville* (both meaning *city*). This process is depicted in Figure 15.2: The Spanish label is

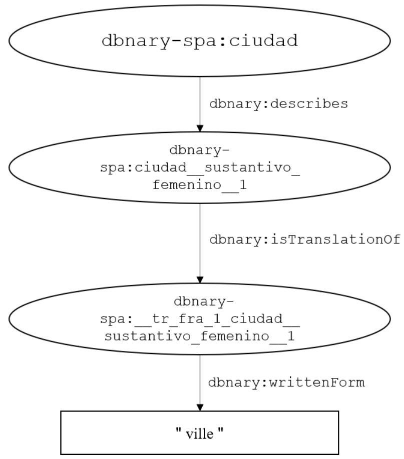


Figure 15.2: Translation via the Wiktionary headword (using the DBnary RDF graph). Here: One (of more) French translations for the Spanish word *ciudad* in the Spanish Wiktionary.

linked to the entry in the Spanish Wiktionary, and from the entry, the translation is derived. If there is no Wiktionary version for the languages to be matched or the approach described above yields very few results, it is checked whether the two labels appear as a translation for the same word. The Chinese label 决定 (juédìng), for instance, is matched to the Arabic label قرار (qrār) because both appear as a translation of the English word *decision* on Wiktionary. This (less precise) approach is particularly important for language pairs for which no Wiktionary dataset is available to the matcher (such as Chinese and Arabic). The process is depicted in Figure 15.3: The Arabic and Chinese labels cannot be linked to Wiktionary entries but, instead, appear as translations for the same concept.

Instance Matching The matcher presented in this chapter can also be used for combined schema and instance matching tasks. If instances are available in the given datasets, the matcher applies a two-step strategy: After aligning the schemas, instances are matched using a string index. As there are typically many instances, Wiktionary is not used for the instance matching task in order to increase the matching runtime performance. Moreover, the coverage of schema-level concepts in Wiktionary is much higher than of instance-level concepts: For example, there is a sophisticated representation of the concept *movie*[7], but there

[7] see `https://en.wiktionary.org/wiki/movie`

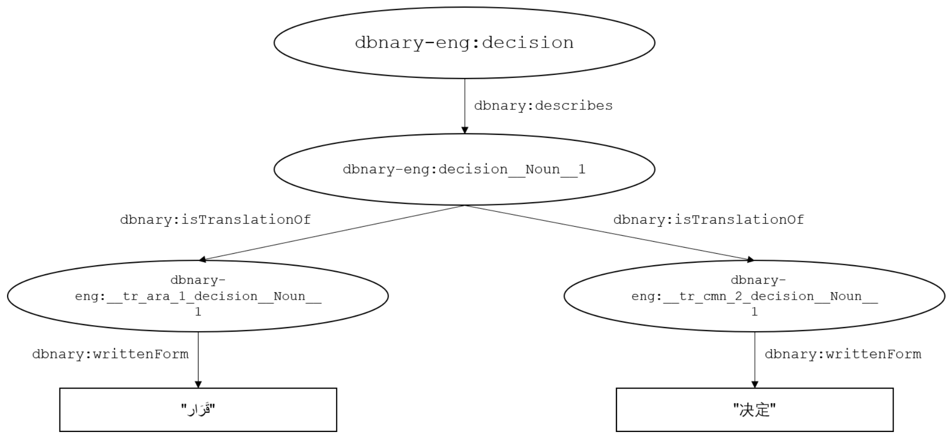


Figure 15.3: Translation via the written forms of Wiktionary entries (using the DBnary RDF graph). Here: An Arabic and a Chinese label appear as translations for the same Wiktionary entry (*decision* in the English Wiktionary).

are hardly any individual movies in Wiktionary. For correspondences where the instances belong to classes that were matched before, a higher confidence is assigned. If one instance matches multiple other instances, the correspondence is preferred where both their classes were matched before.

Explainability Unlike many other ontology matchers, this matcher uses the extension capabilities of the alignment format [96] in order to provide a human-readable explanation of why a correspondence was added to the final alignment. Such explanations can help to interpret and to trust a matching system's decision. Similarly, explanations also allow to comprehend why a correspondence was falsely added to the final alignment: The explanation for the false positive match *(http://confOf#Contribution, http://iasted#Tax)*, for instance, is given as follows: *"The first concept was mapped to dictionary entry [contribution] and the second concept was mapped to dictionary entry [tax]. According to Wiktionary, those two concepts are synonymous."* Here, it can be seen that the matcher was successful in linking the labels to Wiktionary but failed due to the missing word sense disambiguation. In order to explain a correspondence, the `description` property[8] of the *Dublin Core Metadata Initiative* is used.

[8] see `http://purl.org/dc/terms/description`

15.1.3 Extensions to the Matching System for the 2021 Campaign

For the 2021 campaign, the background knowledge has been updated: The system uses DBnary dumps as of late July 2021. The Wiktionary knowledge source grew significantly compared to the version used in the OAEI 2020: In total, 7 million new triples were added, an increase of roughly 9%.

Besides upgrading the background knowledge, the underlying architecture was also improved: Rather than using a custom Wiktionary component, the 2021 version of the matching system was adapted to use the background knowledge modules that were made available with the release of MELT 3.0 [406]. With these changes, the code base is cleaner and better modularized. Improvements to the Wiktionary module will now benefit all MELT users. It is important to emphasize that these architectural improvements do not change the matching algorithm compared to the 2020 version. The system was, furthermore, adapted to be packaged as MELT Web Docker[9] container. The implementation is publicly available on GitHub.[10]

15.2 Results

15.2.1 Anatomy Track

On the anatomy track, recall and F_1 could be slightly improved compared to the 2020 version of the matcher. The system performs at the median of all 2021 systems with an F_1 score of 0.843 (precision = 0.956, recall = 0.753).

15.2.2 Conference Track

The matching system achieves almost the same results as in 2021 on the conference track, with a slightly improved recall. With an F_1 score of 0.59 on `rar2-M3`, the system performs above the median in terms of F_1.

15.2.3 Multifarm Track

The largest overall improvements compared to last year could be observed on the Multifarm track: Here, the F_1 score could be improved through a higher recall (the precision fell slightly). Like in the 2020 campaign, *Wiktionary Matcher* was the system with the overall highest precision and scored third place behind AML and LogMap.

[9] see `https://dwslab.github.io/melt/matcher-packaging/web`

[10] see `https://github.com/janothan/WiktionaryMatcher`

15.2.4 LargeBio Track

On the LargeBio track, *Wiktionary Matcher* was merely run on the FMA-NCI small fragments matching task.[11] The system performed identically in terms of F_1 compared to the previous year. With an F-Measure of 0.913, *Wiktionary Matcher* is the third-best system on the task after AML and LogMap. This is remarkable given that the system does not use any domain-specific resources for matching.

15.2.5 Knowledge Graph Track

As in 2020, *Wiktionary Matcher* is the best matching system on the knowledge graph track.[12] The performance numbers did not change compared to the 2020 version of the matcher.

15.2.6 Common Knowledge Graph Track

In 2021, a new track was added to the OAEI: The *Common Knowledge Graph Track* [136]. Although not optimized for this track, *Wiktionary Matcher* achieved the second-best result in terms of F_1 with a score of 0.89.

15.3 General Comments

It is important to note that the matching system currently exploits only a small share of semantic relations available on Wiktionary. The system is restricted by the available relations extracted by the DBnary project. The additional exploitation of the relations *alternative forms* or *derived terms*, for instance, would likely improve the system. However, those are not yet extracted and are consequently not used for the matching task as of today. The improvements observed on Anatomy and Conference are completely due to the updated Wiktionary version since the core matching code was left unchanged.

[11] This is likely due to the 8 hours timeout. Compared to 2020, the system seems to have been slower and was, therefore, not evaluated on all tasks. The most likely reason is a slower runtime due to the docker container format used this year.

[12] *2021* [403] achieves the same F_1 score – however, as the performance of the latter matcher on classes and properties is slightly worse, *Wiktionary Matcher* comes in first.

15.4 Conclusion

In this chapter, we presented the *Wiktionary Matcher*, a matcher utilizing a collaboratively built lexical resource, as well as the results of the system in the 2021 OAEI campaign. Given Wiktionary's continuous growth, it can be expected that the matching results will continue to improve over time – for example, when additional synonyms and translations are added. In addition, improvements to the DBnary dataset, such as the addition of alternative word forms, may also improve the overall matcher performance in the future.

Chapter 16

Matching with Transformers

Multiple techniques exist to perform the matching operation algorithmically [129] (see also Section 2.6.5 of Chapter 2). One of the strongest signals for automated matching of ontologies and knowledge graphs is the textual description of the concepts. The methods that are typically applied (such as character- or token-based comparisons) are relatively simple and therefore do not capture the actual meaning of the texts. With the rise of transformer-based language models, text comparison based on meaning (rather than lexical features) is possible.

This chapter presents work on language transformers for ontology and knowledge graph matching.

In Section 16.1, we model the ontology matching task as a text classification problem and present approaches based on transformer models. An easy-to-use implementation in the MELT framework is presented and first evaluations are carried out.

In Section 16.2, a knowledge graph matching system based on the combination of bi- and cross-encoders is presented. The system, called *KERMIT*, is evaluated in multiple configurations.

16.1 Matching with Transformers in MELT

In this section, we model the ontology matching task as a classification problem and present approaches based on transformer models. We further provide an easy-to-use implementation in the MELT framework, which is suited for ontology and knowledge graph matching. We show that a transformer-based filter helps to choose the correct correspondences given a high-recall alignment and already achieves a good result with simple alignment post-processing methods.

The work presented in this section has been published in before as:

Hertling, Sven♠; Portisch, Jan♠; Paulheim, Heiko. Matching with Transformers in MELT. In: Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), OM@ISWC 2021. 2022. [205]

16.1.1 Introduction

Multiple techniques exist to perform the matching operation in an automated manner [129]. Labels and descriptions are among the strongest signals concerning the semantics of an element of a knowledge graph. Here, matcher developers often borrow strategies from the *natural language processing* (NLP) community to determine the similarity between two strings. Since the attention mechanism [544] has been presented, so-called transformer models have gained a lot of traction in the NLP area, and transformer models have achieved remarkable results on tasks such as machine translation [544] or question answering [102, 575].

In this section, we bring transformers to the ontology matching task. Our contributions are twofold: Firstly, we present a transformer extension to the *Matching EvaLuation Toolkit* (MELT), which allows users to easily exploit state-of-the-art pre-trained transformer models like BERT[1] [102] or RoBERTa [324] in their matching pipelines. Secondly, we evaluate different transformer-based matching approaches, and we discuss the strengths and weaknesses of transformer models in the matching domain.

16.1.2 Related Work

Transformers are deep learning architectures that combine stacked encoder layers with a self-attention [544] mechanism. These architectures are typically applied in unsupervised pre-training scenarios with massive amounts of data. Since transformers achieved very good results in the NLP domain, they are also used in other disciplines. Brunner and Stockinger [54], for instance, apply transformers for the task of entity matching and show that they achieve better results than classical deep learning models. Peeters et al. [387] report promising results on the similar task of product record matching. In a similar spirit, the DITTO entity matching system consists of a complete architecture (including blocking and data augmentation for fine-tuning) for entity matching that is based on

[1] *BERT* stands for “Bidirectional Encoder Representations from Transformers”.

transformer models [309]. It is evaluated on the ER-Magellan benchmark and achieves good results.

Applications of transformers for the pure ontology matching task are less frequent compared to the entity matching domain. Wu et al. [579] created the *Deep Attentional Embedded Ontology Matching* (DAEOM) system, which jointly encodes the textual description as well as the network structure. It contains negative sampling approaches as well as automatic adjustments of thresholds.

16.1.3 Matching with Transformers

Since transformer models are language models, it is a hard requirement that the elements in the ontology have labels or descriptions. We propose to model the match operation as an unbalanced binary classification problem where the classifier receives a correspondence and predicts whether this correspondence is correct or not. Eventually, only correct correspondences are kept. The match operation can be (i) *complete* or (ii) *partial*. In a *complete* matching setting, each element $e_{1i} \in O_1$ respectively $e_{2i} \in O_2$ needs a textual representation. The latter can be obtained, for instance, by concatenating the URI fragment and all annotation properties. The transformer model then classifies each element in the Cartesian product of the ontologies to be matched. Since the set of comparisons grows quadratically for the complete matching case, and matching with transformers can be computationally intensive, it is also possible to use a candidate generator which reduces the total number of comparisons. This candidate generator can be regarded as a matching system, which returns an alignment A_C. In the *partial* case, we generate textual representations only for candidates in the alignment ($c \in A_C$) and perform a classification operation only for the correspondences $c \in A_C$. Therefore, the focus of the candidate generator should be recall since the generator determines the theoretically largest attainable recall score of the system, i.e., for the final alignment A, $A \subseteq A_C$ holds. This approach can also be seen as a *matching repair* technique.

16.1.4 MELT Transformer Extension

MELT

MELT[2] [203] (see Part II of this dissertation) is a framework for ontology, instance, and knowledge graph matching. It provides functionality for matcher development, tuning, evaluation, and packaging. It supports both, HOBBIT and

[2] `https://github.com/dwslab/melt/`

Figure 16.1: Recommended Pipeline for the MELT Transformer Filter

SEALS, two heavily used evaluation platforms in the ontology matching community. Since 2021, MELT also supports the new *Web Interface*[3] format which was designed for the OAEI. The core parts of the framework are implemented in Java, but the evaluation and packaging of matchers implemented in other languages are also supported. Via the MELT ML extension [204], ML libraries developed in Python can also be used by Java components. Since 2020, MELT has been the official framework recommendation by the OAEI, and the MELT track repository is used to provide all track data required by SEALS. MELT is also capable of rendering Web dashboards for ontology matching results so that interested parties can analyze and compare matching results on the level of correspondences without any coding efforts [400].

In this work, we extend the ML component of MELT so that transformer operations can be called directly from the Java code. Therefore, we use the *Hugging Face transformers* library [576], which allows the usage and fine-tuning of many transformer models.

Obtaining Textual Descriptions from Resources

In order to serialize textual descriptions, MELT offers various classes extending the `TextExtractor` interface. For any given resource, those return extracted text as a set of strings. They do not normalize the text because this is a post-processing step. They merely select specific literals, URI fragments, etc. In our experiments, we use three of those extractors. They are ordered by the number of strings which are returned (most strings to fewest strings)[4]:

`TextExtractorSet` returns the highest amount of literals because it retrieves all literals where the URI fragment of the property is either a label, name, comment, description, or abstract. This includes also `rdfs:label` and `rdfs:comment`. Furthermore, the properties `prefLabel`, `altLabel`, and `hiddenLabel`

[3] `https://dwslab.github.io/melt/matcher-packaging/web`

[4] A more detailed overview can be found in the user guide:
`https://dwslab.github.io/melt/matcher-development/matching-with-jena#textextractors`

from the SKOS[5] vocabulary are included, as well as the longest literal (based on the lexical representation of it). Additionally, all properties which are defined as `owl:AnnotationProperty` are followed in a recursive manner in case the object is not a label but a resource. In such a case, all annotation properties of this resource are added. The extractor reduces the potentially large set of literals by comparing the normalized texts and only returns the ones which are not identical (note here that the original literals are returned, not the normalized ones).

The `TextExtractorShortAndLongTexts` reduces the set of literals further by checking if a normalized literal is fully contained in another literal. In this case, the literal is not returned. This is only applied within the two groups of long and short texts to extract not only a long abstract but also a short label. Label-like properties are regarded as short texts, while comment/description properties are regarded as long texts.

The `TextExtractorForTransformers` extracts the smallest number of literals (out of the text extractors presented here) by returning exclusively labels that are not contained in other labels (without distinguishing in long and short texts). This results in reducing the set of strings even more because labels which appear in a comment are also not returned.

Transformers in the Matching Pipeline

In order to allow for reusable matching code, MELT allows chaining matchers to build a dedicated matching pipeline for various problems. In such a pipeline, each matcher receives the alignment of the previous component together with the ontologies that are to be matched (and optionally configuration parameters).

MELT differentiates between matchers and filters. A filter is a component that does not add new correspondences to the alignment but instead further processes the given alignment by (1) removing correspondences and/or (2) adding new confidence/feature weights to existing correspondences.

Since the transformer evaluation of the Cartesian product of descriptions is not a scalable option for most test cases, MELT offers the usage of transformers as a filter through class `TransformersFilter`. The training process is implemented using TensorFlow and PyTorch; the user can decide which implementation shall be used. Therefore, we recommend a transformer-based matching pipeline as shown in Figure 16.1: In the first step, we use a matcher that generates a recall-oriented alignment. The transformer filter will then use the correspondences in the latter alignment to calculate the estimated similarity. The

[5] *SKOS* stands for “Simple Knowledge Organization System”; for more information, see `http://www.w3.org/TR/skos-reference`

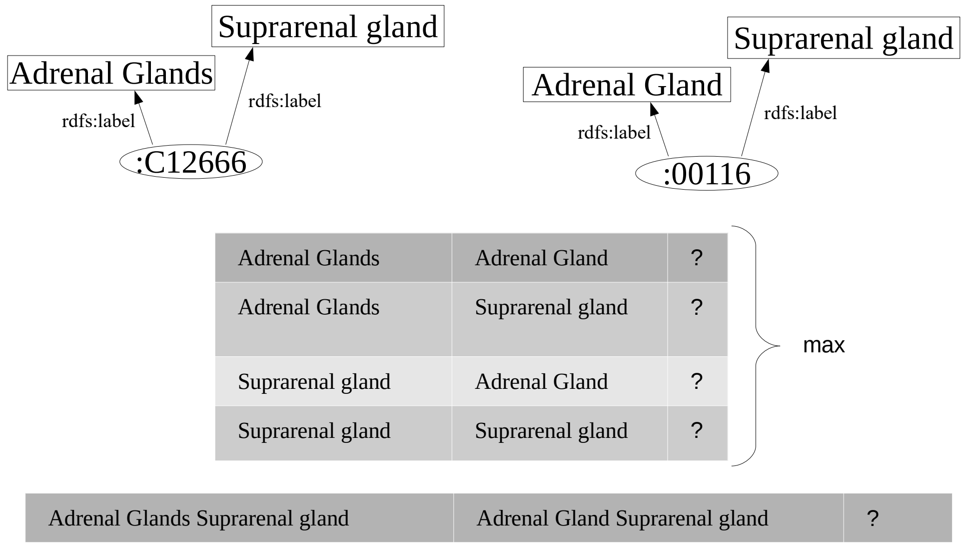


Figure 16.2: Optional multi-text mechanisms implemented in class `TransformersFilter`.

similarity is calculated by first serializing the textual descriptions of each correspondence to a CSV file. Textual descriptions are obtained by a `TextExtractor`. In case there are multiple textual descriptions available, two modes are implemented: (1) A multi-text option (depicted in Figure 16.2), which serializes all combinations of the individual texts; eventually, the maximum similarity will be used. (2) A single-text option which concatenates all textual elements.

After serializing the texts to be compared to a file, the ML Python server is started in the background and predicts the likelihood of a match given the textual description of each correspondence. It is optionally also possible to filter the alignment, for instance, by using a threshold or by reducing the alignment to a one-to-one alignment if applicable.

The MELT extension presented in this section is publicly available in the main branch[6] together with a reference implementation[7] that was used to run the experiments. The new features are documented in the MELT user guide[8].

[6] `https://github.com/dwslab/melt/`

[7] `https://github.com/dwslab/melt/tree/master/examples/transformers`

[8] `https://dwslab.github.io/melt/`

Generating Negatives

In order to run a training process, such as fine-tuning a transformer, data is required for the training step. Positive correspondences can be obtained either from the reference[9] or from a high-precision matching system. However, negative examples are also required. Multiple strategies can be applied here. For example, negatives can be generated randomly using an absolute number of negatives (class `AddNegativesRandomlyAbsolute`) or a relative share of negatives to be generated (class `AddNegativesRandomlyShare`). If the gold standard is not known, it is also possible to exploit the one-to-one assumption and add random correspondences involving elements that already appear in the positive set of correspondences (class `AddNegativesRandomlyOneOneAssumption`). The new extension to the MELT ML module contains multiple out-of-the-box strategies that are already implemented as matching components, which can be used within a matching pipeline. All of them implement the new interface `AddNegatives`. Since multiple flavors can be thought of (e.g., generating type homogeneous or type heterogeneous correspondences), a negatives generator can be easily written from scratch or customized for specific purposes. MELT offers some helper classes to do so, such as `RandomSampleOntModel` which can be used to sample elements from ontologies.

Since the (partial) reference alignments of OAEI tasks are known, and the one-to-one assumption holds, we propose to generate negatives using the same high-recall matcher that is also used in the matching pipeline and to apply the one-to-one sampling strategy: Given the reference and the alignment produced by some high-recall matcher, we determine the wrong correspondences as correspondences where only one element is found in the reference (but not the complete correspondence) and add them to the training set. This is implemented in class `AddNegativesViaMatcher`. Note that for this approach, the reference alignment does not have to be complete. One advantage here is that the characteristics of the training and test set are very similar (such as the share of positives and negatives). This process is visualized in Figure 16.3.

Fine-Tuning Transformers in MELT

A transformer model can be used as-is (particularly if the application is equal to or very similar to its training objective) or be fine-tuned for a specific task. The

[9] Note that convenience methods to do so exist in MELT such as `generateTrackWithSampledReferenceAlignment(Track track, double fraction)` of class `TrackRepository`.

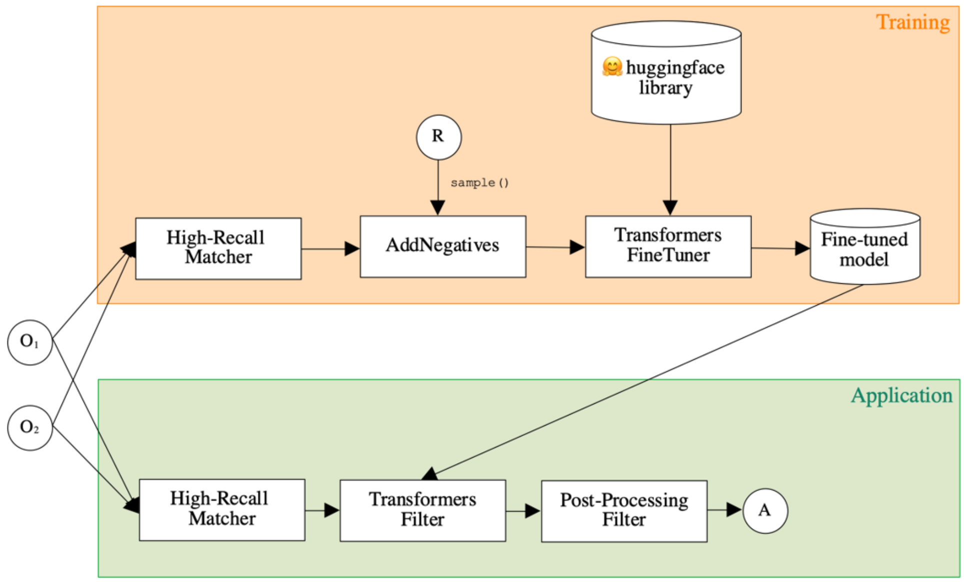


Figure 16.3: Proposed fine-tuning pipeline: The training step is represented by the components in the orange (upper) box, and the application step of the fine-tuned model by the components in the green (lower) box. Note that the *high-recall matcher* is identical in both steps.

default transformer training objectives are not suitable for the task of ontology matching.

Therefore, a pre-trained model needs to be fine-tuned. Once a training alignment is available, class `TransformersFineTuner` can be used to train and persist a model. Like the `TransformersFilter`, the `TransformersFineTuner` is a matching component that can be used in a matching pipeline.[10] Such a training pipeline is visualized in the orange (upper) part of Figure 16.3: A high-recall matcher can be used to generate candidates, and negatives can be generated using a sampled reference (or a reference-like alignment). Repeated calls of the match method will extend the number of training candidates; the actual training is performed when calling method `finetuneModel`. This setup allows for training one model given multiple test cases. The implementation allows, for instance, to train a fine-tuned model per test case, per track, or a global model for multiple tracks. In this section, we fine-tune the model per track to cover their individual characteristics.

[10] Note that this pipeline can only be used for training and model serialization. For the application of the model within a matching pipeline, `TransformersFilter` must be used.

Hyperparameter Optimization

By default, the fine-tuning of the transformer models is executed with the standard training parameters such as a fixed number of epochs (3), a learning rate of $5 \cdot 10^{-5}$, etc. (those default values originate from the transformers library[11]). In hyperparameter optimization, a simple grid search is often applied. But such a tuning method has some disadvantages: (1) each run (parameter combination) needs to be executed until the end to analyze the performance (2) all combinations need to be executed (no information about previous runs is taken into account). Bayesian Optimization [499] solves the latter problem by modeling the performance based on the chosen hyperparameters. Thus, parameter combinations, which do not look too optimistic, are not tried out. Furthermore, runs can be canceled when the optimizing metric does not look promising.

Due to the fact that the training of transformer-based models is rather slow, even more sophisticated methods need to be applied. One of them is *population-based training* (PBT) [229]. Given a population of models, each is trained and evaluated after one epoch. Some models trained with a given parameter combination perform better than others. The better models are duplicated (via checkpointing of model weights) and replace the weaker models to keep the population size fixed. This step is called *exploit* in PBT. Another step, called *explore*, changes the hyperparameters during the training (e.g., the learning rate after the 2^{nd} epoch). With all these mechanisms, it is possible to explore a wide range of parameters in a shorter time frame. PBT is implemented already in *Ray Tune* [311] and uses distributions to describe the search space. Furthermore, it is also used by the transformers library. The initial hyperparameter search space looks as follows:

- `learning rate`: loguniform distribution between 10^{-6} and 10^{-4}
- `epochs`: random choice between 1 and 5
- `seed`: uniform distribution between 1 and 40
- `batch size`: random choice of 4, 8, 16, 32, 64

The search space of the batch size is adjusted by the maximum possible values before the hyperparameter tuning starts. It will determine the maximum batch size by training for one step with the batch size of 4 and checking for out-of-memory errors. If this does not happen, the batch size will be increased in every step by multiplying the value by 2 (such that only powers of 2 are tried out). The final adjusted search space will be all powers of 2 starting from four until the maximum batch size is reached.

[11] `https://huggingface.co/transformers/main_classes/trainer.html#trainingarguments`

The seed is also optimized because different initializations of the classification head of the model can also improve the final metric. The reason behind this is that most models are trained on the masked language modeling task and need a classification layer (usually a linear layer on top of the pooled output) to create the final prediction. This linear layer is initialized with different random weights.

As described above, the hyperparameters can also be changed during training. The following parameters are mutated: `weight decay`: uniform distribution between 0.0 and 0.3; `learning rate`, and `batch size` as defined above.

The metric which is optimized can be chosen from the following KPIs: loss (of the model), accuracy, F_1, recall, precision, or *Area under the ROC Curve* (AUC). The last one is the default because, in a later step in the matching pipeline, the confidence of a correspondence is important for filtering or selection. AUC optimizes this confidence such that all negatives have a low value and all positives a high one. Furthermore, it allows for deciding which model is better, even if they have the same F-measure. The hyperparameter tuning can be easily performed in MELT with class `TransformersFineTunerHpSearch`. It has the same interface as the fine-tuning class, but when calling the `finetuneModel` method, the hyperparameter search is started.

16.1.5 Exemplary Analysis

Experiments

In order to show the effectiveness of transformers for matching in MELT, we performed multiple experiments – each focuses on a different aspect: (1) We evaluate an off-the-shelf transformer model in a zero-shot setting for three OAEI tracks: *Anatomy* [42], *Conference*, and *Knowledge Graph* (KG) [216, 202], (2) we fine-tune well-known models and evaluate them with a sampling rate of 0.2 for the same tracks, (3) for the anatomy track and a fixed model, the sampling rates are modified and the performance is analyzed, (4) for the same track and model we optimize the hyperparameters and analyze their impacts.

We use the following transformer models from the Hugging Face repository: `bert-base-cased` [102], `roberta-base` [324], and `albert-base-v2` [293]. This sample is selected since these models are well known and often used according to the model hub[12] of Hugging Face.

The matching pipeline consists of 4 components: (1) high-recall matcher, (2) transformer filter, (3) confidence threshold cut-off filter, and (4) max weight bipartite partitioning filter.

[12] `https://huggingface.co/models`

The high-recall matcher adds candidates with overlapping tokens, and the transformer filter assigns a confidence to each candidate found in the previous step. An optimal threshold is determined to filter out non-matches. The threshold is calculated not with the complete gold standard but merely with the correspondences that were sampled for the training step. Therefore, the `ConfidenceFinder` class has been extended to work also with incomplete gold standards. Lastly, the max weight bipartite partitioning filter enforces a one-to-one alignment.

Results

In the following, the results of all experiments are presented. The first part covers the zero-shot approach as well as the fine-tuning. Afterward, we report on the impact of different sampling sizes and the results of the hyperparameter search.

Zero-shot and Fine-tuning The results of the zero-shot and fine-tuning experiments are depicted in Table 16.1. The `SimpleString` baseline is a simple matcher which we use as a baseline. The high-recall matcher is the one that is used as a first step in the pipeline in the zero-shot as well as in the fine-tuning setup. This also means that the recall value of this matcher is automatically an upper bound for the recall because the transformer-based filtering will not add any new correspondences. For the zero-shot case where an already fine-tuned model is applied directly (in this case, no reference sampling is necessary), we selected a dataset, which is rather close to our setup. Due to the fact that paraphrasing is very similar to the task of finding the same concepts, the Microsoft Research Paraphrase Corpus [109] is selected. The bert-base-cased model already exists in the Hugging Face hub and is fine-tuned on this dataset. It performs best on the conference track, but these results should be taken with care because of the small number of correspondences and textual descriptions in this track. For the anatomy and knowledge graph track, the fine-tuned models perform much better. For the former dataset, `albert` outperformed `bert` and `roberta` by a large margin. In the KG track, `bert` performed much better. One reason why different models perform better is the different characteristics of the labels and comments.

For Conference and Anatomy, the `TextExtractorSet` is used with the multi-text setup to generate many classification examples, whereas, for the KG track, the `TextExtractorForTransformers` is used to extract fewer literals which are then concatenated together to create only one classification example for each correspondence.

		Conference			Anatomy			Knowledge Graph		
		P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Baseline	**SimpleString**	0.710	0.498	0.586	0.964	0.708	0.816	0.909	0.727	0.808
	High Recall	0.450	0.561	0.179	0.037	0.942	0.071	0.167	0.915	0.283
Zero-Shot	**bert-base-cased (mrpc-tuned)**	0.650	0.548	**0.594**	0.531	0.817	0.644	0.739	0.714	0.726
Fine-Tuned (per Track)	**bert-base-cased**	0.748	0.361	0.487	0.726	0.689	0.707	0.941	0.789	**0.859**
	roberta-base	0.667	0.498	0.570	0.715	0.749	0.732	0.400	0.388	0.393
	albert-base-v2	0.812	0.397	0.533	0.854	0.825	**0.839**	0.687	0.665	0.676

Table 16.1: Results of non-fine-tuned and fine-tuned transformer models (multi-text) with 20% sampling from the reference alignment. As per OAEI customs, we report micro average scores for the conference and macro average scores for the KG track.

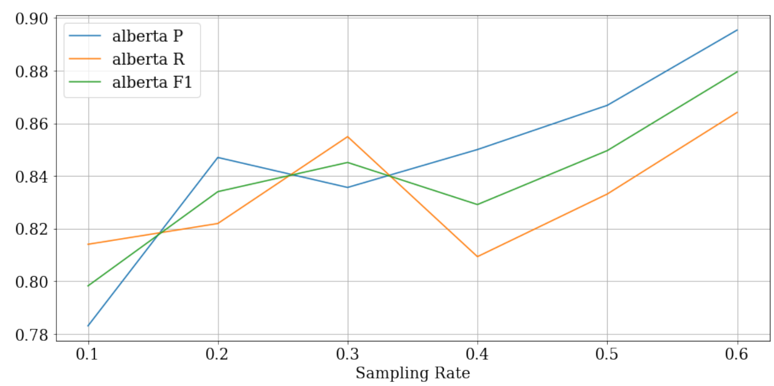


Figure 16.4: `albert-base-v2` performance on the anatomy track using different reference sampling rates.

Sampling Rates We analyzed the performance of the best model on anatomy (albert) using varying sampling rates $s \in [0.1, 0.2, 0.3, 0.4, 0.5, 0.6]$ from the reference. The results are presented in Figure 16.4. Interestingly, fairly good performance can be achieved with very low sampling rates (10% and 20% respectively). Intuitively, the overall performance tends to increase with an increasing share of samples from the reference.

Hyperparameter Tuning The tuning of hyperparameters was executed for the anatomy track and the `albert-base-v2` model. The given search space in Section 16.1.4 is used, and overall, 12 trials are sampled from it, which is also the amount of the model population. The search needs 45 minutes to run in parallel on 4 *graphics processing units* (GPUs) (NVIDIA GeForce GTX 1080 Ti). All other settings are the same as in the normal fine-tuning setup (thus, the numbers are comparable). With PBT, the precision could be improved by 0.02 to 0.874, whereas the recall is only a bit higher (0.832). In terms of F-measure, the hyperparameter tuning additionally gives an improvement of 0.013 (eventually leading to an F_1 of 0.852).

16.1.6 Conclusion

In this section, we introduced a new matching component to the MELT framework, which is based on transformer models. It allows for extracting a textual description of the resource with so-called text extractors and provides an easy option to apply and fine-tune transformer-based models. We propose and evaluate an exemplary matching pipeline for transformer training and application. We hope that our implementation benefits the ontology matching community and enables other researchers to further explore this topic.

In addition, we performed four experiments that demonstrate the capabilities of the newly implemented component. We showed that a transformer-based filter can improve a given alignment by providing a confidence for each correspondence based on its textual description. Moreover, we presented a sophisticated approach for hyperparameter tuning and showed that improvements can be achieved when optimizing the model hyperparameters.

16.2 KERMIT – A Transformer-Based Approach for Knowledge Graph Matching

In the previous section, the core components of a transformer-based matcher were introduced. Since performing pairwise comparisons of all textual descriptions of concepts in two knowledge graphs is expensive and scales quadratically (or even worse if concepts have more than one description), the transformer component was established as a pure filter operation: In Section 16.1, a *high-recall matcher* was used, which represents the upper bound for the attainable recall of the matching system.

In this section, we overcome this problem and present a *pure* transformer-based system. We follow a two-step approach: we first generate matching candidates using a pre-trained sentence transformer (so-called bi-encoder). In a second step, we use fine-tuned transformer cross-encoders to generate the best candidates. We evaluate our approach on multiple datasets and show that it is feasible and produces competitive results.

In addition, the newly presented matching system does not require a reference alignment for fine-tuning the transformer component – instead, it is shown that in many cases, the output of a *high-precision* is sufficient for this training step.

The work presented in this section has been published before as:

Hertling, Sven♠; Portisch, Jan♠; Paulheim, Heiko. KERMIT - A Transformer-Based Approach for Knowledge Graph Matching. Deep Learning meets Ontologies and Natural Language Processing (DeepOntoNLP2022) in conjunction with the ESWC 2022. 2022. [to appear] [206]

16.2.1 Introduction

Traditional transformer models use a cross-encoder which requires that two sentences are used as input to predict the target variable. Since this does not scale when a lot of comparisons have to be performed, blocking methods are typically used to reduce the search space for true positives. However, this approach potentially sacrifices recall since traditional blocking methods rely on basic comparisons such as string overlap. With such blocking methods, a lot of useful correspondences are not found. Sentence-BERT [432] (SBERT) overcomes this disadvantage by providing an approach that allows deriving embeddings for sentences such that two texts are close in this space when they share the same meaning. The idea is to train two transformer models simultaneously with a siamese network architecture. In this section, we evaluate an SBERT-based approach for ontology and KG matching. A challenge of matching ontologies with transformers is the fact that cross-encoders typically have to be fine-tuned; however, this process requires the existence of a (partial) reference alignment, which is not always accessible. Thus, we also use a rather simple matcher, which provides a high precision (with a potentially low recall) to provide positive examples even in the absence of any other training data. In this chapter, we present and evaluate ***KERMIT*** (***K***nowl***E***dge g***R***aph ***M***atch***I***ng with ***T***ransformers), a scalable knowledge graph matching system which is based on SBERT and a fine-tuned transformer component. KERMIT can match knowledge graphs with and without the provisioning of a reference alignment.

16.2.2 Related Work

Transformers are deep learning architectures that combine stacked encoder layers with a self-attention [544] mechanism. These architectures are typically applied in unsupervised pre-training scenarios with massive amounts of data. Applications of transformers for the pure ontology or knowledge graph matching task are less frequent compared to the entity matching domain. Neutel and de Boer [363] use plain *Bidirectional Encoder Representations from Transformers* (BERT) similarity scores. They find that plain fasttext similarities still outperform SBERT descriptions; they further report that SBERT descriptions are the most useful given multiple transformer-based approaches. The work presented in this chapter is similar in that it also uses SBERT. However, we use a more sophisticated pipeline, where SBERT embeddings are complemented with fine-tuned cross-encoders and further alignment repair techniques. This chapter is also more comprehensive in its evaluation comprising of multiple large-scale datasets. The MEDTO system [184] uses a *graph neural network* (GNN) to match data to medical ontologies. Due to the fact that each node in the graph needs a vector representation, they also use transformer-based models to convert the concept names into such a representation. More specifically, they use BioBERT, which is already trained on PubMed abstracts and clinical notes. Unlike the traditional ontology matching system evaluation protocol, they measure their performance in terms of HITS@10/HITS@30 and compare it with state of the art matching systems. In 2021, the MELT framework [203, 204, 400] was extended to also support a transformer filter [205] (see the previous Subsection 16.1 of this chapter). This component is also used by the F-TOM [278] matching system. Similar to F-TOM, the TOM matching system [281] uses a transformer – however, TOM uses a zero-shot SBERT model rather than a fine-tuned cross-encoder. All three publications rely on a traditional blocking approach to reduce the computational complexity (at the expense of reduced recall). Moreover, TOM and the MELT component both require a sample from the reference alignment in order to fine-tune the cross-encoder. In the KERMIT matching system, we overcome those limitations by using a multi-stage matching pipeline.

16.2.3 Approach

Matching Pipeline Overview We propose a multi-step pipeline. The approach is visualized in Figure 16.5. In a first step, the cross encoder needs to be fine-tuned in order to recognize matches (and to discard non-matches). In Figure 16.5, this step is represented in the upper blue box. Once a fine-tuned cross-encoder model is available, it can be applied in the actual matching pipeline.

Training. In a first step, a high-recall alignment is generated (output of the SBERT matcher). For each element $e_1 \in O_1$ the top k closest concepts $E_{2_k} \subset O_2$ with $|E_{2_k}| \leq k$ are added to the high-recall alignment. In this chapter, $k = 5$ is

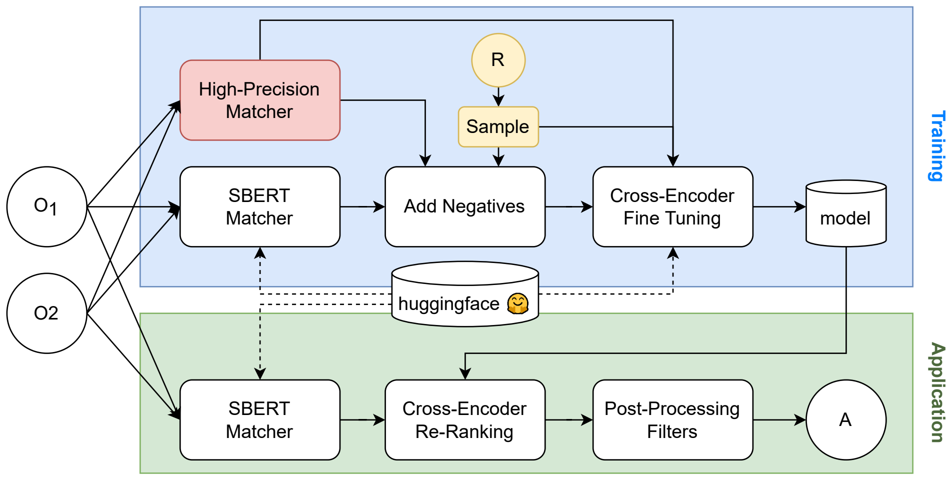


Figure 16.5: Overview of the KERMIT System. The red and yellow parts are alternatives.

used. As we show in the evaluation section, if the correct SBERT model is picked, no fine-tuning is required. In order to train the cross-encoder, positive matches and negative matches are required. For positives, KERMIT offers two options: (1) Exploration of a high-precision matcher (red in Figure 16.5) and (2) Sampling from the reference (yellow in Figure 16.5); both are explained in detail in Subsection 16.2.3.

Application. After a cross-encoder model has been fine-tuned, the actual matching step is carried out. The SBERT matcher generates an initial recall alignment. Afterward, the cross-encoder acts as a re-ranking system and reduces the set generated by the SBERT matcher by picking the best correspondence out of the k correspondences generated per concept. Thus, the resulting complexity is $O(k*(|O_1|+|O_2|))$ in addition to the cost for retrieving the top k results (compared to $O(|O_1|*|O_2|)$ for not using the bi-encoder).

KERMIT uses multiple post-processing filters, which are described in detail in Subsection 16.2.3.

Generating Candidate Correspondences with SBERT

Generating Positives and Negatives. In order to fine-tune the cross-encoder, a set of positive and negative correspondences, more precisely text pairs, is required. KERMIT offers two options to obtain positives: (1) Sampling from the reference and (2) using a high-precision matcher. Option (1) will sample a random share s from the reference alignment. In this chapter, we use a constant share of $s = 20\%$. Option (2) is applicable in situations where a reference alignment is not accessible. Any high-precision matcher can be used whose output will be considered to be correct. In this chapter, we use a string-based matcher, which creates correspondences for classes, properties, and instances (each in isolation). For each resource, all labels and the URI fragment are extracted and normalized (removal of camel case and non-alpha-numeric characters as well as lowercasing) to find matching candidates. Only entities which are mapped to only one other entity are kept. KERMIT assumes that the one-to-one matching assumption holds. The system generates negatives using the same SBERT matcher that is also used in the application pipeline and applies the one-to-one sampling strategy: Given the high-precision alignment and the alignment produced by the SBERT matcher, it determines the wrong correspondences as correspondences where only one element is found in the high-recall alignment (but not the complete correspondence) and adds them to the training set. Note that for this approach, the high-recall alignment does not have to be complete. An example is provided in Figure 16.6: We can directly treat C2 and C5 as positives since they appear in the reference/high-precision alignment. Since we know that C2 must be true and that each concept can only be involved in one correspondence, we regard C1 and C3 as wrong, i.e., add them to the set of negatives. C4 is ignored since we cannot judge whether this correspondence is true or false. One advantage here is that the characteristics of the training and test set are very similar (such as the share of positives and negatives), which is helpful for fine-tuning and using the cross-encoder. This process is visualized in the upper blue box in Figure 16.5.

Obtaining Textual Descriptions. Concepts in ontologies and knowledge graphs may contain more than one textual description. KERMIT extracts all literals where the URI fragment of the property is either *label*, *name*, *comment*, *description*, or *abstract*. This includes `rdfs:label and rdfs:comment`. Furthermore, the properties `prefLabel`, `altLabel`, and `hiddenLabel` from the `SKOS` vocabulary are included. Lastly, all properties which are defined as `owl:Annotation-Property` are followed in a recursive manner in case the object is not a label but a resource. In such a case, all annotation properties of this resource are added. All textual descriptions are collected, normalized, and duplicates are removed. The normalization is only applied to find near duplicates but the actual unmodified text is embedded.

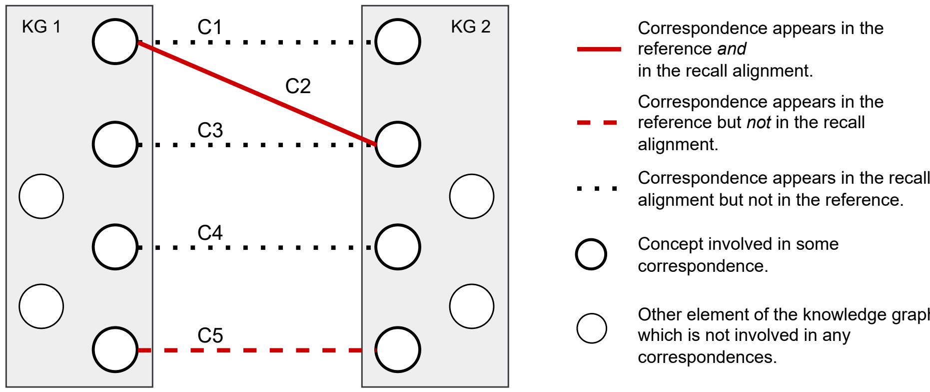


Figure 16.6: Generation of negatives: Given an incomplete reference and applying the one-to-one assumption, we can use C1 and C3 as negatives.

Top K Calculation. Textual descriptions are obtained for each $e_1 \in O_1$ and $e_2 \in O_2$. Classes, properties, and instances are embedded and searched separately such that mixed correspondences like class-instance matches are avoided. Properties are further subdivided into `owl:ObjectProperty`, `owl:DatatypeProperty`, and any other `rdf:Property` to avoid matches which are not compliant with OWL DL. In order to generate a high-recall alignment, all textual descriptions of O_1 and O_2 are embedded using the bi-encoder (SBERT). First, all entities of the source KG are used as query and all entities of the target KG as the corpus. For each textual description of a concept $e_1 \in O_1$, the top k closest descriptions from O_2 are retrieved. They are mapped to their original concepts, which serve as a set of candidates to be matched to e_1. Those correspondences are added to the recall alignment. The confidence is set to the similarity in the embedding space. In case multiple textual representations of two concepts are in the top k descriptions, the closest one is used. The process is repeated in the opposite direction such that each $e_2 \in O_2$ is used as query and all entities from O_1 as corpus. This is necessary because the operation is not symmetric, and target entities may have different top k matches when they are used as a query.

Fine-Tuning of Cross-Encoder

Obtaining Textual Descriptions. In comparison to the bi-encoder, the textual representations of a concept for the cross encoder need to be reduced as they are computed in a cross product. Thus, a slightly different approach to extracting text from resources is applied. Figure 16.7 presents three options:

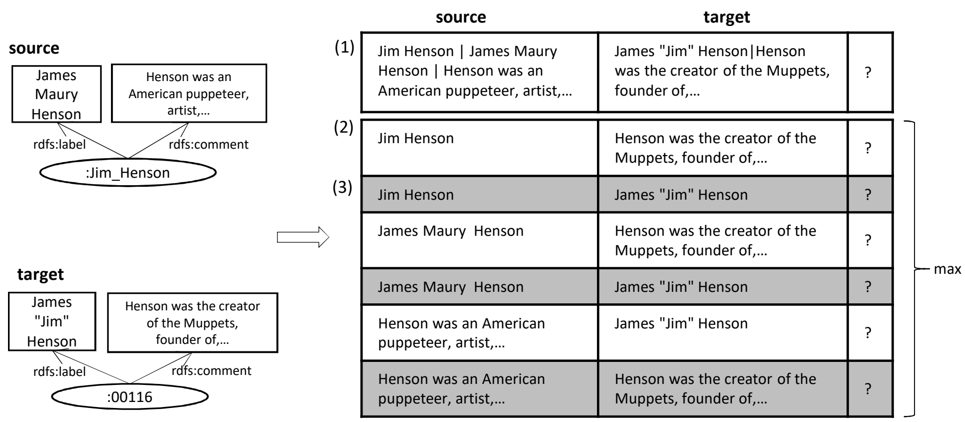


Figure 16.7: Obtaining Textual Descriptions for the Fine-Tuning Step

1. All extracted texts are concatenated together to form one input.

2. All texts are used on their own, and each text is compared with all other extracted texts from the corresponding resource of the target KG. In the example, this corresponds to the whole lower table.

3. The extracted texts are grouped for one resource, and they will be compared only in a cross product with the same group of the other concept. Thus, we still compare individual texts but reduce the overall amount of examples. In Figure 16.7, this corresponds to the rows with a gray background. There are only two groups: (a) short texts like URI fragments, literals of properties where the fragment is `label` or `name`, and literals that are connected with annotation properties. (b) long texts like literals of properties with URI fragment equal to `comment`, `description`, or `abstract` and the longest literal (based on the lexical representation) connected to the resource.

Reducing the number of examples is important because the first component in the matching pipeline (SBERT) will generate lots of correspondences (depending on the k in the top k retrieval step), which need to be analyzed. Furthermore, with the third approach, it can be ensured that short texts are compared only with other short texts. The same logic is also applied to long texts. This reduction of examples further helps the cross-encoder to learn meaningful representations.

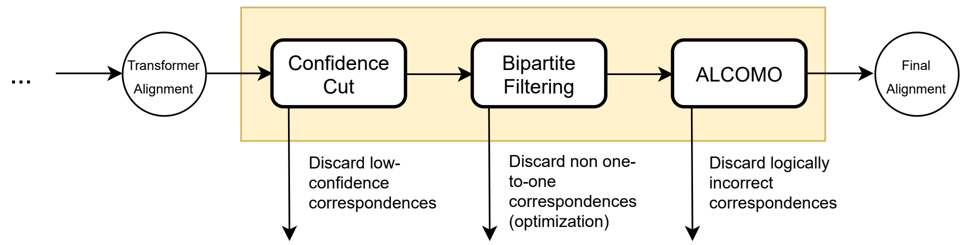


Figure 16.8: KERMIT's Post-Processing Pipeline

Tuning Process. The set of positive and negative text pairs is used to fine-tune the corresponding cross-encoder on a test case basis. The trainer class of the Hugging Face transformers library [576] is used with default settings. If texts are too large to fit into the model, we truncate the longer of the two until both textual representations are short enough. The resulting dataset is highly unbalanced (due to the top 5 retrieval in the first step) and has a lot more negatives than positives (only one correspondence out of five can be correct). During training, examples are randomly assigned to batches which results in batches without positives. Thus, the training batch size is a crucial hyperparameter, and the largest possible value is chosen. It is determined by a dataset-dependent approach which sorts the input texts according to their length and runs one training step to check if everything fits on the GPU. The starting batch size is four and is iteratively multiplied by two until the memory is not sufficient anymore.

Post-Processing Filters.

The cross-encoder reduces the initially obtained recall alignment. However, it still contains at least one correspondence for each concept in the ontologies. The alignment may, in addition, be incoherent since transformers are not aware of the logical constraints found in ontologies. Therefore, KERMIT uses multiple post-processing steps to obtain the final alignment. These steps are implemented as filters, i.e., they reduce the alignment. Hence, they improve the precision of the final alignment. The post-processing pipeline is depicted in Figure 16.8.
(1) Confidence Cut: Ideally, the cross encoder produces meaningful confidence scores $c \in [0,1]$. These scores can be used to automatically remove low-confidence matches. As discussed earlier, the first filter removes all correspondences with a confidence $c < t$. For KERMIT, we use $t = 0.5$, since this complements the softmax activation function used in the last transformer layer.
(2) MWB: The *Max Weight Bipartite Filtering* (MWB) component solves the assignment problem. It further reduces the many-to-many alignment to a one-to-

one alignment. This global optimization usually works better than choosing the best match for each entity in isolation. Due to the high number of correspondences, the Hungarian algorithm cannot be used directly; Cruz et al. [86] provide an efficient alternative by reducing the problem to the maximum weight matching in the bipartite graph. This algorithm is re-implemented in KERMIT to output an optimized one-to-one alignment.
(3) ALCOMO: The *Applying Logical Constraints on Matching Ontologies* [336] (ALCOMO) system is an efficient alignment repair implementation that transforms the potentially incoherent alignment into a coherent one. ALCOMO sorts the matcher alignment according to confidence and then adds the correspondences in sequence to the final alignment. After every addition, it is checked whether the alignment is still coherent. Correspondences are only added if they do not cause the final alignment to be incoherent. This process is algorithmically optimized following a divide and conquer pattern so that larger groups of correspondences are checked and added to the alignment (rather than performing single additions). For this chapter, the algorithm has been re-implemented and integrated into KERMIT based on the original implementation. KERMIT uses the ALCOMO component with the PELLET reasoner and the greedy strategy to obtain a logically coherent alignment.

Implementation and Hardware KERMIT is implemented using Java and Python. The implementation is publicly available as a command line tool.[13] In addition, the best configuration of KERMIT was packaged as a docker container for convenient reuse in other research projects. The evaluation has been performed using the MELT framework. It was performed on a server running Debian with 384 GB of RAM, 40 CPU cores (2.1 GHz), and 4 Nvidia Tesla V100 graphics cards.

16.2.4 Evaluation

KERMIT is evaluated on two different tracks by the *Ontology Alignment Evaluation Initiative* (OAEI): OAEI Anatomy [42] and OAEI LargeBio.

Evaluation of the High Precision Matcher

The results of the high precision matcher on the evaluation data are presented in Table 16.2. For the Anatomy and LargeBio track, the precision is at 99%. Therefore, the cross-encoder needs to tolerate up to 1% of noise in the positives of the training.

[13] `https://github.com/dwslab/melt/tree/master/examples/sentence-transformers`

		Anatomy			LargeBio		
k	**Model**	Prec	Rec	F_1	Prec	Rec	F_1
k=5	all-MiniLM-L6-v2	0.066	0.970	0.124	0.068	**0.954**	0.127
	paraphrase-albert	0.065	0.956	0.122	0.066	0.943	0.124
	paraphrase-TinyBERT	0.066	**0.974**	0.124	0.067	0.948	0.125
k=3	all-MiniLM-L6-v2	0.105	**0.963**	0.189	0.104	**0.943**	0.188
	paraphrase-albert	0.102	0.948	0.185	0.102	0.931	0.185
	paraphrase-TinyBERT	0.105	0.962	0.189	0.104	0.938	0.187
k=1	all-MiniLM-L6-v2	0.307	**0.933**	0.462	0.320	**0.894**	0.471
	paraphrase-albert	0.298	0.912	0.449	0.316	0.886	0.466
	paraphrase-TinyBERT	0.302	0.925	0.455	0.319	0.891	0.470
-	Baseline Matcher	0.964	0.708	0.816	0.460	0.410	0.434
-	High Precision Matcher	0.990	0.617	0.761	0.992	0.444	0.614

Table 16.2: Performance of Zero-Shot bi-Encoders, Baseline, and High Precision Matcher. The best recall per k is highlighted with bold print. For the LargeBio track, macro averages are stated.

Evaluation of SBERT Models

In a first step, we evaluate SBERT models in terms of their ability to generate a high recall. The following models were evaluated:

- `all-MiniLM-L6-v2`
- `paraphrase-albert-small-v2`
- `paraphrase-TinyBERT-L6-v2`
- `paraphrase-mpnet-base-v2`
- `paraphrase-MiniLM-L6-v2`
- `paraphrase-MiniLM-L3-v2`
- `all-mpnet-base-v2`
- `all-distilroberta-v5`

This set of models was generated by choosing (1) the most downloaded sentence similarity models suitable for this task, (2) the top-performing models on six semantic search datasets, and (3) the best performing models on a smaller subset of the data. All of them are publicly available via the Hugging Face model hub. Out of these eight selected models, the first three perform best when applied to all previously discussed datasets of the OAEI. Their evaluation is presented below in more detail.

Results. The results for $k = \{1,3,5\}$ are reported in Table 16.2. As baseline, the `SimpleStringMatcher` of the MELT framework is used. All SBERT models achieve a remarkably high recall: If $k \geq 3$, more than 90% of the correspondences are retrieved independently of the dataset or SBERT model. Interestingly, the drop in recall when reducing k is small. The overall best model independent of k is `all-MiniLM-L6-v2`. The comparison with the baseline shows that each SBERT model outperforms the traditional high-recall matcher in terms of recall – even with $k = 1$. The higher precision of the baseline matcher also shows that the matcher is implemented with string-based methods. Since the performance of the pre-trained models is sufficient for the matching task in terms of recall, we do not fine-tune those. In the following, we continue our experimentation with $k = 5$. It is important to note that KERMIT would still achieve reasonable results with a lower k. KERMIT's runtime performance is linear to k. Hence, k can be scaled down in order to increase the runtime performance.

Significance Testing. Since the performance figures are still relatively close, we performed McNemar's asymptotic significance test for ontology alignments with continuity correction as described in [348]. In cases where the asymptotic test cannot be performed, we calculated the exact numbers as a fallback. With a significance level of $\alpha = 0.05$, we find that the SBERT models do not produce statistically significantly different alignments on the Anatomy track. The only larger statistical variation in alignments occurs on the LargeBio track where roughly half of the alignments are statistically significantly different. Therefore, we conclude that combining SBERT models is not the best option to increase the recall further on most tracks; instead, k should be increased.

Evaluation of KERMIT

The set of experiments is continued with the best-performing SBERT model (`all-MiniLM-L6-v2`). The value of k is set to 5. The idea is that the cross-encoder will assign an even better and more detailed confidence than the bi-encoder because it can analyze both texts (from source and target) at the same time and pay attention to the words which are essential. Similar to selecting SBERT models, the cross encoders are chosen based on the download rate of the Hugging Face model hub and commonly used models. The selected models are: `albert-base-v2`, `bert-base-cased`, and `roberta-base`. The results of the complete matching pipeline are presented in Table 16.3. The two columns *High Prec Matcher* and *20% Reference* refer to the fine-tuning mode of the cross-encoder.

On Anatomy, the results are very competitive with existing OAEI systems. It can be observed that the matching pipeline achieves a very high precision on this task. The performance increase when switching from a zero-shot approach to a reference sampling approach on this track is between 2 and 7 percentage points – depending on the actual cross-encoder used. The models score simi-

			High Prec Matcher			20% Reference		
Track	Test Case	Model	P	R	F_1	P	R	F_1
Anatomy	mouse-human	albert	**0.972**	0.699	0.813	**0.966**	0.795	0.872
		bert	0.963	0.718	0.822	0.962	**0.832**	**0.892**
		roberta	0.968	**0.725**	**0.829**	0.950	0.827	0.884
LargeBio	fma-nci	albert	**0.980**	**0.830**	**0.899**	**0.986**	0.806	0.887
		bert	0.976	0.817	0.889	0.957	0.691	0.803
		roberta	0.978	0.829	0.897	0.984	**0.815**	**0.891**
	snomed-nci	albert	**0.970**	**0.553**	**0.704**	**0.971**	0.621	0.752
		bert	0.962	0.543	0.694	0.953	0.533	0.684
		roberta	0.968	0.552	0.703	0.967	**0.629**	**0.762**
	fma-snomed	albert	0.939	0.222	0.359	**0.978**	0.666	0.792
		bert	**0.940**	0.211	0.345	0.975	0.632	0.767
		roberta	0.927	**0.227**	**0.365**	0.968	**0.716**	**0.823**

Table 16.3: Performance of KERMIT. Best precision, recall, and F-measure per test case are highlighted in bold. The best performance independent of the positive example generation (high precision matcher vs. reference sampling) is additionally underlined.

larly in terms of F_1 when using a precision matcher. Here, `roberta` achieves the overall highest score. When sampling from the reference, the variation in the performance is larger. The `bert` model achieves the highest F_1 of almost 90%. Other systems scoring in this top performance range, such as AML [147], heavily exploit domain-specific background knowledge hand-picked for this matching task.

On LargeBio, the best scores are achieved on the FMA-NCI task, followed by SNOMED-NCI, and FMA-SNOMED. This is generally in line with typical OAEI evaluation systems. An interesting observation on this track is the fact that the unsupervised variants using the high precision matcher outperform the reference sampled versions on some test cases – such as FMA-NCI and SNOMED-NCI. This is most likely due to the fact that the high precision matcher can generate more positives than the static 20% sampling cut. In general, `albert` achieves the highest precisions with both training options. On the one hand, `roberta` is always better performing when using the reference sampling, but on the other hand, `albert` should be used when training examples are generated with a high precision matcher. Again, the results are very competitive. On FMA-SNOMED, for instance, the reference-trained `roberta` configuration performs almost as well as AML, the top-notch 2021 matching system (which makes use of domain-specific background knowledge).

16.2.5 Conclusion

In this section, we presented KERMIT, a knowledge graph matching system that is based on a combination of bi- and cross-encoders, and reuses a logic-based alignment repair step in order to improve precision. We showed that bi-encoders are very suitable for blocking. It can be expected that they will replace traditional string-based blocking approaches in the future since (1) they can be easily configured in terms of how many candidates shall be generated, (2) they produce high-quality results, and (3) they are not biased towards pure string sequence matches.

The good results on domain-specific datasets show that the approach is particularly promising for domains where no specific knowledge sources exist, and traditional matching systems fail due to missing background knowledge.

The fine-tuned cross-encoders further helped to differentiate between true positives and false positives by re-ranking the correspondences accordingly. In comparison to other OAEI systems, KERMIT can already outperform a lot of them. Furthermore, we showed that a simple high-precision matcher can also be used to generate positive correspondences – especially in the case where the label is not the only textual information of a resource.

The bi-encoders already show good performance even when k is reduced to three or one.

Chapter 17

ALOD2vec Matcher

In this chapter, a general-purpose background knowledge matching system is presented: *ALOD2vec Matcher*. The system uses *direct linking* according to system presented in Section 3.6 together with a *neural* strategy (see Section 3.7). Compared to the system presented in Chapter 15, this matcher uses a different dataset and a latent exploitation strategy rather than an explicit one.

ALOD2vec Matcher participated multiple times in different OAEI campaigns: 2018 [488], 2020 [491], and 2021 [492]. It was updated and improved for each campaign. At the heart of the system, the WebIsALOD dataset is used. However, the way *how* the knowledge source is consumed was changed over the years. In 2018, the original source was used, leading to a matching system that is multiple gigabytes large in terms of disk size. Since 2020, the KGvec2go API has been used (see Chapter 8), which shrinks the matching system significantly concerning disk size requirements. With the new knowledge source, vectors are requested on demand.

In terms of the background knowledge classification system presented in Section 3.5, the pure WebIsALOD knowledge source qualifies as *general-purpose* → *strucuted* → *Semantic Web dataset* → *single* knowledge source. The KGvec2-go knowledge source, however, is classified as *general-purpose* → *structured* → *pre-trained neural model* → *monolingual*.

Parts of this chapter have been published before as:

Portisch, Jan; Paulheim, Heiko. ALOD2Vec Matcher. In: CEUR Workshop Proceedings OM 2018 - Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference (ISWC 2018). Monterey, CA, USA. 2018. [409]

Portisch, Jan; Hladik, Michael; Paulheim, Heiko. ALOD2Vec Matcher Results for OAEI 2020. In: The Fifteenth International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020), OM@ISWC 2020. 2020. [403]

Portisch, Jan; Paulheim, Heiko. ALOD2Vec Matcher Results for OAEI 2021. In: Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), OM@ISWC 2021. 2022. [411]

17.1 Presentation of the System

17.1.1 State, Purpose, General Statement

The *ALOD2vec Matcher* is an element-level, label-based matcher which uses a large-scale Web-crawled RDF dataset of hypernymy relations as general-purpose background knowledge. The dataset contains many tail-entities as well as instance data such as persons or places which cannot be found in common thesauri. In order to exploit the external dataset, a neural language model approach is used to obtain a vector for each concept contained in the dataset. This matching system was initially introduced at the OAEI 2018 [409] and also participated in the 2020 campaign [403]. The implementation is based on the *Matching EvaLuation Toolkit* [203] as well as the *KGvec2go* [404] REST API to obtain vector representations via a Web API.

17.1.2 Specific Techniques Used

After the basic concepts of this matcher are introduced (*Foundations*), the specific techniques applied are presented.

Foundations

WebIsALOD Dataset A frequent problem that occurs when working with external background knowledge is the fact that less common entities are not contained within a knowledge base. The *WebIsA* [467] database is an attempt to tackle this problem by providing a dataset that is not based on a single source of knowledge – like *DBpedia* [300] – but instead on the whole Web: The dataset consists of hypernymy relations extracted from the *Common Crawl*[1], a

[1] see `http://commoncrawl.org/`

freely downloadable crawl of a significant portion of the Web. A sample triple from the dataset is *european_union skos:broader international_organization*[2]. The dataset is also available via a *Linked Open Data* (LOD) endpoint[3] under the name *WebIsALOD* [198]. In the LOD dataset, a machine-learned confidence score $c \in [0, 1]$ is assigned to every hypernymy triple indicating the assumed degree of truth of the statement.

RDF2vec The background dataset can be viewed as a very large knowledge graph; in order to obtain a similarity score for nodes and edges in that graph, the *RDF2vec* [442] approach is used. It applies the *word2vec* [344, 345] model to RDF data: Random walks are performed for each node and are interpreted as sentences. After the walk generation, the sentences are used as input for the word2vec algorithm. As a result, one obtains a vector for each word, i.e., a concept in the RDF graph. Multiple flavors of RDF2vec have been developed in the past, such as biased walks [84] or *RDF2vec Light* [405].[4]

KGvec2go Training embeddings on large knowledge graphs can be computationally very expensive. Moreover, the resulting embedding models can be very large since a multidimensional vector needs to be persisted for every node in the knowledge graph. However, most downstream applications require only a small subset of node vectors. The *KGvec2go* project [404] addresses these problems by providing a free REST API[5] for pre-trained RDF2vec models on various large knowledge graphs (among which WebIsALOD is also available).

Monolingual Matching

ALOD2vec Matcher is a monolingual matching system. For the alignment process, the system retrieves the labels of all elements of the ontologies to be matched. A filter adds all simple string matches to the final alignment in order to increase the performance. The remaining labels are linked to concepts in the background dataset, are compared, and the best solution is added to the final alignment. A high-level view of the matching system is provided in Figure 17.1.

The first step is to link the obtained labels from the ontology to concepts in the WebIsALOD dataset. Therefore, string operations are performed on the

[2] see `http://webisa.webdatacommons.org/concept/european_union_`

[3] see `http://webisa.webdatacommons.org/`

[4] For a good overview of the RDF2vec approach and its applications, refer to `http://www.rdf2vec.org/`

[5] see `http://kgvec2go.org/api.html`

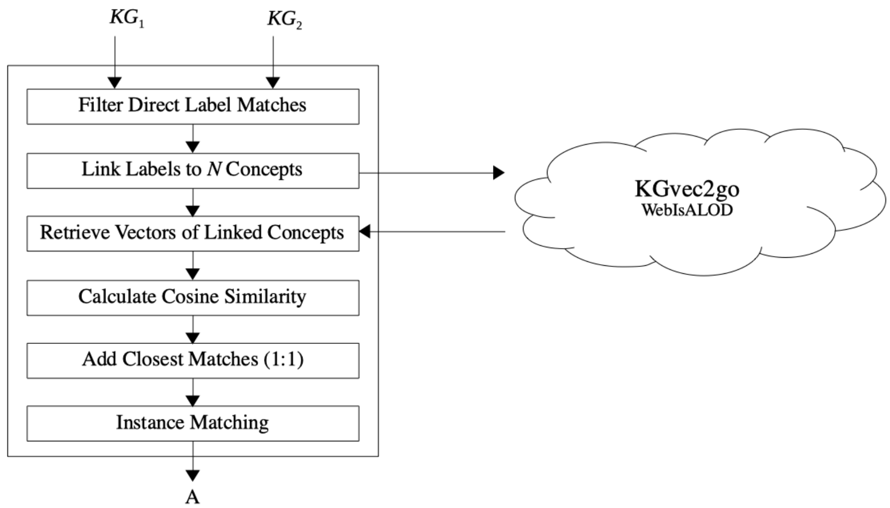


Figure 17.1: High-level view of the ALOD2vec matching process. KG_1 and KG_2 represent the input ontologies and optionally instances. The final alignment is referred to as A.

label, and it is checked whether the label is available in WebIsALOD. If it cannot be found, a token lookup is performed. Given two entities e_1 and e_2, the matcher uses their textual labels to link them to concepts $e_1^{'}$ and $e_2^{'}$ in the external dataset. Afterwards, the embedding vectors $v_{e_1^{'}}$ and $v_{e_2^{'}}$ of the linked concepts ($e_1^{'}$ and $e_2^{'}$) are retrieved via a Web request and the cosine similarity between those is calculated. Hence: $sim(e_1, e_2) = sim_{cosine}(v_{e_1^{'}}, v_{e_2^{'}})$. If $sim(e_1, e_2) > t$ where t is a threshold in the range of 0 and 1, a correspondence is added to a temporary alignment. In a last step, a one-to-one arity is enforced by applying a *Max Weight Bipartite Filtering* (MWB) [86] on the temporary alignment.

In order to consume the vectors in Java, a client has been implemented and contributed to the MELT-ML module. The KGvec2go REST API can now be accessed through class `KGvec2goClient`. Even though this matcher only uses the WebIsALOD dataset, the implementation supports all datasets accessible on KGvec2go. The extension is available by default in MELT 2.6.

Instance Matching

After classes and properties have been matched, instances are matched using a string index. The confidence score assigned to instances belonging to matched classes is higher than that of matches between instances belonging to non-matched classes.

Explainability

ALOD2vec Matcher provides an explanation for every correspondence that is added to the final alignment. Therefore, the extension capabilities of the alignment format [96] are used. Two concrete examples from the *Anatomy track* for explanations of the matching system are: "Label 'aqueous humour' of ontology 1 and label 'Aqueous Humor' of ontology 2 have a very similar writing." or "The following two label sets have a cosine above the given threshold: |lens|anterior|epithelium| and |anterior|surface|lens|". In order to explain a correspondence, the `description` property[6] of the *Dublin Core Metadata Initiative* is used.

17.1.3 Extensions to the Matching System for the 2021 Campaign

For the 2021 campaign, the matching system was adapted to use the latest MELT release and was packaged as MELT Web Docker[7] container. The 2021 implementation is publicly available on GitHub.[8]

17.2 Results

17.2.1 Anatomy Track

On the anatomy dataset, the system scores a precision of 0.828, a recall of 0.766, and an F_1 of 0.796.

17.2.2 Conference Track

On the conference track, the matcher achieves a recall of 0.49 and a precision of 0.64. The overall F_1 score on `ra1-M3` was 0.59.

17.2.3 Multifarm Track

Since the *WebIsALOD* dataset is only available in English, the focus of the *ALOD2vec Matcher* is on monolingual matching tasks.

[6] see `http://purl.org/dc/terms/description`
[7] see `https://dwslab.github.io/melt/matcher-packaging/web`
[8] see `https://github.com/janothan/ALOD2VecMatcher`

17.2.4 LargeBio Track

In its current version, the LargeBio track is too large for the matching system's architecture. There is a trade-off in package size and runtime performance (a large package with all vectors matches faster than the submitted small package, which obtains vectors at runtime from KGvec2go). The current architecture of *ALOD2vec Matcher* is not intended for large-scale matching – however, the matching algorithm itself could be used for large-scale matching.

17.2.5 Knowledge Graph Track

The system could complete all matching tasks in time. As in the previous year, this matcher obtains the second-best results achieving almost the same score as the *Wiktionary Matcher 2021* [413]. The overall F_1 score was 0.87 on the complete track.

17.2.6 Common Knowledge Graph Track

This year, a new track was added to the OAEI: The *Common Knowledge Graph Track* [136]. Although not optimized for this track, *Alod2vec Matcher* achieved the second-best result with an F_1 score of 0.89.

17.3 Conclusion

In this chapter, we presented the latest version of the *ALOD2vec Matcher*, a matcher utilizing an RDF2vec vector representation of the WebIsALOD dataset, as well as its results in the 2021 OAEI. In the future, the matching system could be improved by using another, potentially larger or newer, hypernymy database by exploiting other embedding algorithms[9], and by adding further matching strategies to the overall algorithms such as checking of logical constraints.

[9] Concerning the embedding strategy, e-RDF2vec$_{oa}$ can be expected to perform better than the classic RDF2vec embedding algorithm since the dataset consists of merely one edge type.

Chapter 18

Background Knowledge in Schema Matching: Strategy vs. Data

In this chapter, six general-purpose knowledge graphs are exploited as sources of background knowledge for the matching task. The background sources are evaluated by applying three different exploitation strategies. We find that explicit strategies still outperform latent ones and that the choice of the strategy has a greater impact on the final alignment than the actual background dataset on which the strategy is applied. While we could not identify a universally superior resource, BabelNet achieved consistently good results. The best matcher configuration with BabelNet performs very competitively when compared to other matching systems even though no dataset-specific optimizations were made.

The work presented in this chapter has been published before as: Portisch, Jan; Hladik, Michael; Paulheim, Heiko. Background Knowledge in Schema Matching: Strategy vs. Data. In: Proceedings of the International Semantic Web Conference (ISWC 2021). 2021. [406]

18.1 Introduction

Ontology matching or *schema matching* is the non-trivial task of finding correspondences between entities of two or more given ontologies or schemas. The matching can be performed manually or through the use of an automated matching system. In both cases, the context is very important, and concept knowledge is required. Therefore, automated matching systems require background knowledge to excel at the schema matching task. In most cases, *WordNet* is used as a form of general concept knowledge with a plain synonym lookup strategy. However, over the last decade, many other sources of background knowledge that are much larger and also contain instance data have emerged. In addition, strategies to exploit structured knowledge, such as knowledge graph embedding models, have been developed but are rarely used in ontology matching. Exploiting background knowledge for ontology matching is still one of multiple challenges that are yet to be solved [480].

In this chapter, we compare the performance of six different background datasets of varying sizes and characteristics for the task of schema matching. For each dataset, three different strategies are exploited. Besides an in-depth evaluation of the matching performance, we strive to test the following hypotheses:

H1 The strategy is more important than the resource.
H2 The resource is more important than the strategy.
H3 There is a superior resource.
H4 There is a superior strategy.

The remainder of this chapter is structured as follows: In the next section, we present an overview of related work. Section 18.3 describes the general evaluation architecture that is used, as well as the generic matching process that was implemented for this chapter. The background datasets and the strategies that are explored are presented in Sections 18.4 and 18.5, respectively. The strategies on the background knowledge datasets are evaluated on four different gold standards in Section 18.6. The chapter closes with a summary and an outlook on future work.

18.2 Related Work

Ontology and schema matching systems have been evaluated by the *Ontology Alignment Evaluation Initiative* (OAEI)[1] every year since 2005. While, to our knowledge, there is no large comparison of different general knowledge background sources or exploitation strategies, many individual matching systems exist that make use of external background knowledge. In 2013, Euzenat and Shvaiko [128] counted more than 80 schema matching systems that exploit *WordNet*. Besides WordNet, few other general background data sources are used: The *WikiMatch* [196] system exploits the *Wikipedia* search API by determining concept similarity through the overlap of returned Wikipedia articles for a search term. *WeSeE Match* [383] queries search APIs and determines similarity based

[1] `http://oaei.ontologymatching.org/`

on TF-IDF scores on the returned Web site titles and excerpts. A synonymy and translation lookup strategy based on *Wiktionary* is used in [410] for monolingual and multilingual matching. Lin and Krizhanovsky [314] exploit *Wiktionary* for translation lookups within a larger matching system.

In the biomedical and life science domain, specialized external background knowledge is broadly available and heavily exploited for ontology matching. Chen et al. [81] extend the LogMap matching system to use *BioPortal*, a portal containing multiple ontologies, alignments, and synonyms, by (i) applying an overlap-based approach as well as by (ii) selecting a suitable ontology automatically and using it as mediating ontology. As mappings between biomedical ontologies are available, those are used as well: Groß et al. [171] exploit existing mappings to third ontologies, so-called *intermediate ontologies*, to derive mappings. This approach is extended by Annane et al. [22] who use BioPortal by exploiting existing alignments between the ontologies found there for matching through a path-based approach: By linking source and target concepts into the global mapping graph, the paths that connect the concepts in that graph are used to derive new mappings. In the same domain, research has also been conducted on background knowledge selection. Faria et al. [146] propose the usage of a metric, called *Mapping Gain* (MG), which is based on the number of additional correspondences found given a baseline alignment. Quix et al. [425] use a keyword-based vector similarity approach to identify suitable background knowledge sources. Similarly, Hartung et al. [187] introduce a metric, called *effectiveness*, which is based on the mapping overlap between the ontologies to be matched. While in the biomedical domain, many specialized resources are available, and data schemas are heavily interlinked, this is not the case for other domains. As a consequence, such methods cannot be easily translated and applied.

Background knowledge sources are also used for multilingual matching tasks. Here, translation APIs are often used such as *Microsoft Bing Translator* by *KEPLER* [254] or *Google Translator* by *LogMap* [241].

Approaches that exploit vector representations of concepts are rarely found in the ontology or schema matching domain. The *DOME* [200] matching system employs a *doc2vec* [297] approach to concepts within the ontologies to be mapped. Similarly, *AnyGraphMatcher* [329] attempts to embed the ontologies to be mapped at runtime but achieves very low results in the OAEI 2019. *DESKMatcher* [351] applies a knowledge graph embedding approach on external knowledge but did not perform competitively in the OAEI 2020 either. *WebIsAlod* is exploited as external background knowledge in [403] through a combined string matching and graph embedding strategy.

These examples show that there is a larger body of works exploiting background knowledge with various strategies[2]; however, they are always used in the context of a larger matching system. Ablation studies and, therefore, statements about the utility of a particular source and/or strategy are not available.

18.3 General Approach

To close this gap, we propose a simple, generic matching process that can work with different sources of background knowledge and exploitation strategies. Our aim is *not* to build a top-performing matching system but to provide a testbed for a fair comparison of different background knowledge sets and strategies.

18.3.1 Overview

Figure 18.1 depicts the architectural evaluation setting: A generic *matcher* accepts two ontologies and outputs an alignment. Thereby, it applies a *strategy* that can be exchanged independently of other matcher settings. Given labels, the matcher can ask a generic *linker* whether a concept is available in a background knowledge source. Depending on the request type, the linker returns one or more corresponding concepts from the background knowledge. For *Wiktionary*, for instance, the matcher can ask for concept `European Union` and the linker would return `dbnary-eng:European_Union`. This linking process is also known as *anchoring* or *contextualization* [132]. Now that the matcher knows the representation in the background knowledge set, it can request further information through a generic *resource wrapper* (such as similarities between concepts). Therefore, a *resource* and a corresponding *linking process* (that is wrapped by the *linker*) have to be set. The implementation allows to change the *resource* and the *linking process* independently of other matcher settings such as the *strategy*.

18.3.2 Matching Process

The matching process can be divided into two parts: linking and matching. The linking operation is implemented as a three-step process: (i) *Full Label Linking*, (ii) *Longest Token Linking*, and (iii) *Token Linking*. Later linking steps are only performed when the previous step was not able to link the label. In step (i), the full, i.e., unchanged, label is linked to a concept in the background knowledge source. Often, labels are composite concepts that do not appear in the

[2] For a more complete review of the research field, we direct the reader to Chapter 3 of this dissertation.

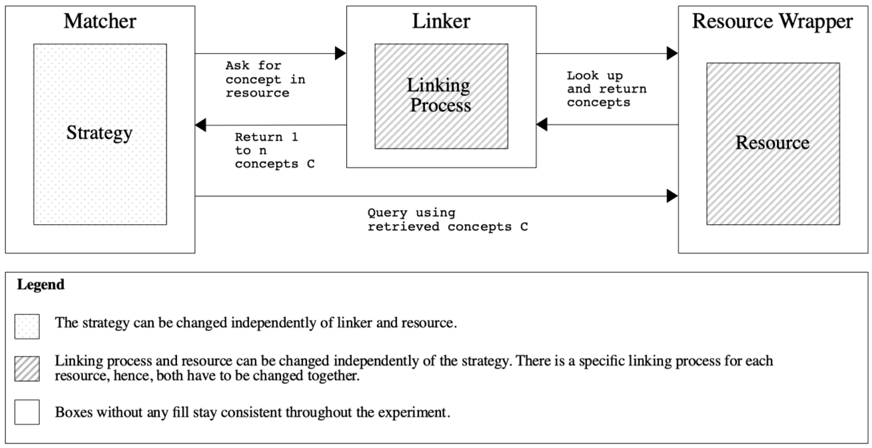


Figure 18.1: Architectural setting to evaluate different background datasets exploiting different strategies.

knowledge source as a whole but in parts. To cover this case, step (ii) tokenizes labels and truncates them from the right. Linked parts are removed, and the process is repeated to check for further concepts. This allows the detection of long subconcepts even if the full string cannot be found. Label *conference banquet*, for example, cannot be linked to the Wiktionary background dataset using the full label. However, by applying right-to-left truncation, the label can be linked to two concepts, namely *conference* and *banquet*, and in the following also be matched to concept *conference dinner*, which is linked in the same fashion. The last fallback strategy is token linking (iii) which tokenizes each label (using spaces, underscores, and camel case recognition) and links the individual tokens to the background dataset.

After completion of the linking process, the match operation is performed. Multiple strategies are implemented here (see Section 18.5), which operate on the links. For the synonymy strategy, a match would be, for instance, annotated for *(person, individual)* given that the two labels are synonymous according to the background dataset employed. If there are multiple links (linking steps (ii) and (iii)), a match requires that every link has a matching partner (according to the strategy applied) in the set of links of the other label. In order to obtain a one-to-one alignment, the Hungarian extraction method [286] is applied.

The overall matching runtime performance is improved by adding string matches directly to the final alignment. This step runs independently of the strategy or the background dataset used. It does not skew the outcome because all strategies under consideration in this chapter are purely label-based. Hence, the same label used for two entities would always lead to a match.

Overall, the matching process scales with $O(nm)$ where n is the number of elements in one ontology and m is the number of elements in the other ontology.[3] It is important to note that the scalability can be improved by adding a candidate pre-selection/blocking component. However, since scalability is not the main concern of this chapter, we decided against complicating the matching pipeline.

The matcher is implemented using the *Matching EvaLuation Toolkit* [203, 204] *(MELT)*[4], an open-source Java framework for matcher development, tuning, evaluation, and packaging recommended by the OAEI. The matcher is implemented so that it is possible to use different sources of background knowledge and different strategies within the matching process. The implementation of this chapter (linker, background sources, significance evaluation) has been unit tested, documented, and contributed to the framework so that other researchers can use the matching parts of the implementation (e.g., to easily use Wikidata synonyms/hypernyms through an API) for their matching system.[5]

18.4 Background Datasets

For this chapter, six knowledge graphs are exploited as background knowledge within the matching process. They are quickly introduced in the following:

BabelNet [362] is a large multilingual knowledge graph that integrates (originally) Wikipedia and WordNet. Later, additional resources such as Wiktionary were added. The integration between the resources is performed in an automated manner. The dataset does not just contain lemma-based knowledge but also instance data (named entities) such as the singer and songwriter *Trent Reznor*. For the embedding strategy, the RDF version of BabelNet 3.6 was used[6], for the other strategies, the BabelNet 4.1 indices.

[3]The size of the external resource is not relevant within the matching process since all similarity functions applied here are lookup-based. When training an embedding with the external resource, the size of the resource affects scalability; however, the training is a one-time process – once the vectors are available, they can be reused in all other matching tasks.

[4]`https://github.com/dwslab/melt/`

[5]`https://dwslab.github.io/melt/matcher-development/with-background-knowledge`

[6]Unfortunately, there is no RDF version of the latest BabelNet version.

Wiktionary is a "collaborative project run by the Wikimedia Foundation to produce a free and complete dictionary in every language"[7]. The project is organized similarly to Wikipedia: Everybody can contribute and edit the dictionary. The content is reviewed in a community process. Like Wikipedia, Wiktionary is available in many languages.

DBnary [470] is an RDF version of Wiktionary that is publicly available.[8] The DBnary dataset makes use of an extended LEMON model [335] to describe the data. For this work, a recent download from March 2021 of the English Wiktionary has been used.

WebIsALOD is a large hypernymy graph based on the *WebIsA* database [467]. The latter is a dataset, which consists of hypernymy relations extracted from the *Common Crawl*, a large set of crawled Web pages. The extraction was performed in an automatic manner through Hearst-like [190] lexico-syntactic patterns. For example, from the sentence "[...] added that the country has favorable economic agreements with major economic powers, including the European Union.", the fact `isA(european_union, major_economic_power)` is extracted.[9] *WebIsALOD* [198] is the Linked Open Data endpoint that allows querying the data in SPARQL.[10] In addition to the endpoint, machine learning was used to assign confidence scores to the extracted triples. For this work, a confidence threshold of 0.5 for hypernymy relations was chosen. The dataset of the endpoint is filtered, i.e., it contains a subset of the original *WebIsA* database, to ensure higher data quality. The knowledge graph contains instances as well as more abstract concepts that can also be found in a dictionary.

WordNet [149] is a well-known and heavily used database of English words that are grouped in sets, which represent one particular meaning, so-called *synsets*. The resource is strictly authored. *WordNet* is publicly available, included in many natural language processing frameworks, and often used in research. An RDF version of the database is also available for download and was used for this work.[11]

Wikidata [550] is a collaboratively built knowledge base containing more than 93 million data items. Like Wikipedia and Wiktionary, the project is run by the Wikimedia Foundation. It is publicly available[12] and under a permissive license. For this work, a download from March 2021 has been used.

---

[7] `https://web.archive.org/web/20190806080601/https://en.wiktionary.org/wiki/Wiktionary/`

[8] `http://kaiko.getalp.org/about-dbnary/download/`

[9] `http://webisa.webdatacommons.org/417880315`

[10] `http://webisa.webdatacommons.org/`

[11] `http://wordnet-rdf.princeton.edu/about/`

[12] `https://www.wikidata.org/wiki/Wikidata:Main_Page`

DBpedia [300] is a knowledge graph that is extracted from Wikipedia infoboxes. The underlying RDF files are available for download. For this work, the latest available files as of March 2021 have been downloaded via the DBpedia Databus[13] (rather than the 2016-10 version of DBpedia that is often used).

18.5 Strategies

In the following, the exploitation strategies applied to the datasets outlined in the previous section are introduced.

18.5.1 Synonymy

The synonymy strategy exploits existing synonymy relations in the datasets. On *Wiktionary*, for instance, *tired* is explicitly named as a synonym for *sleepy*. Given two entities $e_1 \in O_1$ and $e_2 \in O_2$ of two ontologies O_1 and O_2, a match is annotated if the synonymy relation holds between at least one pair of their labels l_{e_1} and l_{e_2} according to the background dataset B that is used. This is depicted in Equation 18.1.

$$isMatch_B(e_1, e_2) = isSynonymous_B(l_{e_1}, l_{e_2}) \tag{18.1}$$

The WebIsALOD dataset does not contain explicitly stated synonyms. Here, a synonym is assumed if both labels l_{e_1} and l_{e_2} appear as hypernyms of each other, as shown in Equation 18.2. This occurs more often than one might assume due to the automatic extraction process that is applied to create this knowledge graph.[14] The intuition behind the assumption here is that two things, X and Y are describing the same thing if it was stated on the Web that X is a Y and that Y is an X.

$$isMatch_{\text{WebIsALOD}}(e_1, e_2) = isHypernymous(l_{e_1}, l_{e_2}) \wedge isHypernymous(l_{e_2}, l_{e_1}) \tag{18.2}$$

For DBpedia, the properties `rdfs:label`, `foaf:name`, `dbo:alias`, `dbp:name`, and `dbp:otherNames` are used to obtain labels, and two entities are considered synonymous if they have at least one label in common. On Wikidata, we use `rdfs:label` and `skos:altLabel` to obtain labels, and determine synonymy with the same mechanism.

[13] `https://databus.dbpedia.org/`

[14] For example, *symposium* and *conference* are mutual hypernyms of each other in WebIsALOD.

18.5.2 Synonymy and Hypernymy

The synonymy and hypernymy strategy exploits the synonymy relations in the background datasets and, in addition, the hypernymy relations. Given two labels l_{e_1} and l_{e_2} of two entities e_1 and e_2, a match is annotated if one of the semantic relations holds between the two labels as depicted in Equation 18.3.

$$\begin{aligned} isMatch_B(e_1, e_2) = isSynonymous_B(l_{e_1}, l_{e_2}) \\ \vee\ isHypernym_B(l_{e_1}, l_{e_2})\ \vee\ isHypernym_B(l_{e_2}, l_{e_1}) \end{aligned} \tag{18.3}$$

For DBpedia, the properties `rdf:type` and `dbo:type` are used to obtain hypernyms. On Wikidata, we use `wdt:P31` (instance of) and `wdt:P279` (subclass of).

18.5.3 Knowledge Graph Embeddings

Knowledge graph embeddings, i.e., the vector-based representation of the elements within a knowledge graph, are a very active research area in recent years. Many such methods are known [263]. For this chapter, we exploit the *RDF2-vec* [442] approach: Random walks through the knowledge graph are generated starting from each node. The walks include the named edges of the graph. After the walk generation, the *word2vec* [344, 345] algorithm is applied. Thereby, a vector representation for each node and each edge is obtained. This embedding approach has been chosen due to its simplicity, its good performance on a multitude of tasks (rather than being developed for only one task, RDF2vec is task agnostic), its previous usage in ontology matching, and its scalability. It is important to note that the background knowledge source is transformed into a vector space – not the ontologies that are to be matched.
Two entities $e_1 \in O_1$ and $e_2 \in O_2$ of two different ontologies O_1 and O_2 are matched if their labels l_{e_1} and l_{e_2} can be mapped to a vector $v_{l_{e1}}$ and $v_{l_{e2}}$ in the background knowledge dataset B and the cosine similarity sim between the two vectors is larger than a predefined threshold t. Hence:

$$isMatch_B(e_1, e_2) = sim(v_{l_{e1}}, v_{l_{e2}}) > t \tag{18.4}$$

For WebIsALOD and WordNet, the pre-trained models from *KGvec2go*[15] [404] were used. The models were trained with the same configuration and, therefore, allow for comparability. Embeddings for the other three graphs are not available for download and were trained specifically for this chapter.

Despite good scalability behavior of the embedding approach, vector representations for BabelNet, Wikidata, and DBpedia could not be calculated within

[15] `http://kgvec2go.org/`

ten days. Therefore, *RDF2vec Light* [405] (see Chapter 9) was used for those very large knowledge graphs. The variant is based on the notion that, given a concrete task, only a small set of nodes within a knowledge graph are of actual interest. For example, given the matching task within the anatomy domain, a vector representation of *Year Zero*, a music album by the industrial rock band *Nine Inch Nails*, is not of particular interest. Therefore, a set of nodes of interest is defined in advance, and walks are only generated for those. For ontology matching, the set of nodes of interest is known through the linking operation. Experiments showed that the performance of the light variant yields good results on various machine learning tasks compared to the classic variant [405]. For this work, the following parameters have been used: 500 walks per node, $depth = 4$ (i.e., 4 node hops), *SG* variant, $window = 5$, and $dimension = 200$. For the matcher configuration, a threshold of $t = 0.7$ was used.

18.5.4 Combination of Sources

The combination strategy exploits all datasets at the same time with the strategies mentioned above. For the synonymy strategy, a match is annotated if any background dataset finds evidence for a synonymy relation. The same logic is also applied in the synonymy and hypernymy strategy and the embedding strategy.

18.6 Evaluation

We evaluate all combinations of the strategies presented in Section 18.5 and background datasets presented in Section 18.4 on four evaluation datasets: (i) *OAEI Anatomy* [42], (ii) *OAEI Conference* [77], (iii) *SAP FS* [401], and (iv) *Large-Bio*. The experiments were performed on a 24 core server (à 2.6 GHz) with 386Gb of RAM running Debian 10.

18.6.1 Evaluation Datasets

Dataset (i) consists of two anatomical ontologies where the human anatomy has to be mapped to the anatomy of a mouse. The *Conference* dataset (ii) consists of 16 ontologies from the conference domain and 120 alignment tasks between them. Out of those, 21 reference alignments are publicly available. The results reported in this chapter refer to the available alignments. In order to allow for comparability with other matching systems, micro averages are reported; those are also reported by the *OAEI Conference* track organizers. The *SAP FS* dataset (iii) is a proprietary evaluation dataset from the banking and insurance industry consisting of 5 matching tasks. The ontologies in that dataset have been derived from conceptual data models. The dataset has been provided to the authors of this chapter for research purposes by *SAP SE Financial Services*. In order to allow for comparability with the numbers reported in the original chapter, macro averages are reported here. From the LargeBio track (iv), the FMA/NCI small test case is used for the evaluation here. Overall, 21 matching system variants are evaluated on four tracks with a total of 28 test cases.

18.6.2 Evaluation Metrics

The alignments are evaluated using precision, recall, and F_1, which is the harmonic mean of the latter two. In addition, it is evaluated whether the alignments obtained by the different strategy-source combinations are significantly different. Therefore, a significance metric is required. For this work, we use McNemar's significance test as proposed by Majid et al. [348]: Be R the reference alignment and A_1, A_2 two system alignments. We can now calculate the two relevant elements from the contingency table as follows:

$$
\begin{aligned}
n_{01} &= |(A_2 \cap R) - A_1| + |A_1 - A_2 - R| \\
n_{10} &= |(A_1 \cap R) - A_2| + |A_2 - A_1 - R|
\end{aligned}
\tag{18.5}
$$

The significance can then be determined using McNemar's asymptotic test with continuity correction:

$$
\chi^2 = \frac{(|n_{01} - n_{10}| - 1)^2}{n_{01} + n_{10}} \tag{18.6}
$$

For a small sample size ($n = n_{01} + n_{10}$; $n < 25$), McNemar's exact test has to be used to obtain the p value:

$$
p = \sum_{x=n_{01}}^{n} (\binom{n}{x})(\frac{1}{2})^2 \tag{18.7}
$$

For this chapter, a significance level alpha of $\alpha = 0.05$ was chosen. As a side contribution of this work, the evaluation code for significance testing has been contributed to the MELT framework [203] to facilitate reuse by other researchers.

18.6.3 Results

The performance results in terms of precision, recall, and F_1 are presented in Table 18.1. The number of significantly different test case alignments is given in Figure 18.2. More detailed performance and significance statistics, as well as all alignments, are available for download.[16] It can be seen that the synonymy strategy consistently achieves the highest precision throughout all background knowledge resources. In terms of F_1, the synonymy strategy performs best in most cases when evaluating the strategy on each background source separately. The only area where the synonymy strategy falls short is recall. The significance tests show that despite similar scores, the alignments within this strategy group are significantly different in 285 out of 588 cases. This is also visible in Figure 18.2, which shows the number of significantly different alignments (given two matching systems). From the figure, it can be seen, for instance, that there are 22 significantly different alignments between DBpedia and Wiktionary using the synonymy strategy, but only five different alignments between DBpedia and the combination approach using the synonymy strategy.

With the exception of BabelNet, the addition of hypernyms increases recall.[17] However, a drop in precision leads to overall lower F_1 scores (with the exception of DBpedia on SAP FS and Wikidata on FMA/NCI). The results indicate that hypernyms could be used in more complex matching strategies, e.g., as part of candidate generation. Nonetheless, a naïve merge of synonymy and hypernymy sets as the main strategy is not generally suitable for precise matching on the given evaluation datasets.

The embedding-based matching approach falls short of performing competitively. While the recall can be increased in some cases, the method generally scores a significantly lower precision leading to an overall low F_1 score. One likely reason for the bad performance of the embeddings is that the RDF2vec vector similarity seems to be an indicator of relatedness rather than actual concept similarity – an observation that has also been made earlier [404]. More promising usage scenarios for the embedding models exploited in this chapter are likely candidate selection and hybrid strategies. Concerning significance testing, the embedding strategies produce the most significantly different alignments of all strategies evaluated in this chapter. In addition, it was observed that embedding large background knowledge datasets is computationally very expensive, which does not apply to the matching run time after the models were trained.

Concerning the choice of background knowledge, WordNet, Wiktionary, and BabelNet are similar in the sense that they are focused on lexical facts. BabelNet, the largest of the three, scores the overall best F_1 score on Anatomy and Conference. On the remaining two tracks, the performance is competitive.

[16] `https://github.com/janothan/bk-strategy-vs-data-supplements/`

[17] This may seem odd at first. However, lower recall values are due to the Hungarian optimization method to obtain a 1:1 alignment, which, in that case, extracts more false positives.

Table 18.1: Evaluation results of six different background knowledge datasets exploiting three different strategies on four different gold standards. Note that for the *Conference* task, micro averages are used, while for the *SAP FS* task, macro averages are reported. Baselines are given as reported by the OAEI organizers (FMA/NCI baselines are not provided).

		Anatomy			Conference			SAP FS			FMA/NCI (small)		
Knowledge Graph	**Strategy**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
BabelNet	**SYN**	0.947	0.752	**0.838**	0.677	0.566	**0.617**	0.404	0.153	0.222	0.909	0.370	0.526
	SYN + HYP	0.935	0.755	0.835	0.562	0.534	0.548	0.358	0.153	0.214	0.869	0.366	0.515
	RDF2vec (light)	0.514	0.697	0.592	0.312	0.269	0.289	0.206	0.138	0.165	0.312	0.269	0.289
WebIsALOD	**SYN**	0.967	0.692	0.807	0.659	0.495	0.566	0.468	0.144	0.220	**0.979**	0.265	0.417
	SYN + HYP	0.915	0.695	0.790	0.458	0.537	0.494	0.349	0.146	0.206	0.867	0.265	0.406
	RDF2vec	0.789	0.701	0.742	0.269	0.541	0.359	0.261	0.140	0.182	0.637	0.269	0.378
Wiktionary	**SYN**	**0.968**	0.712	0.820	0.691	0.534	0.603	0.459	0.146	0.221	0.977	0.282	0.437
	SYN + HYP	0.967	0.712	0.820	0.675	0.538	0.599	0.459	0.146	0.221	0.972	0.282	0.437
	RDF2vec	0.644	0.706	0.674	0.249	0.525	0.338	0.286	0.143	0.190	0.477	0.283	0.355
WordNet	**SYN**	0.964	0.715	0.821	**0.722**	0.528	0.610	0.415	0.148	0.218	0.963	0.322	0.483
	SYN + HYP	0.929	0.722	0.813	0.614	0.531	0.569	0.374	0.151	0.216	0.877	0.322	0.472
	RDF2vec	0.733	0.709	0.721	0.505	0.531	0.531	0.380	0.139	0.204	0.626	0.306	0.411
DBpedia	**SYN**	0.936	0.699	0.800	0.709	0.495	0.583	0.497	0.142	0.221	0.917	0.356	0.512
	SYN + HYP	0.936	0.699	0.800	0.709	0.495	0.583	**0.537**	0.156	**0.242**	0.917	0.356	0.512
	RDF2vec (light)	0.594	0.707	0.646	0.154	0.515	0.238	0.225	0.152	0.181	0.468	0.344	0.397
Wikidata	**SYN**	0.924	0.741	0.823	0.636	0.544	0.587	0.447	0.163	0.239	0.894	0.340	0.492
	SYN + HYP	0.924	0.741	0.823	0.636	0.544	0.587	0.446	0.163	0.239	0.894	0.340	0.493
	RDF2vec (light)	0.546	0.741	0.629	0.125	0.544	0.203	0.204	0.147	0.171	0.354	0.321	0.337
Combinations	**SYN**	0.882	0.788	0.832	0.496	0.577	0.533	0.376	0.178	0.241	0.829	0.477	**0.605**
	SYN + HYP	0.815	**0.791**	0.803	0.339	0.593	0.431	0.301	0.179	0.224	0.727	**0.481**	0.579
	RDF2vec	0.241	0.786	0.369	0.050	**0.620**	0.242	0.139	**0.194**	0.162	0.161	0.418	0.232
Baseline	-	0.997	0.622	0.766	0.760	0.410	0.530	0.520	0.150	0.230	-	-	-

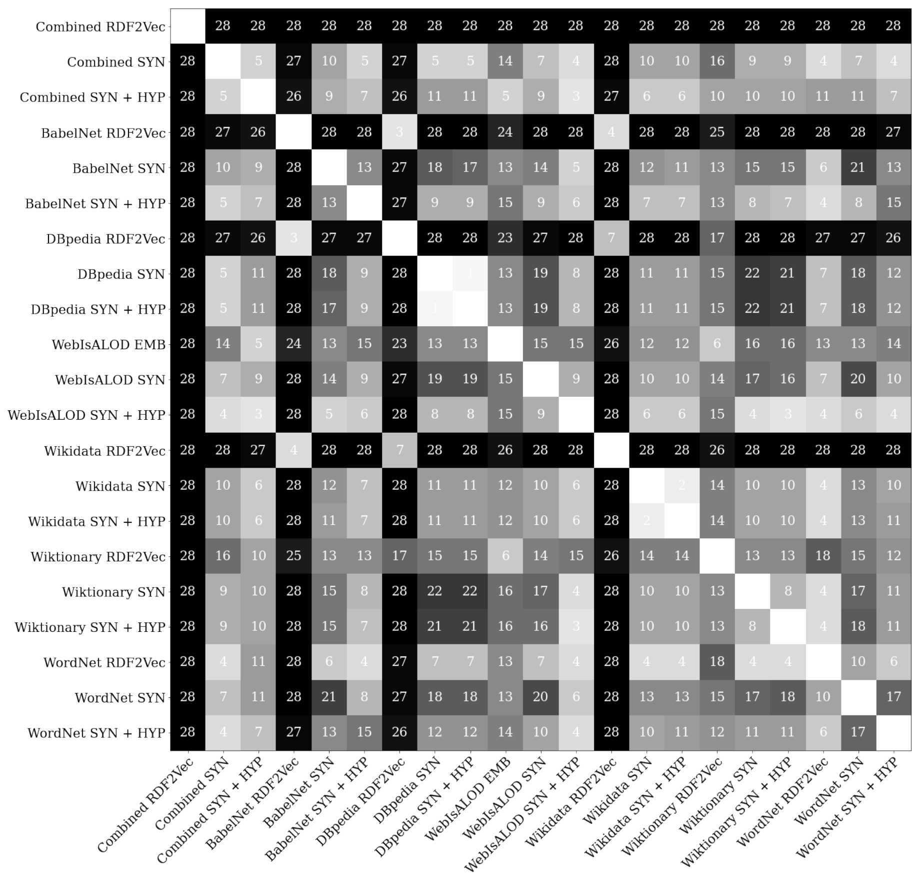


Figure 18.2: Matrix with the number of significantly different test case alignments given two matcher configurations. A higher total number of significantly different test case alignments has a darker shading in the figure. In total, there are 28 test cases.

Despite its small size, WordNet also achieves competitive results compared to Wiktionary on Anatomy, Conference, and SAP FS and outperforms the latter significantly on the LargeBio task. Nevertheless, unlike WordNet, Wiktionary and BabelNet are constantly growing over time due to a community-driven creation process and might outperform WordNet in the long run.

DBpedia performs in the mid-range in terms of F_1. The recall is lower than that of the better-performing systems (BabelNet, Wiktionary, WordNet). The most likely explanation is a lower concept coverage since DBpedia contains rather instances than class concepts. Interestingly, the addition of hypernyms rarely has any effect on this particular background source.

Wikidata performed similarly to DBpedia. Like the latter dataset, the addition of hypernyms does not change the results significantly.

The WebIsALOD dataset achieves the lowest overall results. The most likely reason is that the dataset is not authored but automatically built leading to a lot of noise contained in the dataset (wrong hypernyms). The comparatively bad performance of the synonymy strategy may be grounded in the fact that WebIsALOD is the only graph evaluated here that does not explicitly state synonyms – but instead, those are derived, as outlined before, which is less precise.

The combination of different background knowledge sources increases the recall in all cases. Except on the LargeBio dataset, the drop in precision cannot make up for increases in recall.

When comparing the performance numbers on the evaluation dataset level, it can be seen that the *Anatomy* matching task achieves the best results – this is likely due to a high textual overlap of the labels. On the *Conference* task, the matchers achieve a lower precision and recall score. These observations are in line with those at *OAEI* campaigns. On the domain-specific *SAP FS* dataset, it can be seen that recall and precision scores are low. Likely explanations here are a domain-specific vocabulary, low explicitness of knowledge (many semantic details are hidden in lengthy descriptions), as well as a complex many-to-many matching problem (see [401] for details).

It is important to note that the work presented here is not intended to be a full-scale matching system but rather a comparison of different background knowledge datasets and exploitation strategies. Nonetheless, the performance of the best matching results achieved here on *Anatomy* and *Conference* are comparable to *OAEI* matching results reported in the most recent 2020 campaign. A comparison in terms of F_1 is depicted in Figure 18.3. It can be seen that the best configuration of this chapter performs in both cases above the median of the systems submitted in 2020. On Anatomy, it is noteworthy that the first three systems (AML, Lily, and LogMapBio) use domain-specific resources leading to an advantage over the general-purpose resources exploited in this work.

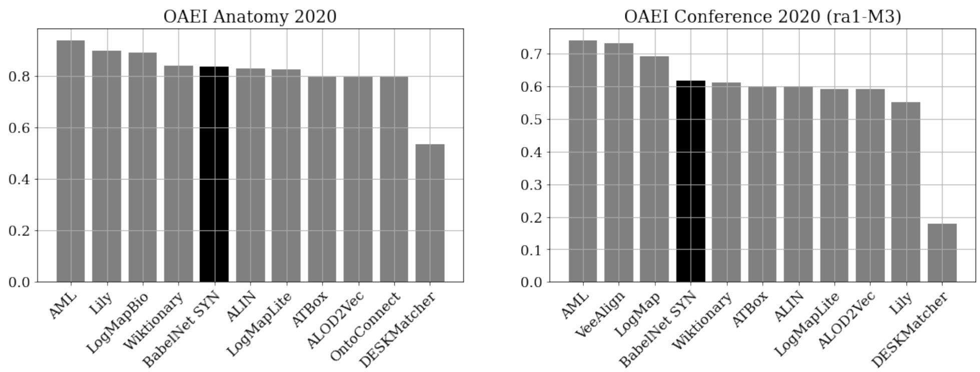


Figure 18.3: Performance in terms of F_1 on the OAEI Anatomy and Conference tracks of 2020.

Hypotheses

In order to evaluate hypotheses 1 and 2, we averaged the relative share of significantly different alignments on all test cases (i) while keeping the background source constant and changing the strategy (Equation 18.8) and (ii) while keeping the strategy constant and changing the background source (Equation 18.9):

$$impact_{strategy} = \frac{\sum_{bk \in BK} \frac{\sum_{tc \in TC} \sum_{s_1 \in S} \sum_{s_2 \in S} sig(m(bk,s_1),m(bk,s_2))}{|TC|*|S|^2-|TC|*|S|}}{|BK|} \tag{18.8}$$

$$impact_{source} = \frac{\sum_{s \in S} \frac{\sum_{tc \in TC} \sum_{bk_1 \in BK} \sum_{bk_2 \in BK} sig(m(bk_1,s),m(bk_2,s))}{|TC|*|BK|^2-|BK|*|S|}}{|S|} \tag{18.9}$$

where S is the set of strategies, BK is the set of background sources, $sig(alignment_1, alignment_2)$ is the significance function which will return 1 if the two provided alignments are significantly different and else 0, and $m(bk, s)$ is the matching function which returns the alignment by using the specified background knowledge source bk and strategy s.

While keeping the background knowledge source constant and changing the strategy, we observed, on average, 57.5% significantly different alignments with a standard deviation of $\sigma = 0.163$. On the other hand, while keeping the strategy constant and changing the background knowledge source, we obtained, on average, 51.76% significantly different alignments with a standard deviation of

$\sigma = 0.181$. Given our experimental setup, we hence accept H1 and reject H2 since a variation in the strategic component has a higher impact on the alignments than a variation of the background sources under consideration in this study. It is noteworthy that both components lead, on average, to more than 50% significantly different alignments. Since our results do not indicate that there is a superior resource over all test sets, we can reject H3. However, it is noteworthy that BabelNet achieves consistently good (on two tracks the best) results in terms of F_1 when using the synonymy strategy. Similarly, we do not find a superior strategy over each and every single test case and reject H4 – but yet, the synonymy strategy achieved the best F_1 score on 3 out of 4 tracks and consistently performed very well compared to the other strategies.

18.7 Conclusion

In this chapter, we evaluated three different matching strategies using six different general-purpose knowledge graphs on various evaluation datasets. We find that the strategy influences the final alignment more than the underlying dataset. Given the strategies evaluated here, those exploiting explicitly stated knowledge outperform a latent strategy. However, the exploitation of graph embeddings for data integration and schema matching is novel, and its performance is still very low. While no superior general knowledge dataset could be identified, BabelNet produced consistently good or the best results. The humanly verified datasets outperformed the automatically generated one. Concerning the level of authoring between the datasets, the results indicate no clear superiority of expert-validated knowledge graphs over those created and validated by an open community.

Chapter 19

Business Applications

We close Part IV with a brief look at concrete business applications for approaches and findings presented in this dissertation. In Section 19.1, a matching system is presented, which assigns domain-specific concepts from the financial services industry to a predefined set of broader concepts from the *Financial Industry Business Ontology* (FIBO). Albeit not a pure ontology matching task, the task is undoubtedly very related and a good case in point for the application of background-based matching technology in an industry application: The automated categorization of financial instruments is a real-world business use case. In Section 19.2, a prototype for a business schema matching system is presented, which was developed in the course of this PhD project. The software pilot demonstrates the value of semi-automated schema matching in business and was also evaluated by some SAP customers.

Parts of the work presented in this chapter have been published before as: Portisch, Jan; Hladik, Michael; Paulheim, Heiko. FinMatcher at FinSim-2: Hypernymy Detection in the Financial Services Domain using Knowledge Graphs. In: Workshop on Financial Technology on the Web (FinWeb) in conjunction with The Web Conference. 2021. [407]

19.1 FinMatcher

19.1.1 Introduction

A *hypernym* or *hyperonym* is a concept that is superordinate to another one.[1] In computer science, it is often represented as an *IS-A* relationship. For example,

[1] For a detailed introduction into paradigmatic relations, see Subsection 2.2.2.

animal is a hypernym of *cat* and *equity index* is a hypernym of *S&P 500 Index*. A *hyponym*, on the other hand, is a concept that is subordinate to another one. For example, *cat* is a hyponym of *animal* and *S&P 500 Index* is a hyponym of *equity index*. [358] Hypernymy detection can be broadly applied in real-world applications. The detection of hypernyms in the financial services domain is particularly interesting due to a domain-specific vocabulary and a lack of publicly available domain-specific resources and concept representations.

The FinSim task models the hypernym detection task as a multi-class classification problem: Given a concept label (i.e., the hyponym), the correct hypernym is to be found from a set of 10 mutually exclusive classes (i.e., hypernyms). A system participating in this task can return a sorted list of classes. The task is evaluated with two performance metrics: mean rank and accuracy.

The FinMatcher system uses two very broad publicly available knowledge graphs (Wikidata and WebIsALOD) as well as a small linguistic graph resource (WordNet). A knowledge graph contains real-world entities from various domains and the relationships that hold between them in a graph format [384]. The system presented in this section calculates multiple explicit features and uses RDF2vec embeddings obtained from WebIsALOD. The features are concatenated into a feature vector which is presented to a neural classifier that was trained with the provided FinSim training data.

In the following subsection, related work is introduced. Afterward, the provided dataset is quickly described. In Subsection 19.1.4, the FinMatcher system is presented. The results of the FinSim task are given in Subsection 19.1.5 together with an ablation study. The section is concluded in Subection 19.1.6.

19.1.2 Related Work

Shared Tasks for Hypernym Detection

Hypernym discovery has been addressed before as a challenge, for example, at SemEval-2018 [64]. Unique to the FinSim task is the focus on the financial services industry. The evaluation campaign premiered in 2020 [118] and has been extended for the 2021 campaign, also referred to as FinSim-2 [332]: Two additional tags have been introduced and the training and evaluation datasets have been extended.

Knowledge Graphs

FinMatcher uses three external knowledge graphs as background knowledge for the task of hypernym detection.

WordNet [149] is a well-known lexical resource. It is a database of English words grouped in sets that represent a particular meaning, called *synsets*; further semantic relations such as hypernymy also exist in the database. The resource is publicly available.[2]

Wikidata is a knowledge graph hosted by the Wikimedia Foundation, which is publicly available[3] and maintained by an open community. The graph contains class-like entities, such as "stock market index", and also instance-like entities, such as "MSCI World". An example for a Wikidata statement would be *"MSCI World" instance of "stock market index"*[4]. The graph can be queried using SPARQL[5].

A frequent problem that occurs when working with external background knowledge in the financial services domain is the fact that less common entities – so-called *long tail entities* – are not contained within a knowledge base. The *WebIsA* [467] database is an attempt to tackle this problem by providing a dataset that is not based on a single source of knowledge – like *DBpedia* [300] – but instead on the whole Web: The dataset consists of hypernymy relations extracted from the *Common Crawl*[6], a freely downloadable crawl of a significant portion of the Web. For the automated extraction, lexico-syntactic patterns similar to those presented by Hearst [190] were used. Like Wikidata, the graph contains class-like and instance-like concepts. A sample triple from the dataset is *"zero-coupon bond" skos:broader "bond"*[7]. The dataset is also available via an LOD endpoint[8] under the name *WebIsALOD* [198] – hence, it can be queried like Wikidata using SPARQL.

Knowledge Graph Embeddings

In recent years, latent representations have gained traction not only in natural language processing but also in other data science communities. *RDF2vec* [442] is a knowledge graph embedding approach, which allows for obtaining a latent representation for the elements of a knowledge graph, i.e., a vector, for each node and each edge in a graph. It applies the *word2vec* [344, 345] model to RDF data: Random walks are performed for each node and are interpreted as sentences. After the walk generation, the sentences are used as input for the

[2] see `https://wordnet.princeton.edu/download`
[3] see `https://www.wikidata.org/wiki/Wikidata:Main_Page`
[4] see `https://www.wikidata.org/wiki/Q1881843`
[5] see `https://query.wikidata.org/`
[6] see `http://commoncrawl.org/`
[7] see `http://webisa.webdatacommons.org/concept/zero-coupon_bond_`
[8] see `http://webisa.webdatacommons.org/`

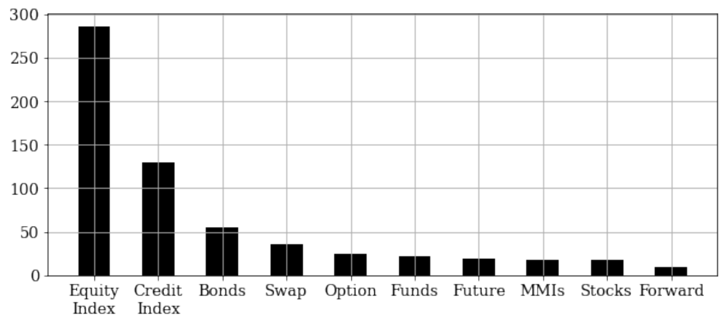


Figure 19.1: Distribution of Class Labels in the FinSim Training Dataset

word2vec algorithm. As a result, one obtains a vector for each word, i.e., a concept in the RDF graph. Multiple flavors of RDF2vec have been developed in the past, such as biased walks [84] or *RDF2vec Light* [405].[9] The calculation of knowledge graph embeddings on large graphs can require a significant amount of resources. Therefore, *KGvec2go*[10] [404] provides pre-trained RDF2vec knowledge graph embeddings through a Web API as well as via download. For the system presented in this section, a pre-trained embedding of WebIsALOD has been downloaded from KGvec2go.

Both RDF2vec and WebIsALOD have been used for integration tasks in the financial services domain before [351, 417].

19.1.3 FinSim Dataset Description

The FinSim dataset consists of 614 hyponym-hypernym pairs. There are 10 class labels (see Figure 19.1), i.e., hypernyms. The class labels classify concepts not according to their features but instead according to their prototypical kind. The distribution of class labels is not balanced. As shown in Figure 19.1, the distribution of labels follows a power-law with 286 entries for "equity index" and only nine entries for "forward". This is a challenging setting for multiple reasons: (i) the training dataset is comparatively small, (ii) the hypernyms are semantically very related, (iii) industry abbreviations are used, and (iv) there are textual overlaps. The FinSim-2 test dataset consists of 212 entries; the distribution of class labels is not known.

[9] For a good overview of the RDF2vec approach and its applications, refer to `http://www.rdf2vec.org/`

[10] see Chapter 8 and `http://www.kgvec2go.org`

Compared to other evaluation campaigns where participants have to submit their implementations, such as the Ontology Alignment Evaluation Initiative, participants of the FinSim task run their system on their own premises and submit the predictions made by their system.

19.1.4 System Description

The FinMatcher system combines explicit and latent features. In total, there are five groups of features which will be presented in the following. The overall architecture is shown in Figure 19.2.

Features

Word Overlap The overlap between hyponym and class label is a strong signal for a match. An example would be “Supranational Bond” which is a “Bond”. As such constellations are relatively frequent in the provided dataset, the first feature vector encodes whether the label contains the class label. For this feature, minimal text pre-processing is applied including lower-casing and removal of the plural suffix “s”. As this step is performed for each class label, a vector of length 10 is obtained. The overlap feature vector is displayed in green in Figure 19.2.

Wikidata Hypernym Lookup Wikidata is a large general-purpose knowledge graph, which is not tailored to the financial domain. Nonetheless, the data source contains many financial concepts and relations between them. For example, the concept “UCITS” can be linked to“Undertakings for Collective Investment in Transferable Securities” via the *also known as* label; due to the annotated relation *subclass of*, it is easily recognizable that “UCITS” is an “investment fund”.[11] This notion is exploited in this set of features: A comprehensive linking mechanism from the MELT framework[12] [203, 204] is used to link classes (the hypernyms) as well as labels (the hyponyms) to Wikidata concepts and then relations $P31$ (instance of) and $P279$ (subclass of) are followed up to two hops to evaluate whether the class label appears. Distant matches receive a lower signal strength which is calculated through the inverse hop-distance: A direct hypernym annotation (as in the UCITS example stated earlier) receives the value $\frac{1}{1} = 1$ whereas

[11] see `https://www.wikidata.org/wiki/Q25323628`

[12] The Matching EvaLuation Toolkit is a framework for ontology and instance matching (development, evaluation, visualization [400]). However, components can also be exploited for other tasks. For a better overview, see `https://github.com/dwslab/melt/` and Part II of this dissertation.

a two-hop match would receive a value of $\frac{1}{2} = 0.5$. As this step is performed for each class label, a vector of length 10 is obtained. The Wikidata lookup feature vector is displayed in blue in Figure 19.2.

WordNet Hypernym Lookup The same exploitation approach chosen for Wikidata is applied on the WordNet graph: Hypernyms and hyponyms are linked into WordNet, and then the inverse hop-distance is used as feature value. This is done for each class label that could be linked. The WordNet lookup feature vector is displayed in yellow in Figure 19.2.

WebIsALOD Hypernym Lookup In a similar fashion to the Wikidata hypernym lookup, class labels as well as hyponym labels are linked to the WebIsALOD graph using a linker from the MELT framework. In this graph, there exists only one significant relation: *skos:broader*. For each hyponym, the broader concepts are obtained, and it is checked whether the hypernym appears. Due to a high level of noise, the number of upwards hops is limited to 1. As this step is performed for each class label, a vector of length 10 is obtained. The WebIsALOD lookup feature vector is displayed in purple in Figure 19.2.

WebIsALOD RDF2vec Similarity For the embedding feature, each class label, as well as each hyponym label, is linked again into the WebIsALOD knowledge graph. Each concept in WebIsALOD has an associated embedding vector $v \in IR^{200}$. For comparisons, the cosine similarity between the hyponym and the class label is calculated.

If the whole concept cannot be linked, multiple sub-concepts are detected and linked. Within this linking process, longer sub-concepts are favored. For example, the string "CDX Emerging Markets" cannot be directly linked – however, the longest substring that can be linked here is "Emerging Markets"; in addition, "CDX" can also be linked. Comparisons in such cases are performed as follows:

$$\frac{\sum_{i=0}^{I} \max_{j=0}^{J}(sim(v_i, v_j))}{|I|} \tag{19.1}$$

where I represents the set of links of the hyponym, J represents the set of links of the hypernym, v_i and v_j correspond to the vectors of the links, and sim refers to a similarity function. In this case, the cosine is used as the similarity function. As this step is performed for each class label, a vector of length 10 is obtained. The WebIsALOD lookup feature vector is displayed in salmon in Figure 19.2.

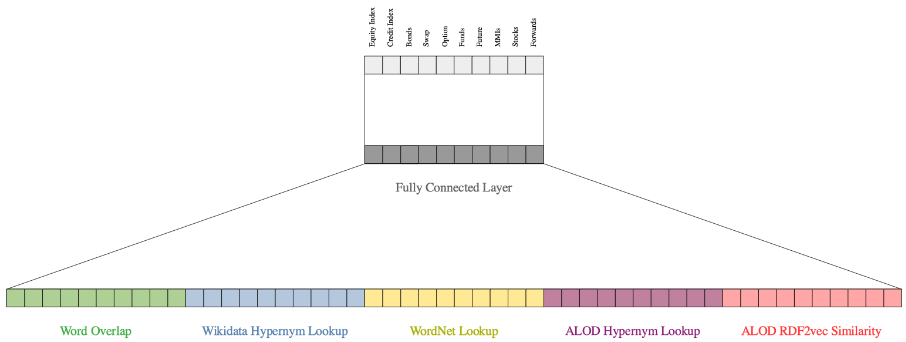


Figure 19.2: Architecture of FinMatcher

Feature Composition Each of the features i returns a signal vector $s_i \in \mathbb{R}^{10}$. All vectors are concatenated to form the final signal vector $S = ||_{i=1}^{5} s_i$, which is used as input for the classifier.

Classifier

Due to the small total number of training examples, a very simple artificial neural network architecture has been chosen. It is configured with one fully connected layer of size 10 and mean squared error as loss. The network was trained with 100 epochs and a batch size of 25 on a consumer PC. The vector that is to be predicted is of size 10 and represents the one-hot-encoded class label. The neural network classifier performed best among the classifiers evaluated: Naïve Bayes, J48 decision trees, random forests, and a regression.

As the distribution of class labels is skewed (see Figure 19.1), we applied the *synthetic minority oversampling technique* (SMOTE) [74] to upsample underrepresented class labels. We experimentally chose 33% of the majority class total as the upsampling barrier; this means that if the majority class in the training split totals 229 records, upsampling for class labels with less than $\frac{1}{3} * 229 = 76$ records will be performed so that there are 76 records for the underrepresented class label.

19.1.5 Results

Results of the Training Data and Ablation Study

We evaluated our matching system by performing a stratified fivefold cross-validation on the training data. We trained each *artificial neural network* (ANN) configuration 10 times and report the average results for accuracy and mean

Table 19.1: Ablation Study

Left-out Vector Group	Mean Accuracy	Mean Rank (HITS@10)
Submission	86.69	1.432
No SMOTE	85.55	1.371
Word Overlap	60.51	2.007
Wikidata Hypernyms	85.50	1.490
WordNet Hypernyms	86.66	1.440
ALOD Hypernyms	85.88	1.361
ALOD RDF2vec	85.56	1.481

Table 19.2: Absolute Weights per Feature Group

Vector Group	Weight
Word Overlap	13.64
Wikidata Hypernyms	13.01
WordNet Hypernyms	13.35
ALOD Hypernyms	9.12
ALOD RDF2vec	14.01

rank. We further performed an ablation study by training and evaluating the performance when leaving out each of the five feature groups. The results can be found in Table 19.1.

It is visible that the most important feature group in terms of accuracy is word overlap. This is not surprising given the high number of labels that contain the hypernym within their name (for example "green bonds" → "bonds") and shows that it is sensible for the task at hand to combine explicit and latent features. The observation that the inclusion of the target label in the term is a significant signal has also been made in the last FinSim campaign [118]. The negligible role of WordNet in terms of accuracy is also comprehensible since this particular external background knowledge dataset contains merely general-purpose class knowledge (such as "call option") but no knowledge about instances (such as "MSCI EMU Index"). For the FinSim dataset, very large knowledge graphs that contain class, as well as instance knowledge, are more beneficial due to their higher concept coverage. However, the information in the knowledge graphs used also contain some redundancy, as can be observed in Table 19.1: leaving out a single knowledge graph does not significantly change the results.

To further analyze the contribution of the different signals, we plotted the weights of the input features. As the weight of each input neuron s_i relates to

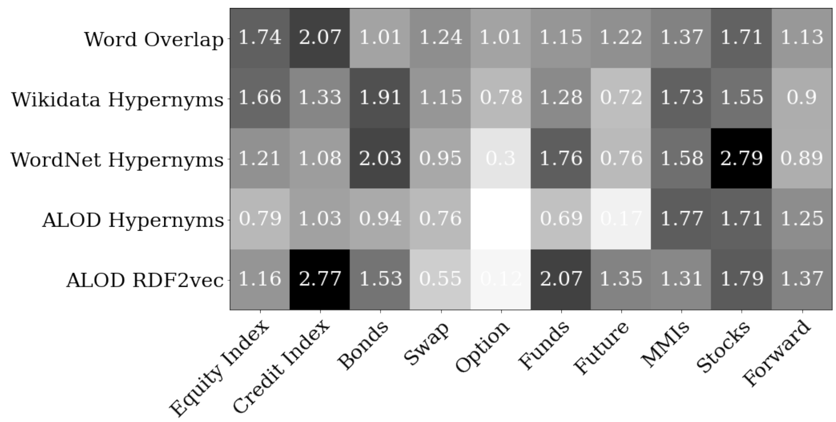


Figure 19.3: Heatmap of the Absolute Weights per Feature Group and Class Label

label i, we can directly observe which features the trained model considers relevant to identify which label.

Table 19.2 shows the summed absolute weight per feature group. This allows for analyzing the overall contribution of the individual feature group. Here, it is visible that the latent RDF2vec feature group receives the highest weight – higher than the word overlap group.

While the word overlap feature is important for the majority labels (equity index, credit index), it is not equally important for all labels and does not have the overall highest weight: Figure 19.3 shows the summed absolute weight per feature group and class label. The class labels are sorted in descending order by frequency. Here, it is visible that the word overlap has the highest contribution for the equity index as well as a high contribution for the credit index but low weights for the remaining minority classes.

Results Using the Reference Data

FinMatcher participated only with one configuration and achieved an accuracy of 81.1% and a mean rank of 1.415 on the reference data below the expected scores from the training data shown in Table 19.1.

19.1.6 Conclusion

In this section, we presented *FinMatcher*, a hypernym detection system for the financial services domain which exploits multiple knowledge graphs by combining explicit and latent features. We could show that the task can be addressed by including external knowledge in the form of knowledge graphs and that the combination of multiple graphs is overall beneficial.

19.2 SAP Use Case

This dissertation was sponsored by SAP SE[13]. Naturally, the need for semantic integration in businesses is substantial. Therefore, SAP decided to implement a pilot software product based on the findings of this dissertation. The project was initiated in 2020 under the name *Project "Bucharest"*. After a development phase, a pilot release was shipped to multiple customers in the financial services area in order to receive feedback and to test marketability. This section provides a short overview of the software pilot. Note that due to the limitation in space in this dissertation, the description of the tool is not complete (not every screen is shown). Consider further that implementation details cannot be fully revealed here since the intellectual property belongs to SAP SE.

19.2.1 Product Scope

The core product scope of project "Bucharest" is machine-assisted schema matching.[14] The tool, therefore, allows for managing two main data objects: *Schemas* and *Alignments*. These two objects are also shown on the entry screen in Figure 19.4.

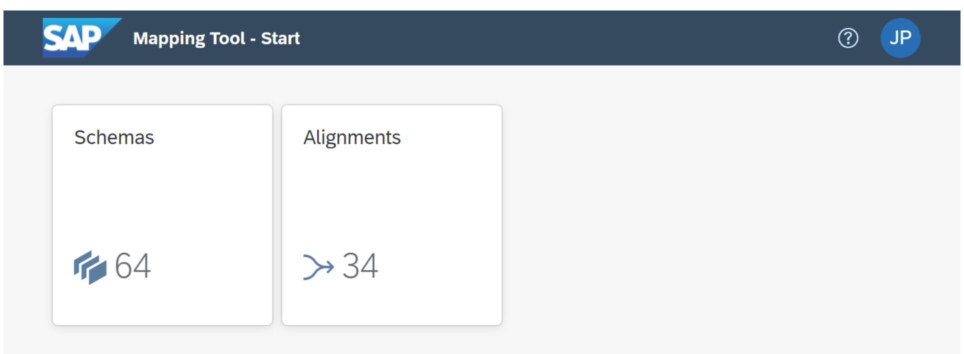


Figure 19.4: Mapping Tool Entry Screen on iPad Mini

[13]For a complete list of patents publications filed for SAP SE, see the list on page xviii.

[14]For brevity, the project "Bucharest" pilot software release is also referred to as *tool* in the following.

Business Requirements

Before designing the solution, business requirements were collected through multiple expert interviews and design thinking workshops. Table 19.3 lists the most important high-level requirements. Based on the collected requirements, the pilot was designed and implemented. The subsequent paragraphs refer to the corresponding requirements whereby the requirement ID is stated in brackets (*R* followed by the requirement number).

Schema Import (R1, R4)

Since enterprises use heterogeneous schema representation paradigms (e.g., controlled vocabularies, OWL ontologies, or entity-relationship models), a software solution needs to support multiple such representation paradigms.

Therefore, multiple importers are available in the tool, which translate the corresponding schemas upon import into a graph format without loss of semantic information. Examples of valid import formats are OWL, SAP Enterprise Architect Designer models, or SAP PowerDesigner models. Once at least two schemas are available within the tool, the match operation can be initiated. Each schema is versioned so that the schema life cycle can be fully represented in the tool.

Schema Display (R2)

All imported schemas can be explored in a *schema overview* screen (Figure 19.5) and multiple *schema detail* screens (Figure 19.6) which allow exploring the schema and individual schema elements. Since data modeling is out of the scope of the tool, there are no edit functions implemented.

Schema Matching (R3, R4, R5, R6)

Alignments can be directly imported (R3) or created (R5). For the creation of a new alignment, the source and target schemas (with versions) have to be provided together with a version code and the desired matching strategy (manual or automatic).

Once an alignment is created, its correspondences can be viewed. This overview screen is provided in Figure 19.7. The figure has been annotated for better explainability by defining functional areas in the user interface with assigned numbers. In area (1), the user can retrieve specific correspondences based on various search criteria. Area (2) provides the user with the option to directly remove an alignment or to edit it (see next paragraph). The user can also change

ID	Requirement	Description
R1	Schema Import	Schemas of various formats shall be importable. If required, additional formats shall be timely supported.
R2	Schema View	Once imported, schemas shall be explorable, including all details on individual schema elements.
R3	Alignment Import	Existing alignments in various formats shall be importable. If required, additional formats shall be timely supported.
R4	Versioning	The life cycle of schemas and alignments shall be represented in the tool.
R5	Alignment Creation	The user shall be able to create, edit, and delete alignments between any two schemas available in the tool.
R6	Alignment View	Once alignments are imported or created, they shall be explorable on correspondence level.
R7	Correspondence Smart Service	A smart service shall be in place to propose correspondence candidates to the user.
R8	Alignment Smart Service	A smart service shall be in place to propose an alignment to the user.
R9	Alignment Upgrade	If a schema changes, alignments shall also be easily upgradeable.
R10	Alignment Export	Alignments managed in the tool shall be exportable for external analysis, review, or for downstream systems. The export shall be available in various formats. If required, additional formats shall be timely supported.
R11	Schema Export	Schemas managed in the tool shall be exportable. The export shall be available in various formats. If required, additional formats shall be timely supported.

Table 19.3: High Level Requirements

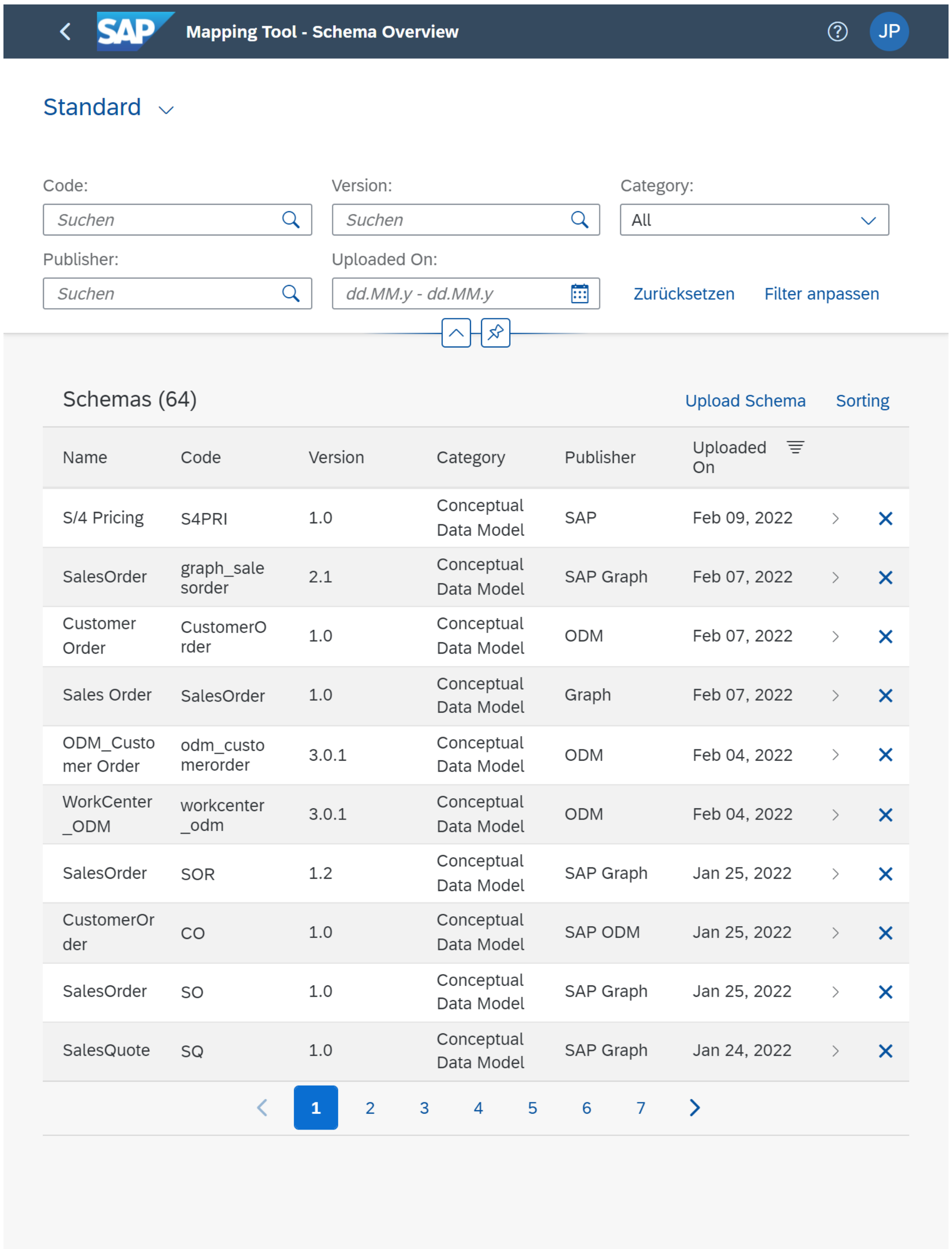


Figure 19.5: Mapping Tool Schema Overview Screen on iPadMini

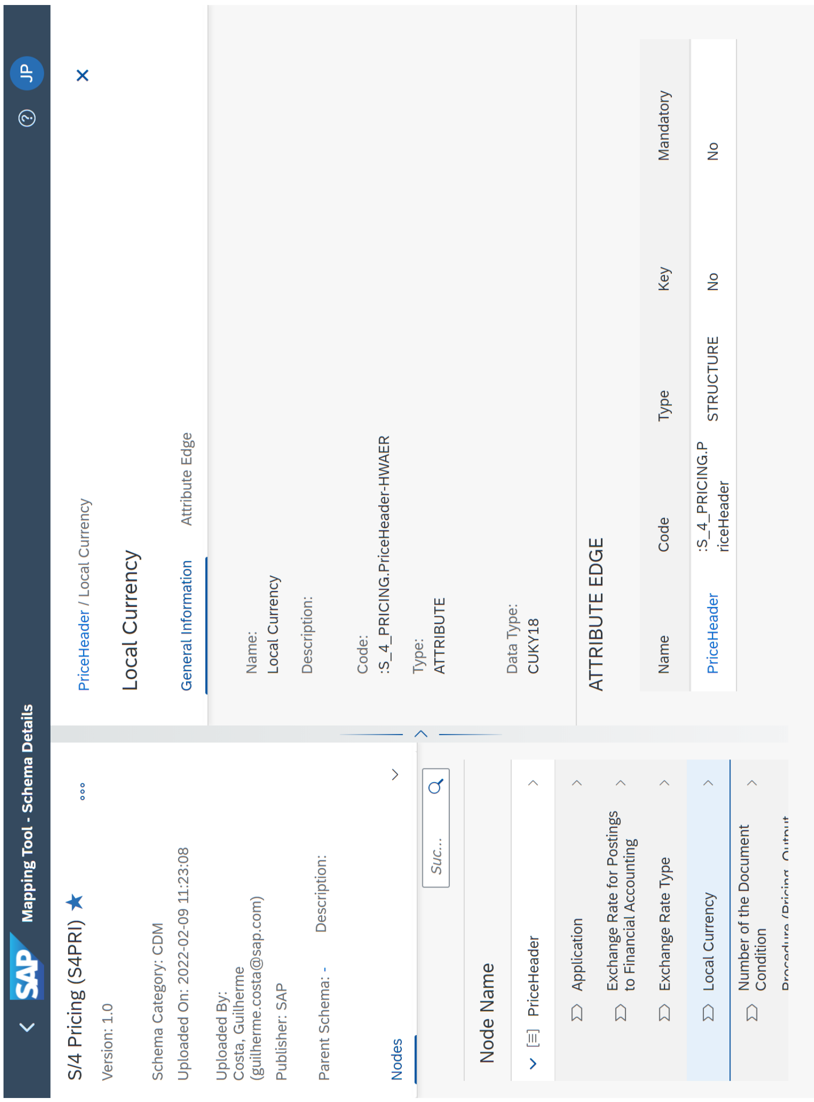


Figure 19.6: Schema detail screen on iPadMini (landscape mode); shown is the element `LocalCurrency` of schema `S/4 Pricing`.

the status of individual correspondences by clicking on the status buttons ∧ and ∨, which is shown in area (3). This reflects the business requirements to discuss correspondences in meetings in order to come to a final decision on whether a correspondence is correct or not. If the user is missing a correspondence, she can add one using the `Create` button. Lastly, the user can download the alignment for further tasks such as usage in a downstream *extract, transform, load* (ETL) system, data sharing, or analysis. Multiple (SAP-proprietary and public) formats are available such as the one proposed by the Alignment API [96].

Figure 19.8 shows the individual correspondence editing screen for manual matching. The figure has been annotated for better explainability by defining functional areas in the user interface with assigned numbers. Area (1) allows the user to pick the element from the target schema for which she wants to create a correspondence; in the example, she selected property *industry sector* of the *corporate account*. Once a selection is made, proposals are generated in area (2). A star rating provides the user with a visual indication of the algorithmic confidence. If one of the proposals is correct, the user can use the + button to add the element from the other schema to the alignment. If none of the proposals is correct or if the user wants to browse the schema, she can use the functionality provided in area (3). If source and target elements are selected, they will appear in the correspondence definition in area (4). The user can now either save the correspondence and return to the overview screen (`Save`), save the correspondence and stay on the screen in order to create a new correspondence (`Save and New`), or discard the existing correspondence and return to the overview screen – via buttons near annotation (5).

19.2.2 Alignment Smart Service (R7, R8)

The previous subsubsection already demonstrated the automated capabilities of the tool: A proposal engine is capable of generating match candidates which can be proposed on the level of individual correspondences (as shown in Figure 19.8); it is also possible to generate a complete alignment or an alignment where the top X matching candidates are already provided for each element in one schema.

The correspondence proposal engine combines various matching features and calculates confidence scores for match candidates. The implementation uses multiple external background knowledge resources such as corporate business thesauri. In addition, SAP licensed the usage of a large hypernymy knowledge graph of the University of Mannheim. The graph is exploited via RDF2vec embeddings.

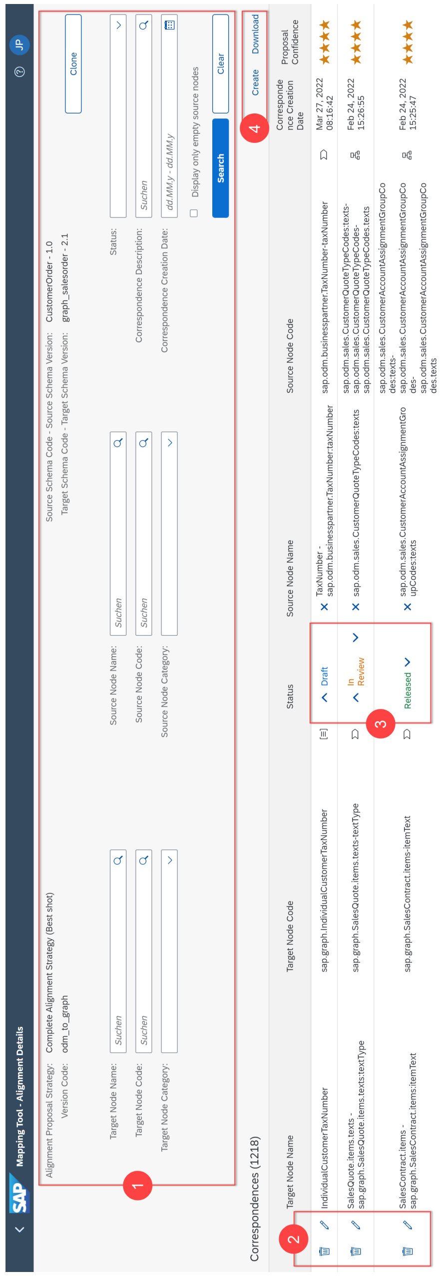


Figure 19.7: Correspondence Overview Screen. Annotations were added in red.

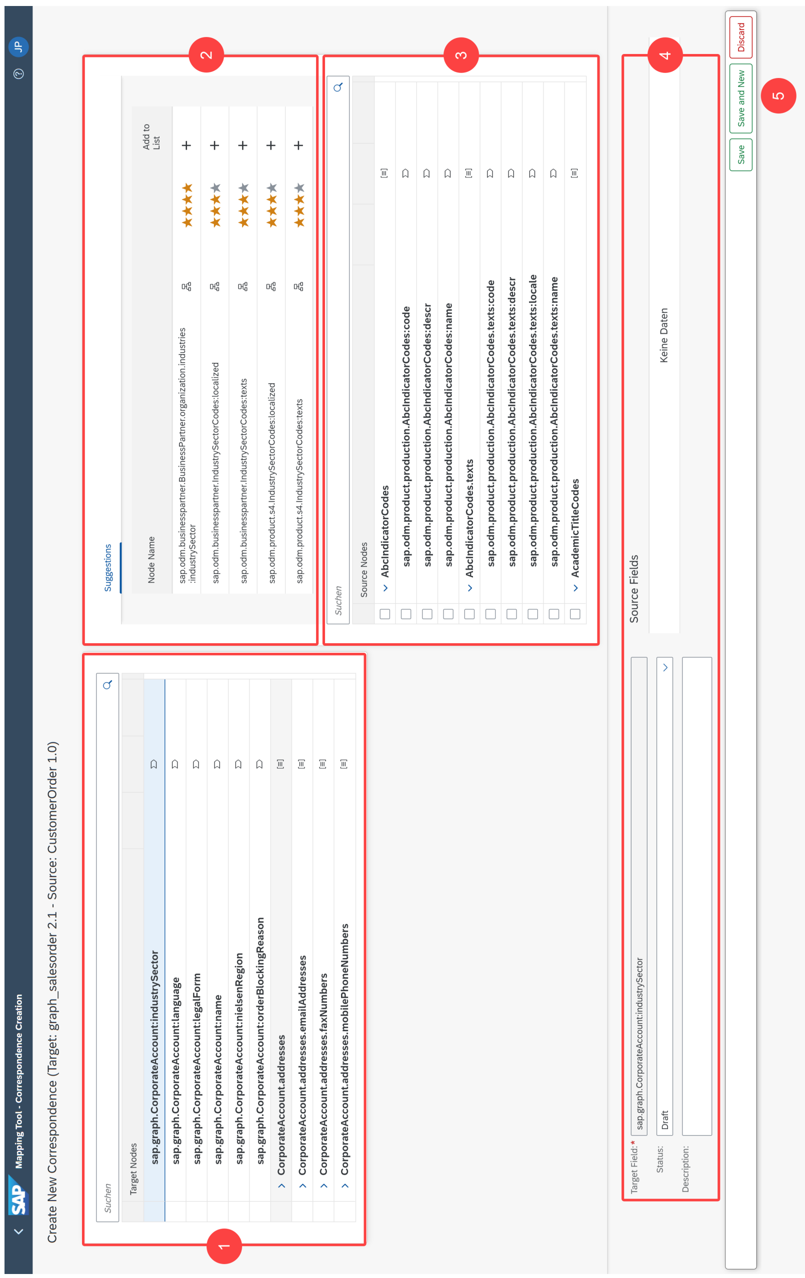


Figure 19.8: Correspondence Annotation Screen on a Desktop Computer. Annotations were added in red.

19.2.3 Alignment Upgrades (R9)

Since corporate schemas change regularly, it is very important to be able to quickly upgrade alignments to another schema version. This function is available through the *cloning* operation, which will create a new alignment with a new version using an updated schema. Non-conflicting correspondences are copied to the new alignment while the user is asked to resolve conflicts. The entry screen for this process is shown in Figure 19.9.

19.2.4 Implementation and Deployment

The tool is developed as a responsive cloud application in Java using SAP UI5 as a front-end framework. SAP HANA is used as a high-performance in-memory database for all data.

19.2.5 Future Developments

The presented pilot provides a first glimpse into a potential schema matching tool by SAP SE. SAP Project "Bucharest" is a pilot release without any service level agreements. SAP SE may decide to productize the release further or to integrate it into existing SAP solutions.

Interesting areas to discover further are an improved alignment proposal engine (e.g., based on multi-tenant data) and the integration with ETL tools, i.e., to bridge schema matching with data translation. The product team has already developed further concepts in this respect. Figure 19.10 (an iterative improvement to the screen shown in Figure 19.7), for instance, shows a UI mock, which incorporates, besides better visual elements, an overview of data flow mapping rules for the integration of the tool within an ETL integration landscape.

19.2.6 Conclusion

In this section, SAP Project "Bucharest", a schema matching prototype for businesses, was presented. The pilot release underlines the business value of this research and demonstrates what enterprise schema matching may look like in the future.

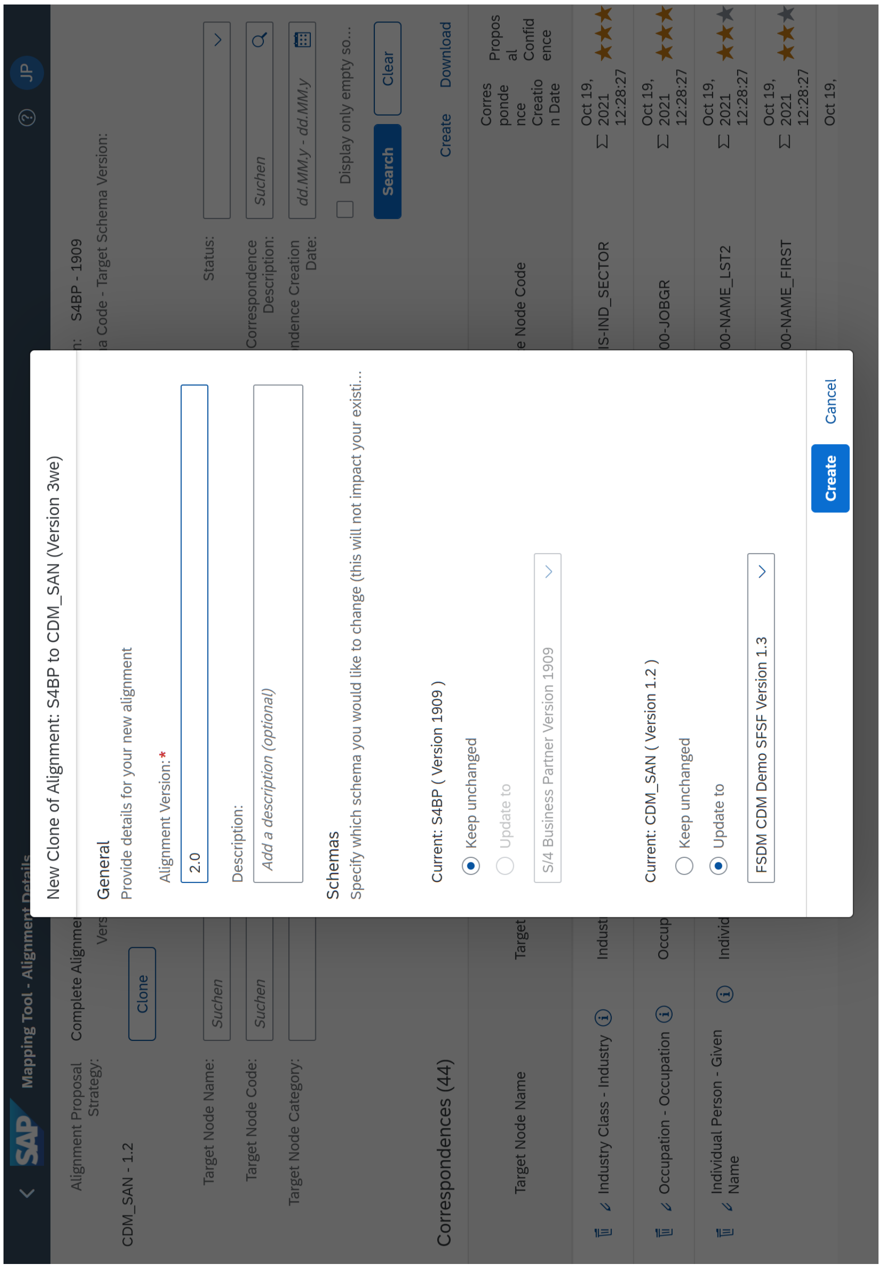


Figure 19.9: Alignment upgrade screen on iPad Air (landscape mode).

Figure 19.10: Future Matching UI

Part V

Outlook and Conclusion

Chapter 20

Thesis Conclusion

This chapter summarizes the previous parts of this thesis. The contributions are outlined and open issues are addressed together with future work.

20.1 Part I: Motivation and Foundation

In Part I, the frame for this dissertation was set. The core concepts were introduced. Via a literature-based approach, three classification systems were presented: (1) A classification for background knowledge sources in ontology matching, (2) a classification for linking approaches, and (3) a classification for exploitation strategies. Existing matching systems were classified in order to identify interesting research desiderata. It could be numerically shown that Semantic Web datasets are rarely used as background knowledge sources. It could further be shown that the most prevalent exploitation technique is rooted in factual queries and that logical, as well as statistical/neural techniques, are underexplored. Interest could be observed in the latter category, but the focus of neural methods is still mostly combined with textual resources or pre-trained language models. Multiple biases were identified; mentionable is a significant skew towards biomedical datasets and knowledge sources, extreme utilization of WordNet when it comes to general-purpose background knowledge, a focus on matching problems in the English language, and a focus on monolingual matching. Based on the findings of the survey, this dissertation addresses some white spots, namely the exploitation of general-purpose datasets with neural and explicit methods.

20.2 Part II: A Framework for Knowledge Graph Matching

In Part II, the Matching EvaLuation Toolkit was presented – a framework for developing, fine-tuning, evaluating, and packaging knowledge graph matching systems. This part of the dissertation answers RQ1, which was posed in the beginning (see Subsection 1.1 for all research questions).

MELT was developed to perform extensive evaluations required for this dissertation, such as comparisons of matching systems down to the level of correspondences, ablation studies, or significance tests. The framework provides simple, programming language independent APIs to develop matching modules. Over the course of this dissertation, MELT was gradually extended so that all main matching contributions are available to the research community. Since 2020, MELT has been officially endorsed by the OAEI. Over the short time frame of this dissertation, MELT has already experienced significant third-party usage. The MELT Dashboard allows for exploring alignments in an interactive way and is also used at the OAEI. The machine learning extension provides powerful tools for supervised ML in ontology matching. Each chapter of this part is accompanied by valuable analyses demonstrating the strengths of the framework. In Chapter 6, for instance, a feature-based matching system and an RDF2vec vector projection matching system are pioneered. The latter is later refined by introducing rotations in Chapter 14.

20.3 Part III: Knowledge Graph Embeddings

The third part of this dissertation focuses on knowledge graph embeddings and particularly strives to answer RQ2 and RQ3. Two task-oriented research strands could be identified: Data-mining-based approaches and knowledge-based-completion-oriented approaches. It could be shown that the models of both strands can be used for both purposes. RDF2vec mixes similarity and relatedness while link prediction approaches do not cover relatedness well. However, when it comes to finer-grained similarities, RDF2vec performs better. RDF2vec further performs better on noisy datasets.

With a focus on RDF2vec, further contributions were made. Multiple improvements were presented. The order-aware extension was introduced and performs generally better on data mining tasks and link prediction tasks compared to the classic variant of the embedding approach. Furthermore, deviations in the walk strategies were presented, which can be used to influence the representation of similarity and relatedness in the embedding space.

In order to better understand RDF2vec, its variants, and also knowledge graph embedding approaches in general, a gold standard rooted in description logic constructors was introduced. By providing a real and a synthetic dataset, the gold standard enables researchers to identify which constructors are learned by recognizing correlations and which constructors can actually be learned. An extensive evaluation of publicly available datasets and the gold standard developed in the scope of this thesis was carried out using twelve RDF2vec configurations together with multiple state of the art models.

Besides a comprehensive comparison of embedding approaches and multiple extensions to RDF2vec, the applicability of embeddings in downstream applications was also improved: With KGvec2go, a simple consumption framework was presented. RDF2vec Light can be used to embed very large knowledge graphs by focusing on the actual parts of interest. Both improvements can be directly used for schema matching and answer RQ2.

Chapter 14 closes this part and directly introduced first answers to the question of how we can combine embeddings and matching approaches.

20.4 Part IV: Background Knowledge in Knowledge Graph Matching

In Part IV, concrete instances of matching systems exploiting general-purpose background knowledge were presented addressing RQ5 from multiple angles: Wiktionary Matcher uses an explicit strategy on a large general-purpose multilingual dataset which was transformed into an RDF graph. The system is capable of matching monolingual and multilingual ontologies.

A comprehensive framework for incorporating transformers into the matching process was presented as well. Together with the technical foundation for language-model-based matching, multiple evaluations were carried out: First, zero-shot and fine-tuned models were evaluated for the task of matching. With KERMIT, a complete matching architecture based on bi- and cross-encoders was presented, which can be used with and without a reference alignment for fine-tuning. It could be shown that the overall approach is feasible, yields good results and that bi-encoders are very suitable for blocking.

ALOD2vec Matcher exploits RDF2vec knowledge graph embeddings of an automatically generated Semantic Web hypernymy dataset for matching ontologies and knowledge graphs.

In an extensive comparative study, six general-purpose knowledge graphs were evaluated on the task of schema matching using three different exploitation strategies. The study directly addresses RQ4. It could be shown that there is no superior knowledge source but that BabelNet performs comparatively well. It could further be shown that, statistically, varying the exploitation strategy has a greater effect than varying a (large) general-purpose knowledge source. Another interesting finding is that there is no clear superiority of expert-validated knowledge graphs over those created and validated by an open community.

The part is completed with a look at real-world applications for the findings of this dissertation. The FinMatcher system combines multiple knowledge sources and exploitation strategies to assign concepts from the financial industry to entities of the Financial Industry Business Ontology. Furthermore, a real-world use case of a corporate matching system at SAP SE was presented.

20.5 Open Issues and Limitations

Multiple chapters addressed specific open issues and limitations of the presented work. In this section, we summarize and present generally open issues and limitations of this dissertation.

While this dissertation contributes to a better understanding of knowledge graph embeddings, it is important to outline that this challenge is far from complete. In this thesis, we used multiple evaluation datasets and presented also novel ones. Nonetheless, we would like to mention that the set of benchmarks can still be extended.

This also applies to the matching tasks. The dissertation at hand builds mostly on the datasets provided by the OAEI. However, as discussed in more depth in Chapter 3, these datasets are subject to various biases: They are mostly monolingual, focused on biomedical tasks, and available only in English. Consequently, this dissertation also reflects these biases. In addition, the OAEI tracks are not sufficient for supervised approaches (a dedicated machine learning track may be helpful here).

Concerning the background knowledge exploitation strategies, this dissertation presents factual-query-based, embedding-based, and transformer-based approaches. However, it is particularly important to stress that the exploitation methods presented are not exhaustive and exclusive. We think that – particularly when it comes to knowledge graph embeddings and transformers – the approaches presented and evaluated are rather exemplary in nature and that many more interesting approaches can be developed and evaluated based on the findings in this dissertation. A concrete example would be the application of specialized knowledge graph embedding spaces for matching. The ALOD2vec matching system may, for instance, benefit from the application of e-RDF2vec$_{oa}$ for matching since the graph contains merely one edge type and since closer concepts are more relevant than distant ones when it comes to determining similarity.

Most approaches presented in this dissertation use one matching strategy exclusively. However, we think that the best results will be obtained by combining multiple sources and strategies. Although this will significantly complicate future ablation studies, hybrid systems are very interesting and are only superficially discussed in this thesis.

Background knowledge selection is not a core part of this dissertation. This reflects the broader research landscape. The adoption of automatic background knowledge selection is still in its infancy and another open issue to be addressed.

All linking operations to a background knowledge source in this dissertation are string-based. Improving linking approaches is yet another issue that may be addressed in the future.

With the exception of the application presented in Chapter 19, the user of the matching system is not discussed in this dissertation.

20.6 Future Work

We think that in the future, more task-specific evaluation datasets are required to judge the quality of embeddings for specific tasks other than link prediction. Furthermore, existing algorithms still need to be improved with a focus on scalability aspects (notably re-training and large-scale graphs), user explainability, and performance on dedicated downstream tasks (such as matching).

An interesting research direction is to consider more industry-specific matching datasets in evaluations such as corporate semantic models and corporate process models. While it is not easy to obtain real-world data, the analysis of such datasets will help to drive the application of semantic matching systems. An easier to achieve but yet underrepresented aspect is the provisioning of datasets for supervised matching and the broad evaluation and comparison of supervised matching algorithms.

While this dissertation presents multiple general-purpose background knowledge-based matching approaches – among which some use knowledge graph embeddings and transformers – we think that in the near future matching with transformers and knowledge graph embeddings will gain significantly more traction. Based on the findings of this dissertation, more involved matching systems can be investigated. This particularly includes matching systems that exploit one or more embedding spaces simultaneously, such as e-RDF2vec$_{oa}$ for candidate generation and p-RDF2vec$_{oa}$ for candidate selection.

Similarly, task-customized transformer models may provide an additional boost in performance for matching systems. Particularly interesting are, furthermore, hybrid systems, which would combine multiple knowledge sources and exploitation strategies and systems that are fine-tuned for specific matching tasks.

Chapter 3 gives a very good overview of broader future challenges that still need to be researched. As addressed in the previous section, automated background knowledge selection is still an interesting area to investigate. Similarly, more advanced background knowledge linking techniques may lead to better results for knowledge-based matching systems.

Lastly, we must not forget that while the ultimate goal of ontology and knowledge graph matching is on-demand, truly autonomously integrated data consumption, we are not there yet. As of today, humans need to interact with matching systems, understand the algorithmic output of matching systems, and correct them. This is particularly true for critical integration projects, e.g., in the business world. Therefore, sophisticated visualization techniques and user interfaces are required, together with extensive human interaction models.

Bibliography

[1] Abbas, N., Alghamdi, K., Alinam, M., Alloatti, F., Amaral, G.C.M., d'Amato, C., Asprino, L., Beno, M., Bensmann, F., Biswas, R., Cai, L., Capshaw, R., Carriero, V.A., Celino, I., Dadoun, A., Giorgis, S.D., Delva, H., Domingue, J., Dumontier, M., Emonet, V., van Erp, M., Espinoza-Arias, P., Fallatah, O., Ferrada, S., Ocaña, M.G., Georgiou, M., Gesese, G.A., Gillis-Webber, F., Giovannetti, F., Buey, M.G., Harrando, I., Heibi, I., Horta, V., Huber, L., Igne, F., Jaradeh, M.Y., Keshan, N., Koleva, A., Koteich, B., Kurniawan, K., Liu, M., Ma, C., Maas, L., Mansfield, M., Mariani, F., Marzi, E., Mesbah, S., Mistry, M., Tirado, A.C.M., Nguyen, A., Nguyen, V.B., Oelen, A., Pasqual, V., Paulheim, H., Polleres, A., Porena, M., Portisch, J., Presutti, V., Pustu-Iren, K., Mendez, A.R., Roshankish, S., Rudolph, S., Sack, H., Sakor, A., Salas, J., Schleider, T., Shi, M., Spinaci, G., Sun, C., Tietz, T., Dhouib, M.T., Umbrico, A., van den Berg, W., Xu, W.: Knowledge graphs evolution and preservation - A technical report from ISWS 2019. CoRR **abs/2012.11936** (2020), `https://arxiv.org/abs/2012.11936`

[2] Abdi, M.E., Souid, H., Kachroudi, M., Yahia, S.B.: CLONA results for OAEI 2015. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Proceedings of the 10th International Workshop on Ontology Matching collocated with the 14th International Semantic Web Conference (ISWC 2015), Bethlehem, PA, USA, October 12, 2015. CEUR Workshop Proceedings, vol. 1545, pp. 124–129. CEUR-WS.org (2015), `http://ceur-ws.org/Vol-1545/oaei15_paper2.pdf`

[3] Abolhassani, H., Hariri, B.B., Haeri, S.H.: On ontology alignment experiments. Webology **3**(3) (2006), `http://www.webology.org/2006/v3n3/a28.html`

[4] Acampora, G., Kaymak, U., Loia, V., Vitiello, A.: A fml-based fuzzy tuning for a memetic ontology alignment system. In: FUZZ-IEEE 2013, IEEE International Conference on Fuzzy Systems, Hyderabad, India, 7-10 July,

2013, Proceedings. pp. 1–8. IEEE (2013). `https://doi.org/10.1109/FUZZ-IEEE.2013.6622490`

[5] Acampora, G., Loia, V., Salerno, S., Vitiello, A.: A hybrid evolutionary approach for solving the ontology alignment problem. Int. J. Intell. Syst. **27**(3), 189–216 (2012). `https://doi.org/10.1002/int.20517`

[6] Acampora, G., Loia, V., Vitiello, A.: Enhancing ontology alignment through a memetic aggregation of similarity measures. Inf. Sci. **250**, 1–20 (2013). `https://doi.org/10.1016/j.ins.2013.06.052`

[7] Aguirre, J., Eckert, K., Euzenat, J., Ferrara, A., van Hage, W.R., Hollink, L., Meilicke, C., Nikolov, A., Ritze, D., Scharffe, F., Shvaiko, P., Sváb-Zamazal, O., dos Santos, C.T., Jiménez-Ruiz, E., Grau, B.C., Zapilko, B.: Results of the ontology alignment evaluation initiative 2012. In: Shvaiko, P., Euzenat, J., Kementsietsidis, A., Mao, M., Noy, N.F., Stuckenschmidt, H. (eds.) Proceedings of the 7th International Workshop on Ontology Matching, Boston, MA, USA, November 11, 2012. CEUR Workshop Proceedings, vol. 946. CEUR-WS.org (2012), `http://ceur-ws.org/Vol-946/oaei12_paper0.pdf`

[8] Alasoud, A., Haarslev, V., Shiri, N.: A multi-level matching algorithm for combining similarity measures in ontology integration. In: Collard, M. (ed.) Ontologies-Based Databases and Information Systems, First and Second VLDB Workshops, ODBIS 2005/2006 Trondheim, Norway, September 2-3, 2005, Seoul, Korea, September 11, 2006, Revised Papers. Lecture Notes in Computer Science, vol. 4623, pp. 1–17. Springer (2006). `https://doi.org/10.1007/978-3-540-75474-9_1`

[9] Alasoud, A., Haarslev, V., Shiri, N.: An effective ontology matching technique. In: An, A., Matwin, S., Ras, Z.W., Slezak, D. (eds.) Foundations of Intelligent Systems, 17th International Symposium, ISMIS 2008, Toronto, Canada, May 20-23, 2008, Proceedings. Lecture Notes in Computer Science, vol. 4994, pp. 585–590. Springer (2008). `https://doi.org/10.1007/978-3-540-68123-6_63`

[10] Aleksovski, Z., ten Kate, W., van Harmelen, F.: Exploiting the structure of background knowledge used in ontology matching. In: Shvaiko, P., Euzenat, J., Noy, N.F., Stuckenschmidt, H., Benjamins, V.R., Uschold, M. (eds.) Proceedings of the 1st International Workshop on Ontology Matching (OM-2006) Collocated with the 5th International Semantic Web Conference (ISWC-2006), Athens, Georgia, USA, November 5, 2006. CEUR

Workshop Proceedings, vol. 225. CEUR-WS.org (2006), http://ceur-ws.org/Vol-225/paper2.pdf

[11] Aleksovski, Z., Klein, M.C.A.: Ontology mapping using background knowledge. In: Clark, P., Schreiber, G. (eds.) Proceedings of the 3rd International Conference on Knowledge Capture (K-CAP 2005), October 2-5, 2005, Banff, Alberta, Canada. pp. 175–176. ACM (2005). https://doi.org/10.1145/1088622.1088655

[12] Aleksovski, Z., Klein, M.C.A., ten Kate, W., van Harmelen, F.: Matching unstructured vocabularies using a background ontology. In: Staab, S., Svátek, V. (eds.) Managing Knowledge in a World of Networks, 15th International Conference, EKAW 2006, Podebrady, Czech Republic, October 2-6, 2006, Proceedings. Lecture Notes in Computer Science, vol. 4248, pp. 182–197. Springer (2006). https://doi.org/10.1007/11891451_18

[13] Alexopoulos, P.: Semantic Modeling for Data. O'Reilly Media (2020)

[14] Algergawy, A., Cheatham, M., Faria, D., Ferrara, A., Fundulaki, I., Harrow, I., Hertling, S., Jiménez-Ruiz, E., Karam, N., Khiat, A., Lambrix, P., Li, H., Montanelli, S., Paulheim, H., Pesquita, C., Saveta, T., Schmidt, D., Shvaiko, P., Splendiani, A., Thiéblin, É., Trojahn, C., Vatascinová, J., Zamazal, O., Zhou, L.: Results of the ontology alignment evaluation initiative 2018. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018. CEUR Workshop Proceedings, vol. 2288, pp. 76–116. CEUR-WS.org (2018), http://ceur-ws.org/Vol-2288/oaei18_paper0.pdf

[15] Ali, M., Berrendorf, M., Hoyt, C.T., Vermue, L., Sharifzadeh, S., Tresp, V., Lehmann, J.: Pykeen 1.0: A python library for training and evaluating knowledge graph embeddings. J. Mach. Learn. Res. **22**, 82:1–82:6 (2021), http://jmlr.org/papers/v22/20-825.html

[16] Allemang, D., Hendler, J.A.: Semantic web for the working ontologist - modeling in RDF, RDFS and OWL. Elsevier (2008)

[17] Alshargi, F., Shekarpour, S., Soru, T., Sheth, A.P.: Metrics for evaluating quality of embeddings for ontological concepts. In: Martin, A., Hinkelmann, K., Gerber, A., Lenat, D., van Harmelen, F., Clark, P. (eds.) Proceedings of the AAAI 2019 Spring Symposium on Combining Machine

Learning with Knowledge Engineering (AAAI-MAKE 2019) Stanford University, Palo Alto, California, USA, March 25-27, 2019., Stanford University, Palo Alto, California, USA, March 25-27, 2019. CEUR Workshop Proceedings, vol. 2350. CEUR-WS.org (2019), http://ceur-ws.org/Vol-2350/paper26.pdf

[18] Alwan, A.A., Nordin, A., Alzeber, M., Abualkishik, A.Z.: A survey of schema matching research using database schemas and instances. International Journal of Advanced Computer Science and Applications **8**(10) (2017)

[19] Amin, M.B., Ahmad, M., Khan, W.A., Lee, S.: Biomedical ontology matching as a service. In: Bodine, C., Helal, S., Gu, T., Mokhtari, M. (eds.) Smart Homes and Health Telematics - 12th International Conference, ICOST 2014, Denver, CO, USA, June 25-27, 2014, Revised Papers. Lecture Notes in Computer Science, vol. 8456, pp. 195–203. Springer (2014). https://doi.org/10.1007/978-3-319-14424-5_21

[20] Anam, S., Kim, Y.S., Kang, B.H., Liu, Q.: Review of ontology matching approaches and challenges. International journal of Computer Science and Network Solutions **3**(3), 1–27 (2015)

[21] Anam, S., Kim, Y.S., Kang, B.H., Liu, Q.: Adapting a knowledge-based schema matching system for ontology mapping. In: Proceedings of the Australasian Computer Science Week Multiconference, Canberra, Australia, February 2-5, 2016. p. 27. ACM (2016). https://doi.org/10.1145/2843043.2843048

[22] Annane, A., Bellahsene, Z., Azouaou, F., Jonquet, C.: Selection and combination of heterogeneous mappings to enhance biomedical ontology matching. In: Blomqvist, E., Ciancarini, P., Poggi, F., Vitali, F. (eds.) Knowledge Engineering and Knowledge Management - 20th International Conference, EKAW 2016, Bologna, Italy, November 19-23, 2016, Proceedings. Lecture Notes in Computer Science, vol. 10024, pp. 19–33 (2016). https://doi.org/10.1007/978-3-319-49004-5_2

[23] Annane, A., Bellahsene, Z., Azouaou, F., Jonquet, C.: Building an effective and efficient background knowledge resource to enhance ontology matching. J. Web Semant. **51**, 51–68 (2018). https://doi.org/10.1016/j.websem.2018.04.001

[24] Arch-int, N., Arch-int, S.: Semantic ontology mapping for interoperability of learning resource systems using a rule-based reasoning approach. Ex-

pert Syst. Appl. **40**(18), 7428–7443 (2013). https://doi.org/10.1016/j.eswa.2013.07.027

[25] Ardjani, F., Bouchiha, D., Malki, M.: Ontology-alignment techniques: Survey and analysis. International Journal of Modern Education & Computer Science **7**(11) (2015). https://doi.org/10.5815/ijmecs.2015.11.08

[26] Ashpole, B., Ehrig, M., Euzenat, J., Stuckenschmidt, H. (eds.): Integrating Ontologies '05, Proceedings of the K-CAP 2005 Workshop on Integrating Ontologies, Banff, Canada, October 2, 2005, CEUR Workshop Proceedings, vol. 156. CEUR-WS.org (2005), http://ceur-ws.org/Vol-156

[27] Aumueller, D., Do, H.H., Massmann, S., Rahm, E.: Schema and ontology matching with COMA++. In: Özcan, F. (ed.) Proceedings of the ACM SIGMOD International Conference on Management of Data, Baltimore, Maryland, USA, June 14-16, 2005. pp. 906–908. ACM (2005). https://doi.org/10.1145/1066157.1066283

[28] Barker, R.: CASE method - entity relationship modelling. Computer aided systems engineering, Addison-Wesley (1990)

[29] Basu, S., Chakraborty, S., Hassan, A., Siddique, S., Anand, A.: ERLKG: entity representation learning and knowledge graph based association analysis of COVID-19 through mining of unstructured biomedical corpora. In: Chandrasekaran, M.K., de Waard, A., Feigenblat, G., Freitag, D., Ghosal, T., Hovy, E.H., Knoth, P., Konopnicki, D., Mayr, P., Patton, R.M., Shmueli-Scheuer, M. (eds.) Proceedings of the First Workshop on Scholarly Document Processing, SDP@EMNLP 2020, Online, November 19, 2020. pp. 127–137. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.sdp-1.15

[30] Bella, G., Giunchiglia, F., McNeill, F.: Language and domain aware lightweight ontology matching. J. Web Semant. **43**, 1–17 (2017). https://doi.org/10.1016/j.websem.2017.03.003

[31] Bellahsene, Z., Bonifati, A., Rahm, E. (eds.): Schema Matching and Mapping. Data-Centric Systems and Applications, Springer (2011). https://doi.org/10.1007/978-3-642-16518-4

[32] Bellahsene, Z., Bonifati, A., Rahm, E. (eds.): Schema Matching and Mapping. Data-Centric Systems and Applications, Springer (2011). https://doi.org/10.1007/978-3-642-16518-4

[33] Benaissa, M., Khiat, A.: A new approach for combining the similarity values in ontology alignment. In: Amine, A., Bellatreche, L., Elberrichi, Z., Neuhold, E.J., Wrembel, R. (eds.) Computer Science and Its Applications - 5th IFIP TC 5 International Conference, CIIA 2015, Saida, Algeria, May 20-21, 2015, Proceedings. IFIP Advances in Information and Communication Technology, vol. 456, pp. 343–354. Springer (2015). `https://doi.org/10.1007/978-3-319-19578-0_28`

[34] Benerecetti, M., Bouquet, P., Ghidini, C.: Contextual reasoning distilled. J. Exp. Theor. Artif. Intell. **12**(3), 279–305 (2000). `https://doi.org/10.1080/09528130050111446`

[35] Berners-Lee, T.: Linked Data - Design Issues (2006), `https://www.w3.org/DesignIssues/LinkedData.html`, accessed: 2022-05-10

[36] Berners-Lee, T.: Semantic Web and Linked Data (2009), `https://www.w3.org/2009/Talks/0120-campus-party-tbl/#(1)`, accessed: 2022-05-10

[37] Biniz, M., El Ayachi, R., Fakir, M.: Ontology matching using babelnet dictionary and word sense disambiguation algorithms. Indonesian Journal of Electrical Engineering and Computer Science **5**(1), 196–205 (2017), `https://doi.org/10.11591/ijeecs.v5.i1.pp196-205`

[38] Biniz, M., Fakir, M.: An ontology alignment hybrid method based on decision rules. Int. Arab J. Inf. Technol. **16**(6), 1114–1120 (2019), `https://iajit.org/index.php?option=com_content&task=blogcategory&id=143&Itemid=483`

[39] Biswas, R., Portisch, J., Alam, M., Sack, H., Paulheim, H.: Entity type prediction leveraging graph walks and entity descriptions. In: International Semantic Web Conference (2022), under review

[40] Bloem, P., Wilcke, X., van Berkel, L., de Boer, V.: kgbench: A collection of knowledge graph datasets for evaluating relational and multimodal machine learning. In: Verborgh, R., Hose, K., Paulheim, H., Champin, P., Maleshkova, M., Corcho, Ó., Ristoski, P., Alam, M. (eds.) The Semantic Web - 18th International Conference, ESWC 2021, Virtual Event, June 6-10, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12731, pp. 614–630. Springer (2021). `https://doi.org/10.1007/978-3-030-77385-4_37`

[41] Bock, J., Hettenhausen, J.: Mappso results for OAEI 2008. In: Shvaiko, P., Euzenat, J., Giunchiglia, F., Stuckenschmidt, H. (eds.) Proceedings of the 3rd International Workshop on Ontology Matching (OM-2008) Collocated with the 7th International Semantic Web Conference (ISWC-2008), Karlsruhe, Germany, October 26, 2008. CEUR Workshop Proceedings, vol. 431. CEUR-WS.org (2008), http://ceur-ws.org/Vol-431/oaei08_paper8.pdf

[42] Bodenreider, O., Hayamizu, T.F., Ringwald, M., de Coronado, S., Zhang, S.: Of mice and men: Aligning mouse and human anatomies. In: AMIA 2005, American Medical Informatics Association Annual Symposium, Washington, DC, USA, October 22-26, 2005. AMIA (2005), https://knowledge.amia.org/amia-55142-a2005a-1.613296/t-001-1.616182/f-001-1.616183/a-012-1.616655/a-013-1.616652

[43] Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Trans. Assoc. Comput. Linguistics **5**, 135–146 (2017), https://transacl.org/ojs/index.php/tacl/article/view/999

[44] Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Burges, C.J.C., Bottou, L., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States. pp. 2787–2795 (2013), https://proceedings.neurips.cc/paper/2013/hash/1cecc7a77928ca8133fa24680a88d2f9-Abstract.html

[45] Bouquet, P., Ehrig, M., Euzenat, J., Franconi, E., Hitzler, P., Krotzsch, M., Serafini, L., Stamu, G., Sure, Y., Tessaris, S.: Specification of a Common Framework for Characterizing Alignment. Deliverable D2.2.1 (2005)

[46] Bouzeghoub, A., Elbyed, A., Tahi, F.: OMIE: ontology mapping within an interactive and extensible environment. In: Bairoch, A., Boulakia, S.C., Froidevaux, C. (eds.) Data Integration in the Life Sciences, 5th International Workshop, DILS 2008, Evry, France, June 25-27, 2008. Proceedings. Lecture Notes in Computer Science, vol. 5109, pp. 161–168. Springer (2008). https://doi.org/10.1007/978-3-540-69828-9_16

[47] Breit, A., Waltersdorfer, L., Ekaputra, F., Sabou, M., Ekelhart, A., Iana, A., Paulheim, H., Portisch, J., Revenko, A., Teije, A.T., van Harmelen, F.: Com-

bining machine learning and semantic web: A systematic mapping study. ACM Computing Surveys (2022), under review

[48] Brickley, D., Miller, L.: FOAF Vocabulary Specification (2014), `http://xmlns.com/foaf/spec/20140114.html`, accessed: 2022-05-10

[49] Brown, K., Miller, J.: The Cambridge Dictionary of Linguistics, p. 431. Cambridge University Press, Cambridge; New York (2013)

[50] Brown, R.: Integration definition for information modelling (idef1x). National Institute of Standards and Technology (NIST), viewed Jan **24**, 2011 (1993), `https://web.archive.org/web/20131203223034/http://www.itl.nist.gov/fipspubs/idef1x.doc`

[51] de Bruijn, J., Ehrig, M., Feier, C., Martín-Recuerda, F., Scharffe, F., Weiten, M.: Ontology Mediation, Merging, and Aligning. In: Davies, J., Studer, R., Warren, P. (eds.) Semantic Web Technologies: Trends and Research in Ontology-Based Systems, pp. 95–113. Wiley and Sons, Chichester, England (2006)

[52] de Bruijn, J., Foxvog, D., Zimmerman, K.: D4.3.1 Ontology Mediation Patterns Library V1 (2005), `https://www.sti-innsbruck.at/sites/default/files/D4.3.1.pdf`, accessed: 2022-05-10

[53] Bruni, E., Boleda, G., Baroni, M., Tran, N.: Distributional semantics in technicolor. In: The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 1: Long Papers. pp. 136–145. The Association for Computer Linguistics (2012), `https://aclanthology.org/P12-1015/`

[54] Brunner, U., Stockinger, K.: Entity matching with transformer architectures - A step forward in data integration. In: Bonifati, A., Zhou, Y., Salles, M.A.V., Böhm, A., Olteanu, D., Fletcher, G.H.L., Khan, A., Yang, B. (eds.) Proceedings of the 23rd International Conference on Extending Database Technology, EDBT 2020, Copenhagen, Denmark, March 30 - April 02, 2020. pp. 463–473. OpenProceedings.org (2020). `https://doi.org/10.5441/002/edbt.2020.58`

[55] Budanitsky, A., Hirst, G.: Evaluating wordnet-based measures of lexical semantic relatedness. Comput. Linguistics **32**(1), 13–47 (2006). `https://doi.org/10.1162/coli.2006.32.1.13`

[56] Bulygin, L.: Combining lexical and semantic similarity measures with machine learning approach for ontology matching problem. In: Kalinichenko, L.A., Manolopoulos, Y., Stupnikov, S.A., Skvortsov, N.A., Sukhomlin, V. (eds.) Selected Papers of the XX International Conference on Data Analytics and Management in Data Intensive Domains (DAMDID/RCDL 2018), Moscow, Russia, October 9-12, 2018. CEUR Workshop Proceedings, vol. 2277, pp. 245–249. CEUR-WS.org (2018), `http://ceur-ws.org/Vol-2277/paper42.pdf`

[57] Bulygin, L., Stupnikov, S.A.: Applying of machine learning techniques to combine string-based, language-based and structure-based similarity measures for ontology matching. In: Elizarov, A.M., Novikov, B., Stupnikov, S.A. (eds.) Selected Papers of the XXI International Conference on Data Analytics and Management in Data Intensive Domains (DAMDID/RCDL 2019), Kazan, Russia, October 15-18, 2019. CEUR Workshop Proceedings, vol. 2523, pp. 129–147. CEUR-WS.org (2019), `http://ceur-ws.org/Vol-2523/paper14.pdf`

[58] Bunge, M., Mahner, M.: Über die Natur der Dinge: Materialismus und Wissenschaft, chap. 1, pp. 1–16. S. Hirzel Verlag, Stuttgart (2004)

[59] Bunge, M., Mahner, M.: Über die Natur der Dinge: Materialismus und Wissenschaft. S. Hirzel Verlag, Stuttgart (2004)

[60] Busch, A., Stenschke, O.: Germanistische Linguistik: Eine Einführung. Narr Francke Attempto, 4 edn. (2018)

[61] Busse, D.: Semantik. No. 3280 in UTB Sprachwissenschaft, Fink, Paderborn (2009)

[62] Busse, D.: Semantik, chap. 6, pp. 102–110. No. 3280 in UTB Sprachwissenschaft, Fink, Paderborn (2009)

[63] Cai, H., Zheng, V.W., Chang, K.C.: A comprehensive survey of graph embedding: Problems, techniques, and applications. IEEE Trans. Knowl. Data Eng. **30**(9), 1616–1637 (2018). `https://doi.org/10.1109/TKDE.2018.2807452`

[64] Camacho-Collados, J., Bovi, C.D., Anke, L.E., Oramas, S., Pasini, T., Santus, E., Shwartz, V., Navigli, R., Saggion, H.: Semeval-2018 task 9: Hypernym discovery. In: Apidianaki, M., Mohammad, S.M., May, J., Shutova, E., Bethard, S., Carpuat, M. (eds.) Proceedings of The 12th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2018,

New Orleans, Louisiana, USA, June 5-6, 2018. pp. 712–724. Association for Computational Linguistics (2018). https://doi.org/10.18653/v1/s18-1115

[65] Cardoso, H.L., Teixeira, D.D., Oliveira, E.: Enhancing interoperability: Ontology-mapping in an electronic institution. In: Fischer, K., Müller, J.P., Odell, J., Berre, A. (eds.) Agent-Based Technologies and Applications for Enterprise Interoperability - International Workshops, ATOP@AAMAS 2005, Utrecht, The Netherlands, July 25-26, 2005, and ATOP@AAMAS 2008, Estoril, Portugal, May 12-13, 2008, Revised Selected Papers. Lecture Notes in Business Information Processing, vol. 25, pp. 47–62. Springer (2008). https://doi.org/10.1007/978-3-642-01668-4_3

[66] Carroll, J.J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., Wilkinson, K.: Jena: implementing the semantic web recommendations. In: Feldman, S.I., Uretsky, M., Najork, M., Wills, C.E. (eds.) Proceedings of the 13th international conference on World Wide Web - Alternate Track Papers & Posters, WWW 2004, New York, NY, USA, May 17-20, 2004. pp. 74–83. ACM (2004). https://doi.org/10.1145/1013367.1013381

[67] Castano, S., Ferrara, A., Messa, G.: Results of the hmatch ontology matchmaker in OAEI 2006. In: Shvaiko, P., Euzenat, J., Noy, N.F., Stuckenschmidt, H., Benjamins, V.R., Uschold, M. (eds.) Proceedings of the 1st International Workshop on Ontology Matching (OM-2006) Collocated with the 5th International Semantic Web Conference (ISWC-2006), Athens, Georgia, USA, November 5, 2006. CEUR Workshop Proceedings, vol. 225. CEUR-WS.org (2006), http://ceur-ws.org/Vol-225/paper12.pdf

[68] Celebi, R., Uyar, H., Yasar, E., Gümüs, Ö., Dikenelli, O., Dumontier, M.: Evaluation of knowledge graph embedding approaches for drug-drug interaction prediction in realistic settings. BMC Bioinform. **20**(1), 726 (2019). https://doi.org/10.1186/s12859-019-3284-5

[69] Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., John, R.S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Sung, Y., Strope, B., Kurzweil, R.: Universal sentence encoder. CoRR **abs/1803.11175** (2018), http://arxiv.org/abs/1803.11175

[70] Chaker, J., Khaldi, M., Aammou, S.: Enhancing the interoperability of multimedia learning objects based on the ontology mapping. iJET **9**(5), 40–44 (2014), https://www.online-journals.org/index.php/i-jet/article/view/3934

[71] Chakraborty, J., Bansal, S.K., Virgili, L., Konar, K., Yaman, B.: Ontoconnect: unsupervised ontology alignment with recursive neural network. In: Hung, C., Hong, J., Bechini, A., Song, E. (eds.) SAC '21: The 36th ACM/SIGAPP Symposium on Applied Computing, Virtual Event, Republic of Korea, March 22-26, 2021. pp. 1874–1882. ACM (2021). `https://doi.org/10.1145/3412841.3442059`

[72] Chakraborty, J., Zahera, H.M., Sherif, M.A., Bansal, S.K.: ONTOCONNECT: domain-agnostic ontology alignment using graph embedding with negative sampling. In: Wani, M.A., Sethi, I.K., Shi, W., Qu, G., Raicu, D.S., Jin, R. (eds.) 20th IEEE International Conference on Machine Learning and Applications, ICMLA 2021, Pasadena, CA, USA, December 13-16, 2021. pp. 942–945. IEEE (2021). `https://doi.org/10.1109/ICMLA52953.2021.00155`

[73] Chang, S., Colace, F., Santo, M.D., Zegarra, E., Qie, Y.: MAMA: A novel approach to ontology mapping. In: Proceedings of the 17th International Conference on Distributed Multimedia Systems, DMS 2011, October 18-20, 2011, Convitto della Calza, Florence, Italy. pp. 34–41. Knowledge Systems Institute (2011)

[74] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002). `https://doi.org/10.1613/jair.953`

[75] Cheatham, M.: Targeted ontology mapping. In: Smari, W.W., McQuay, W.K. (eds.) 2010 International Symposium on Collaborative Technologies and Systems, CTS 2010, Chicago, Illinois, USA, May 17-21, 2010. pp. 123–132. IEEE (2010). `https://doi.org/10.1109/CTS.2010.5478518`

[76] Cheatham, M., Hitzler, P.: Stringsauto and mapsss results for OAEI 2013. In: Shvaiko, P., Euzenat, J., Srinivas, K., Mao, M., Jiménez-Ruiz, E. (eds.) Proceedings of the 8th International Workshop on Ontology Matching co-located with the 12th International Semantic Web Conference (ISWC 2013), Sydney, Australia, October 21, 2013. CEUR Workshop Proceedings, vol. 1111, pp. 146–152. CEUR-WS.org (2013), `http://ceur-ws.org/Vol-1111/oaei13_paper7.pdf`

[77] Cheatham, M., Hitzler, P.: Conference v2.0: An uncertain version of the OAEI conference benchmark. In: Mika, P., Tudorache, T., Bernstein, A., Welty, C., Knoblock, C.A., Vrandecic, D., Groth, P., Noy, N.F., Janowicz, K.,

Goble, C.A. (eds.) The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part II. Lecture Notes in Computer Science, vol. 8797, pp. 33–48. Springer (2014). https://doi.org/10.1007/978-3-319-11915-1_3

[78] Chen, B., Tan, H., Lambrix, P.: Structure-based filtering for ontology alignment. In: 15th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE 2006), 26-28 June 2006, Manchester, United Kingdom. pp. 364–369. IEEE Computer Society (2006). https://doi.org/10.1109/WETICE.2006.64

[79] Chen, G., Zhang, S.: Fcamapx results for OAEI 2018. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018. CEUR Workshop Proceedings, vol. 2288, pp. 160–166. CEUR-WS.org (2018), http://ceur-ws.org/Vol-2288/oaei18_paper7.pdf

[80] Chen, J., Chen, X., Horrocks, I., Myklebust, E.B., Jiménez-Ruiz, E.: Correcting knowledge base assertions. In: Huang, Y., King, I., Liu, T., van Steen, M. (eds.) WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020. pp. 1537–1547. ACM / IW3C2 (2020). https://doi.org/10.1145/3366423.3380226

[81] Chen, X., Xia, W., Jiménez-Ruiz, E., Cross, V.V.: Extending an ontology alignment system with bioportal: a preliminary analysis. In: Horridge, M., Rospocher, M., van Ossenbruggen, J. (eds.) Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web Conference, ISWC 2014, Riva del Garda, Italy, October 21, 2014. CEUR Workshop Proceedings, vol. 1272, pp. 313–316. CEUR-WS.org (2014), http://ceur-ws.org/Vol-1272/paper_67.pdf

[82] Christen, P.: Data Matching - Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection, chap. 6, pp. 129–162. Data-Centric Systems and Applications, Springer (2012). https://doi.org/10.1007/978-3-642-31164-2

[83] Chua, W.W.K., Kim, J.: Eff2match results for OAEI 2010. In: Shvaiko, P., Euzenat, J., Giunchiglia, F., Stuckenschmidt, H., Mao, M., Cruz, I.F. (eds.) Proceedings of the 5th International Workshop on Ontology Matching (OM-2010), Shanghai, China, November 7, 2010. CEUR Workshop Proceed-

ings, vol. 689. CEUR-WS.org (2010), http://ceur-ws.org/Vol-689/oaei10_paper5.pdf

[84] Cochez, M., Ristoski, P., Ponzetto, S.P., Paulheim, H.: Biased graph walks for RDF graph embeddings. In: Akerkar, R., Cuzzocrea, A., Cao, J., Hacid, M. (eds.) Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics, WIMS 2017, Amantea, Italy, June 19-22, 2017. pp. 21:1–21:12. ACM (2017). https://doi.org/10.1145/3102254.3102279

[85] Cochez, M., Ristoski, P., Ponzetto, S.P., Paulheim, H.: Global RDF vector space embeddings. In: d'Amato, C., Fernández, M., Tamma, V.A.M., Lécué, F., Cudré-Mauroux, P., Sequeda, J.F., Lange, C., Heflin, J. (eds.) The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10587, pp. 190–207. Springer (2017). https://doi.org/10.1007/978-3-319-68288-4_12

[86] Cruz, I.F., Antonelli, F.P., Stroe, C.: Efficient selection of mappings and automatic quality-driven combination of matching methods. In: Shvaiko, P., Euzenat, J., Giunchiglia, F., Stuckenschmidt, H., Noy, N.F., Rosenthal, A. (eds.) Proceedings of the 4th International Workshop on Ontology Matching (OM-2009) collocated with the 8th International Semantic Web Conference (ISWC-2009) Chantilly, USA, October 25, 2009. CEUR Workshop Proceedings, vol. 551. CEUR-WS.org (2009), http://ceur-ws.org/Vol-551/om2009_Tpaper5.pdf

[87] Cruz, I.F., Antonelli, F.P., Stroe, C., Keles, U.C., Maduko, A.: Using agreementmaker to align ontologies for OAEI 2009: Overview, results, and outlook. In: Shvaiko, P., Euzenat, J., Giunchiglia, F., Stuckenschmidt, H., Noy, N.F., Rosenthal, A. (eds.) Proceedings of the 4th International Workshop on Ontology Matching (OM-2009) collocated with the 8th International Semantic Web Conference (ISWC-2009) Chantilly, USA, October 25, 2009. CEUR Workshop Proceedings, vol. 551. CEUR-WS.org (2009), http://ceur-ws.org/Vol-551/oaei09_paper2.pdf

[88] Cruz, I.F., Stroe, C., Caimi, F., Fabiani, A., Pesquita, C., Couto, F.M., Palmonari, M.: Using agreementmaker to align ontologies for OAEI 2011. In: Shvaiko, P., Euzenat, J., Heath, T., Quix, C., Mao, M., Cruz, I.F. (eds.) Proceedings of the 6th International Workshop on Ontology Matching, Bonn, Germany, October 24, 2011. CEUR Workshop Proceed-

ings, vol. 814. CEUR-WS.org (2011), http://ceur-ws.org/Vol-814/oaei11_paper1.pdf

[89] Crystal, D.: The Cambridge Encyclopedia of the English Language, chap. 8, pp. 128–133. Cambridge University Press, Cambridge, U.K.; New York, 3 edn. (2018). https://doi.org/10.1017/9781108528931

[90] Crystal, D.: The Cambridge Encyclopedia of the English Language, chap. 23, pp. 452–475. Cambridge University Press, Cambridge, U.K.; New York, 3 edn. (2018). https://doi.org/10.1017/9781108528931

[91] Crystal, D.: The Cambridge Encyclopedia of the English Language, chap. 11, pp. 166–181. Cambridge University Press, Cambridge, U.K.; New York, 3 edn. (2018). https://doi.org/10.1017/9781108528931

[92] Curino, C., Orsi, G., Tanca, L.: X-SOM results for OAEI 2007. In: Shvaiko, P., Euzenat, J., Giunchiglia, F., He, B. (eds.) Proceedings of the 2nd International Workshop on Ontology Matching (OM-2007) Collocated with the 6th International Semantic Web Conference (ISWC-2007) and the 2nd Asian Semantic Web Conference (ASWC-2007), Busan, Korea, November 11, 2007. CEUR Workshop Proceedings, vol. 304. CEUR-WS.org (2007), http://ceur-ws.org/Vol-304/paper26.pdf

[93] Dai, Y., Wang, S., Xiong, N.N., Guo, W.: A survey on knowledge graph embedding: Approaches, applications and benchmarks. Electronics **9**(5), 750 (2020), https://doi.org/10.3390/electronics9050750

[94] Dang, T., Gabriel, A., Hertling, S., Roskosch, P., Wlotzka, M., Zilke, J.R., Janssen, F., Paulheim, H.: Hotmatch results for OEAI 2012. In: Shvaiko, P., Euzenat, J., Kementsietsidis, A., Mao, M., Noy, N.F., Stuckenschmidt, H. (eds.) Proceedings of the 7th International Workshop on Ontology Matching, Boston, MA, USA, November 11, 2012. CEUR Workshop Proceedings, vol. 946. CEUR-WS.org (2012), http://ceur-ws.org/Vol-946/oaei12_paper5.pdf

[95] Davarpanah, S.H., Algergawy, A., Babalou, S.: Fuzzy inference-based ontology matching using upper ontology. In: Morzy, T., Valduriez, P., Bellatreche, L. (eds.) New Trends in Databases and Information Systems - ADBIS 2015 Short Papers and Workshops, BigDap, DCSA, GID, MEBIS, OAIS, SW4CH, WISARD, Poitiers, France, September 8-11, 2015. Proceedings. Communications in Computer and Information Science, vol. 539, pp. 392–402. Springer (2015). https://doi.org/10.1007/978-3-319-23201-0_40

[96] David, J., Euzenat, J., Scharffe, F., dos Santos, C.T.: The alignment API 4.0. Semantic Web **2**(1), 3–10 (2011). `https://doi.org/10.3233/SW-2011-0028`

[97] Daza, D., Cochez, M., Groth, P.: Inductive entity representations from text via link prediction. In: Leskovec, J., Grobelnik, M., Najork, M., Tang, J., Zia, L. (eds.) WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021. pp. 798–808. ACM / IW3C2 (2021). `https://doi.org/10.1145/3442381.3450141`

[98] Destro, J.M., dos Reis, J.C., Carvalho, A.M.B.R., Ricarte, I.L.M.: Influence of semantic similarity measures on ontology cross-language mappings. In: Seffah, A., Penzenstadler, B., Alves, C., Peng, X. (eds.) Proceedings of the Symposium on Applied Computing, SAC 2017, Marrakech, Morocco, April 3-7, 2017. pp. 323–329. ACM (2017). `https://doi.org/10.1145/3019612.3019836`

[99] Destro, J.M., dos Santos, G.O., dos Reis, J.C., da Silva Torres, R., Carvalho, A.M.B.R., Ricarte, I.L.M.: EVOCROS: results for OAEI 2018. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018. CEUR Workshop Proceedings, vol. 2288, pp. 152–159. CEUR-WS.org (2018), `http://ceur-ws.org/Vol-2288/oaei18_paper6.pdf`

[100] Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2d knowledge graph embeddings. In: McIlraith, S.A., Weinberger, K.Q. (eds.) Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018. pp. 1811–1818. AAAI Press (2018), `https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17366`

[101] Dev, S., Hassan, S., Phillips, J.M.: Closed form word embedding alignment. Knowl. Inf. Syst. **63**(3), 565–588 (2021). `https://doi.org/10.1007/s10115-020-01531-7`

[102] Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the

North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/n19-1423

[103] Ding, L., Finin, T.W., Joshi, A., Pan, R., Cost, R.S., Peng, Y., Reddivari, P., Doshi, V., Sachs, J.: Swoogle: a search and metadata engine for the semantic web. In: Grossman, D.A., Gravano, L., Zhai, C., Herzog, O., Evans, D.A. (eds.) Proceedings of the 2004 ACM CIKM International Conference on Information and Knowledge Management, Washington, DC, USA, November 8-13, 2004. pp. 652–659. ACM (2004). https://doi.org/10.1145/1031171.1031289

[104] Djeddi, W.E., Khadir, M.T.: Xmapgen and xmapsig results for OAEI 2013. In: Shvaiko, P., Euzenat, J., Srinivas, K., Mao, M., Jiménez-Ruiz, E. (eds.) Proceedings of the 8th International Workshop on Ontology Matching co-located with the 12th International Semantic Web Conference (ISWC 2013), Sydney, Australia, October 21, 2013. CEUR Workshop Proceedings, vol. 1111, pp. 203–210. CEUR-WS.org (2013), http://ceur-ws.org/Vol-1111/oaei13_paper15.pdf

[105] Djeddi, W.E., Khadir, M.T.: Xmap++: results for OAEI 2014. In: Shvaiko, P., Euzenat, J., Mao, M., Jiménez-Ruiz, E., Li, J., Ngonga, A. (eds.) Proceedings of the 9th International Workshop on Ontology Matching collocated with the 13th International Semantic Web Conference (ISWC 2014), Riva del Garda, Trentino, Italy, October 20, 2014. CEUR Workshop Proceedings, vol. 1317, pp. 163–169. CEUR-WS.org (2014), http://ceur-ws.org/Vol-1317/oaei14_paper9.pdf

[106] Djeddi, W.E., Yahia, S.B., Khadir, M.T.: Xmap: results for OAEI 2018. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018. CEUR Workshop Proceedings, vol. 2288, pp. 210–215. CEUR-WS.org (2018), http://ceur-ws.org/Vol-2288/oaei18_paper15.pdf

[107] Doan, A., Halevy, A., Ives, Z.: Principles of data integration, chap. 1, pp. 1–18. Elsevier (2012)

[108] Doan, A., Halevy, A., Ives, Z.: Principles of data integration, chap. 5, pp. 121–160. Elsevier (2012)

[109] Dolan, W.B., Brockett, C.: Automatically constructing a corpus of sentential paraphrases. In: Proceedings of the Third International Workshop on Paraphrasing, IWP@IJCNLP 2005, Jeju Island, Korea, October 2005, 2005. Asian Federation of Natural Language Processing (2005), `https://aclanthology.org/I05-5002/`

[110] Dong, Z., Dong, Q.: Hownet-a hybrid language and knowledge resource. In: International Conference on Natural Language Processing and Knowledge Engineering, 2003. Proceedings. 2003. pp. 820–824. IEEE (2003). `https://doi.org/10.1109/NLPKE.2003.1276017`

[111] Doulaverakis, C., Vrochidis, S., Kompatsiaris, I.: Exploiting visual similarities for ontology alignment. In: Fred, A.L.N., Dietz, J.L.G., Aveiro, D., Liu, K., Filipe, J. (eds.) KEOD 2015 - Proceedings of the International Conference on Knowledge Engineering and Ontology Development, part of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015), Volume 2, Lisbon, Portugal, November 12-14, 2015. pp. 29–37. SciTePress (2015). `https://doi.org/10.5220/0005588200290037`

[112] Dragisic, Z., Ivanova, V., Lambrix, P., Faria, D., Jiménez-Ruiz, E., Pesquita, C.: User validation in ontology alignment. In: Groth, P., Simperl, E., Gray, A.J.G., Sabou, M., Krötzsch, M., Lécué, F., Flöck, F., Gil, Y. (eds.) The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9981, pp. 200–217 (2016). `https://doi.org/10.1007/978-3-319-46523-4_13`

[113] Dragisic, Z., Ivanova, V., Li, H., Lambrix, P.: Experiences from the anatomy track in the ontology alignment evaluation initiative. J. Biomed. Semant. **8**(1), 56:1–56:28 (2017). `https://doi.org/10.1186/s13326-017-0166-5`

[114] Eckert, K., Meilicke, C., Stuckenschmidt, H.: Improving ontology matching using meta-level learning. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E.P.B. (eds.) The Semantic Web: Research and Applications, 6th European Semantic Web Conference, ESWC 2009, Heraklion, Crete, Greece, May 31-June 4, 2009, Proceedings. Lecture Notes in Computer

Science, vol. 5554, pp. 158–172. Springer (2009). https://doi.org/10.1007/978-3-642-02121-3_15

[115] Ehrig, M.: Ontology Alignment: Bridging the Semantic Gap, Semantic Web and Beyond: Computing for Human Experience, vol. 4, chap. 3, pp. 37–43. Springer (2007). https://doi.org/10.1007/978-0-387-36501-5

[116] Ehrlinger, L., Wöß, W.: Towards a definition of knowledge graphs. In: Martin, M., Cuquet, M., Folmer, E. (eds.) Joint Proceedings of the Posters and Demos Track of the 12th International Conference on Semantic Systems - SEMANTiCS2016 and the 1st International Workshop on Semantic Change & Evolving Semantics (SuCCESS'16) co-located with the 12th International Conference on Semantic Systems (SEMANTiCS 2016), Leipzig, Germany, September 12-15, 2016. CEUR Workshop Proceedings, vol. 1695. CEUR-WS.org (2016), http://ceur-ws.org/Vol-1695/paper4.pdf

[117] Ehrmann, M., Cecconi, F., Vannella, D., McCrae, J.P., Cimiano, P., Navigli, R.: Representing multilingual data as linked data: the case of babelnet 2.0. In: Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (eds.) Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014. pp. 401–408. European Language Resources Association (ELRA) (2014), http://www.lrec-conf.org/proceedings/lrec2014/summaries/810.html

[118] El Maarouf, I., Mansar, Y., Mouilleron, V., Valsamou-Stanislawski, D.: The finsim 2020 shared task: Learning semantic representations for the financial domain. In: Proceedings of the Second Workshop on Financial Technology and Natural Language Processing. pp. 81–86 (2020)

[119] Euzenat, J., Ferrara, A., van Hage, W.R., Hollink, L., Meilicke, C., Nikolov, A., Ritze, D., Scharffe, F., Shvaiko, P., Stuckenschmidt, H., Sváb-Zamazal, O., dos Santos, C.T.: Results of the ontology alignment evaluation initiative 2011. In: Shvaiko, P., Euzenat, J., Heath, T., Quix, C., Mao, M., Cruz, I.F. (eds.) Proceedings of the 6th International Workshop on Ontology Matching, Bonn, Germany, October 24, 2011. CEUR Workshop Proceedings, vol. 814. CEUR-WS.org (2011), http://ceur-ws.org/Vol-814/oaei11_paper0.pdf

[120] Euzenat, J., Loup, D., Touzani, M., Valtchev, P.: Ontology alignment with OLA. In: Sure, Y., Corcho, Ó., Euzenat, J., Hughes, T. (eds.) EON 2004,

Evaluation of Ontology-based Tools, Proceedings of the 3rd International Workshop on Evaluation of Ontology-based Tools held at the 3rd International Semantic Web Conference ISWC 2004, 7th November 2004, Hiroshima Prince Hotel, Hiroshima, Japan. CEUR Workshop Proceedings, vol. 128. CEUR-WS.org (2004), http://ceur-ws.org/Vol-128/EON2004_EXP_Euzenat.pdf

[121] Euzenat, J., Meilicke, C., Stuckenschmidt, H., Shvaiko, P., dos Santos, C.T.: Ontology alignment evaluation initiative: Six years of experience. J. Data Semant. **15**, 158–192 (2011). https://doi.org/10.1007/978-3-642-22630-4_6

[122] Euzenat, J., Mocan, A., Scharffe, F.: Ontology alignments an ontology management perspective. In: Hepp, M., Leenheer, P.D., de Moor, A., Sure, Y. (eds.) Ontology Management, Semantic Web, Semantic Web Services, and Business Applications, Semantic Web and Beyond: Computing for Human Experience, vol. 7, pp. 177–206. Springer (2008)

[123] Euzenat, J., Scharffe, F., Zimmermann, A.: Expressive Alignment Language and Implementation (2007)

[124] Euzenat, J., Shvaiko, P.: Ontology Matching, chap. 2, pp. 25–54. Springer, New York, 2nd edn. (2013)

[125] Euzenat, J., Shvaiko, P.: Ontology Matching, chap. 10, pp. 321–351. Springer, New York, 2nd edn. (2013)

[126] Euzenat, J., Shvaiko, P.: Ontology Matching, chap. 3, pp. 55–69. Springer, New York, 2nd edn. (2013)

[127] Euzenat, J., Shvaiko, P.: Ontology Matching, chap. Preface, pp. vii–xiii. Springer, New York, 2nd edn. (2013)

[128] Euzenat, J., Shvaiko, P.: Ontology Matching. Springer, New York, 2nd edn. (2013)

[129] Euzenat, J., Shvaiko, P.: Ontology Matching, chap. 4, pp. 73–84. Springer, New York, 2nd edn. (2013)

[130] Euzenat, J., Shvaiko, P.: Ontology Matching, chap. 13, pp. 399–461. Springer, New York, 2nd edn. (2013)

[131] Euzenat, J., Shvaiko, P.: Ontology Matching, chap. 9, pp. 285–317. Springer, New York, 2nd edn. (2013)

[132] Euzenat, J., Shvaiko, P.: Ontology Matching, chap. 7, pp. 149–197. Springer, New York, 2nd edn. (2013)

[133] Fahad, M.: ER2OWL: generating OWL ontology from ER diagram. In: Shi, Z., Mercier-Laurent, E., Leake, D.B. (eds.) Intelligent Information Processing IV, 5th IFIP International Conference on Intelligent Information Processing, October 19-22, 2008, Beijing, China. IFIP Advances in Information and Communication Technology, vol. 288, pp. 28–37. Springer (2008). `https://doi.org/10.1007/978-0-387-87685-6_6`

[134] Fahad, M.: DKP-AOM: results for OAEI 2015. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Proceedings of the 10th International Workshop on Ontology Matching collocated with the 14th International Semantic Web Conference (ISWC 2015), Bethlehem, PA, USA, October 12, 2015. CEUR Workshop Proceedings, vol. 1545, pp. 136–144. CEUR-WS.org (2015), `http://ceur-ws.org/Vol-1545/oaei15_paper4.pdf`

[135] Fahad, M.: DKP-AOM: results for OAEI 2016. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O., Ichise, R. (eds.) Proceedings of the 11th International Workshop on Ontology Matching co-located with the 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 18, 2016. CEUR Workshop Proceedings, vol. 1766, pp. 166–171. CEUR-WS.org (2016), `http://ceur-ws.org/Vol-1766/oaei16_paper6.pdf`

[136] Fallatah, O., Zhang, Z., Hopfgartner, F.: A gold standard dataset for large knowledge graphs matching. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 15th International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020), Virtual conference (originally planned to be in Athens, Greece), November 2, 2020. CEUR Workshop Proceedings, vol. 2788, pp. 24–35. CEUR-WS.org (2020), `http://ceur-ws.org/Vol-2788/om2020_LTpaper3.pdf`

[137] Fallatah, O., Zhang, Z., Hopfgartner, F.: Kgmatcher results for OAEI 2021. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), Virtual conference, October 25, 2021. CEUR Workshop Proceedings, vol. 3063, pp. 160–166. CEUR-WS.org (2021), `http://ceur-ws.org/Vol-3063/oaei21_paper8.pdf`

[138] Fang, A., Hong, N., Wu, S., Zheng, J., Qian, Q.: An integrated biomedical ontology mapping strategy based on multiple mapping methods. In: Huang, Z., Liu, C., He, J., Huang, G. (eds.) Web Information Systems Engineering - WISE 2013 Workshops - WISE 2013 International Workshops BigWebData, MBC, PCS, STeH, QUAT, SCEH, and STSC 2013, Nanjing, China, October 13-15, 2013, Revised Selected Papers. Lecture Notes in Computer Science, vol. 8182, pp. 373–386. Springer (2013). `https://doi.org/10.1007/978-3-642-54370-8_32`

[139] Faria, D., Balasubramani, B.S., Shivaprabhu, V.R., Mott, I., Pesquita, C., Couto, F.M., Cruz, I.F.: Results of AML in OAEI 2017. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Proceedings of the 12th International Workshop on Ontology Matching co-located with the 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 21, 2017. CEUR Workshop Proceedings, vol. 2032, pp. 122–128. CEUR-WS.org (2017), `http://ceur-ws.org/Vol-2032/oaei17_paper2.pdf`

[140] Faria, D., Lima, B., Silva, M.C., Couto, F.M., Pesquita, C.: AML and AMLC results for OAEI 2021. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), Virtual conference, October 25, 2021. CEUR Workshop Proceedings, vol. 3063, pp. 131–136. CEUR-WS.org (2021), `http://ceur-ws.org/Vol-3063/oaei21_paper4.pdf`

[141] Faria, D., Martins, C., Nanavaty, A., Taheri, A., Pesquita, C., Santos, E., Cruz, I.F., Couto, F.M.: Agreementmakerlight results for OAEI 2014. In: Shvaiko, P., Euzenat, J., Mao, M., Jiménez-Ruiz, E., Li, J., Ngonga, A. (eds.) Proceedings of the 9th International Workshop on Ontology Matching collocated with the 13th International Semantic Web Conference (ISWC 2014), Riva del Garda, Trentino, Italy, October 20, 2014. CEUR Workshop Proceedings, vol. 1317, pp. 105–112. CEUR-WS.org (2014), `http://ceur-ws.org/Vol-1317/oaei14_paper1.pdf`

[142] Faria, D., Pesquita, C., Balasubramani, B.S., Martins, C., Cardoso, J., Curado, H., Couto, F.M., Cruz, I.F.: OAEI 2016 results of AML. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O., Ichise, R. (eds.) Proceedings of the 11th International Workshop on Ontology Matching co-located with the 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 18, 2016. CEUR Workshop Pro-

ceedings, vol. 1766, pp. 138–145. CEUR-WS.org (2016), `http://ceur-ws.org/Vol-1766/oaei16_paper2.pdf`

[143] Faria, D., Pesquita, C., Balasubramani, B.S., Tervo, T., Carriço, D., Garrilha, R., Couto, F.M., Cruz, I.F.: Results of AML participation in OAEI 2018. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018. CEUR Workshop Proceedings, vol. 2288, pp. 125–131. CEUR-WS.org (2018), `http://ceur-ws.org/Vol-2288/oaei18_paper2.pdf`

[144] Faria, D., Pesquita, C., Mott, I., Martins, C., Couto, F.M., Cruz, I.F.: Tackling the challenges of matching biomedical ontologies. J. Biomed. Semant. **9**(1), 4:1–4:19 (2018). `https://doi.org/10.1186/s13326-017-0170-9`

[145] Faria, D., Pesquita, C., Santos, E., Cruz, I.F., Couto, F.M.: Agreementmakerlight results for OAEI 2013. In: Shvaiko, P., Euzenat, J., Srinivas, K., Mao, M., Jiménez-Ruiz, E. (eds.) Proceedings of the 8th International Workshop on Ontology Matching co-located with the 12th International Semantic Web Conference (ISWC 2013), Sydney, Australia, October 21, 2013. CEUR Workshop Proceedings, vol. 1111, pp. 101–108. CEUR-WS.org (2013), `http://ceur-ws.org/Vol-1111/oaei13_paper1.pdf`

[146] Faria, D., Pesquita, C., Santos, E., Cruz, I.F., Couto, F.M.: Automatic background knowledge selection for matching biomedical ontologies. PloS one **9**(11), e111226 (2014), `https://www.doi.org/10.1371/journal.pone.0111226`

[147] Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I.F., Couto, F.M.: The agreementmakerlight ontology matching system. In: Meersman, R., Panetto, H., Dillon, T.S., Eder, J., Bellahsene, Z., Ritter, N., Leenheer, P.D., Dou, D. (eds.) On the Move to Meaningful Internet Systems: OTM 2013 Conferences - Confederated International Conferences: CoopIS, DOA-Trusted Cloud, and ODBASE 2013, Graz, Austria, September 9-13, 2013. Proceedings. Lecture Notes in Computer Science, vol. 8185, pp. 527–541. Springer (2013). `https://doi.org/10.1007/978-3-642-41030-7_38`

[148] Fatemi, H., Sayyadi, M., Abolhassani, H.: Using background knowledge and context knowledge in ontology mapping. In: Borgo, S., Lesmo, L. (eds.) Formal Ontologies Meet Industry, Proceedings of the 3th Workshop FOMI 2008, June 5-6, 2008, Torino, Italy. Frontiers in Artificial In-

telligence and Applications, vol. 174, pp. 56–64. IOS Press (2008), `http://www.booksonline.iospress.nl/Content/View.aspx?piid=8891`

[149] Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. Language, Speech, and Communication, MIT Press, Cambridge, Massachusetts (1998). `https://doi.org/10.7551/mitpress/7287.001.0001`

[150] Fernández, J.D., Martínez-Prieto, M.A., Gutiérrez, C., Polleres, A., Arias, M.: Binary RDF representation for publication and exchange (HDT). J. Web Semant. **19**, 22–41 (2013). `https://doi.org/10.1016/j.websem.2013.01.002`

[151] Fernández, S., Marsá-Maestre, I., Velasco, J.R.: Performing ontology alignment via a fuzzy-logic multi-layer architecture. In: Fred, A.L.N., Dietz, J.L.G., Liu, K., Filipe, J. (eds.) Knowledge Discovery, Knowledge Engineering and Knowledge Management - 4th International Joint Conference, IC3K 2012, Barcelona, Spain, October 4-7, 2012, Revised Selected Papers. Communications in Computer and Information Science, vol. 415, pp. 194–210. Springer (2012). `https://doi.org/10.1007/978-3-642-54105-6_13`

[152] Fernández, S., Velasco, J.R., López-Carmona, M.A.: A fuzzy rule-based system for ontology mapping. In: Yang, J., Yokoo, M., Ito, T., Jin, Z., Scerri, P. (eds.) Principles of Practice in Multi-Agent Systems, 12th International Conference, PRIMA 2009, Nagoya, Japan, December 14-16, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5925, pp. 500–507. Springer (2009). `https://doi.org/10.1007/978-3-642-11161-7_35`

[153] Fernández, S., Velasco, J.R., Marsá-Maestre, I., López-Carmona, M.A.: Fuzzyalign - A fuzzy method for ontology alignment. In: Filipe, J., Dietz, J.L.G. (eds.) KEOD 2012 - Proceedings of the International Conference on Knowledge Engineering and Ontology Development, Barcelona, Spain, 4 - 7 October, 2012. pp. 98–107. SciTePress (2012)

[154] Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., Ruppin, E.: Placing search in context: the concept revisited. ACM Trans. Inf. Syst. **20**(1), 116–131 (2002). `https://doi.org/10.1145/503104.503110`

[155] Forcada, M.L., Ginestí-Rosell, M., Nordfalk, J., O'Regan, J., Ortiz-Rojas, S., Pérez-Ortiz, J.A., Sánchez-Martínez, F., Ramírez-Sánchez, G., Tyers, F.M.:

Apertium: a free/open-source platform for rule-based machine translation. Mach. Transl. **25**(2), 127–144 (2011). `https://doi.org/10.1007/s10590-011-9090-0`

[156] Forsati, R., Shamsfard, M.: Symbiosis of evolutionary and combinatorial ontology mapping approaches. Inf. Sci. **342**, 53–80 (2016). `https://doi.org/10.1016/j.ins.2016.01.025`

[157] Fu, B., Brennan, R., O'Sullivan, D.: Cross-lingual ontology mapping - an investigation of the impact of machine translation. In: Gómez-Pérez, A., Yu, Y., Ding, Y. (eds.) The Semantic Web, Fourth Asian Conference, ASWC 2009, Shanghai, China, December 6-9, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5926, pp. 1–15. Springer (2009). `https://doi.org/10.1007/978-3-642-10871-6_1`

[158] Fu, B., Brennan, R., O'Sullivan, D.: Cross-lingual ontology mapping and its use on the multilingual semantic web. In: Buitelaar, P., Cimiano, P., Montiel-Ponsoda, E. (eds.) Proceedings of the 1st International Workshop on the Multilingual Semantic Web, Raleigh, North Carolina, USA, April 27th, 2010. CEUR Workshop Proceedings, vol. 571, pp. 13–20. CEUR-WS.org (2010), `http://ceur-ws.org/Vol-571/paper3.pdf`

[159] Fu, B., Brennan, R., O'Sullivan, D.: Using pseudo feedback to improve cross-lingual ontology mapping. In: Antoniou, G., Grobelnik, M., Simperl, E.P.B., Parsia, B., Plexousakis, D., Leenheer, P.D., Pan, J.Z. (eds.) The Semantic Web: Research and Applications - 8th Extended Semantic Web Conference, ESWC 2011, Heraklion, Crete, Greece, May 29-June 2, 2011, Proceedings, Part I. Lecture Notes in Computer Science, vol. 6643, pp. 336–351. Springer (2011). `https://doi.org/10.1007/978-3-642-21034-1_23`

[160] Fu, B., Brennan, R., O'Sullivan, D.: A configurable translation-based cross-lingual ontology mapping system to adjust mapping outcomes. J. Web Semant. **15**, 15–36 (2012). `https://doi.org/10.1016/j.websem.2012.06.001`

[161] Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Schneider, L.: Sweetening ontologies with DOLCE. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, 13th International Conference, EKAW 2002, Siguenza, Spain, October 1-4, 2002, Proceedings. Lecture Notes in Computer Sci-

ence, vol. 2473, pp. 166–181. Springer (2002). https://doi.org/10.1007/3-540-45810-7_18

[162] García-Castro, R., Esteban-Gutiérrez, M., Gómez-Pérez, A.: Towards an infrastructure for the evaluation of semantic technologies. In: eChallenges e-2010 Conference. pp. 1–7 (2010), https://ieeexplore.ieee.org/document/5756530

[163] Gesese, G.A., Biswas, R., Alam, M., Sack, H.: A survey on knowledge graph embeddings with literals: Which model links better literal-ly? Semantic Web **12**(4), 617–647 (2021). https://doi.org/10.3233/SW-200404

[164] Ghazvinian, A., Noy, N.F., Jonquet, C., Shah, N.H., Musen, M.A.: What four million mappings can tell you about two hundred ontologies. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) The Semantic Web - ISWC 2009, 8th International Semantic Web Conference, ISWC 2009, Chantilly, VA, USA, October 25-29, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5823, pp. 229–242. Springer (2009). https://doi.org/10.1007/978-3-642-04930-9_15

[165] Gherbi, S., Khadir, M.T.: ONTMAT: results for OAEI 2017. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Proceedings of the 12th International Workshop on Ontology Matching co-located with the 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 21, 2017. CEUR Workshop Proceedings, vol. 2032, pp. 166–170. CEUR-WS.org (2017), http://ceur-ws.org/Vol-2032/oaei17_paper9.pdf

[166] Gherbi, S., Khadir, M.T.: ONTMAT1: results for OAEI 2019. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 14th International Workshop on Ontology Matching co-located with the 18th International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 26, 2019. CEUR Workshop Proceedings, vol. 2536, pp. 164–168. CEUR-WS.org (2019), http://ceur-ws.org/Vol-2536/oaei19_paper12.pdf

[167] Ghidini, C., Serafini, L., Tessaris, S.: On relating heterogeneous elements from different ontologies. In: Kokinov, B.N., Richardson, D.C., Roth-Berghofer, T., Vieu, L. (eds.) Modeling and Using Context, 6th International and Interdisciplinary Conference, CONTEXT 2007, Roskilde, Denmark, August 20-24, 2007, Proceedings. Lecture Notes in Computer Sci-

ence, vol. 4635, pp. 234–247. Springer (2007). https://doi.org/10.1007/978-3-540-74255-5_18

[168] Gligorov, R., ten Kate, W., Aleksovski, Z., van Harmelen, F.: Using google distance to weight approximate ontology matches. In: Williamson, C.L., Zurko, M.E., Patel-Schneider, P.F., Shenoy, P.J. (eds.) Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007. pp. 767–776. ACM (2007). https://doi.org/10.1145/1242572.1242676

[169] Gracia, J., Asooja, K.: Monolingual and cross-lingual ontology matching with CIDER-CL: evaluation report for OAEI 2013. In: Shvaiko, P., Euzenat, J., Srinivas, K., Mao, M., Jiménez-Ruiz, E. (eds.) Proceedings of the 8th International Workshop on Ontology Matching co-located with the 12th International Semantic Web Conference (ISWC 2013), Sydney, Australia, October 21, 2013. CEUR Workshop Proceedings, vol. 1111, pp. 109–116. CEUR-WS.org (2013), http://ceur-ws.org/Vol-1111/oaei13_paper2.pdf

[170] Gracia, J., Bernad, J., Mena, E.: Ontology matching with CIDER: evaluation report for OAEI 2011. In: Shvaiko, P., Euzenat, J., Heath, T., Quix, C., Mao, M., Cruz, I.F. (eds.) Proceedings of the 6th International Workshop on Ontology Matching, Bonn, Germany, October 24, 2011. CEUR Workshop Proceedings, vol. 814. CEUR-WS.org (2011), http://ceur-ws.org/Vol-814/oaei11_paper3.pdf

[171] Groß, A., Hartung, M., Kirsten, T., Rahm, E.: Mapping composition for matching large life science ontologies. In: Bodenreider, O., Martone, M.E., Ruttenberg, A. (eds.) Proceedings of the 2nd International Conference on Biomedical Ontology, Buffalo, NY, USA, July 26-30, 2011. CEUR Workshop Proceedings, vol. 833. CEUR-WS.org (2011), http://ceur-ws.org/Vol-833/paper15.pdf

[172] Groß, A., Hartung, M., Kirsten, T., Rahm, E.: GOMMA results for OAEI 2012. In: Shvaiko, P., Euzenat, J., Kementsietsidis, A., Mao, M., Noy, N.F., Stuckenschmidt, H. (eds.) Proceedings of the 7th International Workshop on Ontology Matching, Boston, MA, USA, November 11, 2012. CEUR Workshop Proceedings, vol. 946. CEUR-WS.org (2012), http://ceur-ws.org/Vol-946/oaei12_paper3.pdf

[173] Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Krishnapuram, B., Shah, M., Smola, A.J., Aggarwal, C.C., Shen, D., Ras-

togi, R. (eds.) Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016. pp. 855–864. ACM (2016). `https://doi.org/10.1145/2939672.2939754`

[174] Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing? Int. J. Hum. Comput. Stud. **43**(5-6), 907–928 (1995). `https://doi.org/10.1006/ijhc.1995.1081`

[175] Grütze, T., Böhm, C., Naumann, F.: Holistic and scalable ontology alignment for linked open data. In: Bizer, C., Heath, T., Berners-Lee, T., Hausenblas, M. (eds.) WWW2012 Workshop on Linked Data on the Web, Lyon, France, 16 April, 2012. CEUR Workshop Proceedings, vol. 937. CEUR-WS.org (2012), `http://ceur-ws.org/Vol-937/ldow2012-paper-08.pdf`

[176] Gulic, M., Magdalenic, I., Vrdoljak, B.: Ontology matching using TF/IDF measure with synonym recognition. In: Skersys, T., Butleris, R., Butkiene, R. (eds.) Information and Software Technologies - 19th International Conference, ICIST 2013, Kaunas, Lithuania, October 2013. Proceedings. Communications in Computer and Information Science, vol. 403, pp. 22–33. Springer (2013). `https://doi.org/10.1007/978-3-642-41947-8_3`

[177] Gulic, M., Vrdoljak, B., Banek, M.: Cromatcher results for OAEI 2016. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O., Ichise, R. (eds.) Proceedings of the 11th International Workshop on Ontology Matching co-located with the 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 18, 2016. CEUR Workshop Proceedings, vol. 1766, pp. 153–160. CEUR-WS.org (2016), `http://ceur-ws.org/Vol-1766/oaei16_paper4.pdf`

[178] Haendel, M.A., Balhoff, J.P., Bastian, F.B., Blackburn, D.C., Blake, J.A., Bradford, Y.M., Comte, A., Dahdul, W.M., Dececchi, T., Druzinsky, R.E., Hayamizu, T.F., Ibrahim, N., Lewis, S.E., Mabee, P.M., Niknejad, A., Robinson-Rechavi, M., Sereno, P.C., Mungall, C.J.: Unification of multi-species vertebrate anatomy ontologies for comparative biology in uberon. J. Biomed. Semant. **5**, 21 (2014). `https://doi.org/10.1186/2041-1480-5-21`

[179] van Hage, W.R., Katrenko, S., Schreiber, G.: A method to combine linguistic ontology-mapping techniques. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) The Semantic Web - ISWC 2005, 4th International

Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3729, pp. 732–744. Springer (2005). https://doi.org/10.1007/11574620_52

[180] Hameed, A., Preece, A.D., Sleeman, D.H.: Ontology reconciliation. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies, pp. 231–250. International Handbooks on Information Systems, Springer (2004)

[181] Hamp, B., Feldweg, H.: GermaNet - a Lexical-Semantic Net for German. In: Proceedings of the ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications. Madrid, Spain (1997)

[182] Hamp, B., Feldweg, H.: Germanet-a lexical-semantic net for german. In: Automatic information extraction and building of lexical semantic resources for NLP applications (1997)

[183] Han, X., Cao, S., Lv, X., Lin, Y., Liu, Z., Sun, M., Li, J.: Openke: An open toolkit for knowledge embedding. In: Blanco, E., Lu, W. (eds.) Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018. pp. 139–144. Association for Computational Linguistics (2018). https://doi.org/10.18653/v1/d18-2024

[184] Hao, J., Lei, C., Efthymiou, V., Quamar, A., Özcan, F., Sun, Y., Wang, W.: MEDTO: medical data to ontology matching using hybrid graph neural networks. In: Zhu, F., Ooi, B.C., Miao, C. (eds.) KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021. pp. 2946–2954. ACM (2021). https://doi.org/10.1145/3447548.3467138

[185] Harrow, I., Jiménez-Ruiz, E., Splendiani, A., Romacker, M., Woollard, P., Markel, S., Alam-Faruque, Y., Koch, M., Malone, J., Waaler, A.: Matching disease and phenotype ontologies in the ontology alignment evaluation initiative. J. Biomed. Semant. **8**(1), 55:1–55:13 (2017). https://doi.org/10.1186/s13326-017-0162-9

[186] Harrow, I., Liener, T., Jiménez-Ruiz, E.: Ontology matching for the laboratory analytics domain. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 15th International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020), Virtual conference (originally

planned to be in Athens, Greece), November 2, 2020. CEUR Workshop Proceedings, vol. 2788, pp. 237–238. CEUR-WS.org (2020), `http://ceur-ws.org/Vol-2788/om2020_poster3.pdf`

[187] Hartung, M., Groß, A., Kirsten, T., Rahm, E.: Effective composition of mappings for matching biomedical ontologies. In: Simperl, E., Norton, B., Mladenic, D., Valle, E.D., Fundulaki, I., Passant, A., Troncy, R. (eds.) The Semantic Web: ESWC 2012 Satellite Events - ESWC 2012 Satellite Events, Heraklion, Crete, Greece, May 27-31, 2012. Revised Selected Papers. Lecture Notes in Computer Science, vol. 7540, pp. 176–190. Springer (2012). `https://doi.org/10.1007/978-3-662-46641-4_13`

[188] Hassen, W.: MEDLEY results for OAEI 2012. In: Shvaiko, P., Euzenat, J., Kementsietsidis, A., Mao, M., Noy, N.F., Stuckenschmidt, H. (eds.) Proceedings of the 7th International Workshop on Ontology Matching, Boston, MA, USA, November 11, 2012. CEUR Workshop Proceedings, vol. 946. CEUR-WS.org (2012), `http://ceur-ws.org/Vol-946/oaei12_paper8.pdf`

[189] He, Y., Xue, X., Zhang, S.: Using artificial bee colony algorithm for optimizing ontology alignment. J. Inf. Hiding Multim. Signal Process. **8**(4), 766–773 (2017), `http://bit.kuas.edu.tw/%7Ejihmsp/2017/vol8/JIH-MSP-2017-04-002.pdf`

[190] Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: 14th International Conference on Computational Linguistics, COLING 1992, Nantes, France, August 23-28, 1992. pp. 539–545 (1992), `https://aclanthology.org/C92-2082/`

[191] Heist, N., Hertling, S., Ringler, D., Paulheim, H.: Knowledge graphs on the web - an overview. In: Tiddi, I., Lécué, F., Hitzler, P. (eds.) Knowledge Graphs for eXplainable Artificial Intelligence: Foundations, Applications and Challenges, Studies on the Semantic Web, vol. 47, pp. 3–22. IOS Press (2020). `https://doi.org/10.3233/SSW200009`

[192] Helou, M.A., Palmonari, M., Jarrar, M.: Effectiveness of automatic translations for cross-lingual ontology mapping. J. Artif. Intell. Res. **55**, 165–208 (2016). `https://doi.org/10.1613/jair.4789`

[193] Hendler, J.: Web 3.0 Emerging. Computer **42**(1), 111–113 (2009). `https://doi.org/10.1109/MC.2009.30`

[194] Henrich, V., Hinrichs, E.W.: Gernedit - the germanet editing tool. In: Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.) Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta. European Language Resources Association (2010), `http://www.lrec-conf.org/proceedings/lrec2010/summaries/264.html`

[195] Hepp, M., Leenheer, P.D., de Moor, A., Sure, Y. (eds.): Ontology Management, Semantic Web, Semantic Web Services, and Business Applications, Semantic Web and Beyond: Computing for Human Experience, vol. 7. Springer (2008)

[196] Hertling, S., Paulheim, H.: Wikimatch - using wikipedia for ontology matching. In: Shvaiko, P., Euzenat, J., Kementsietsidis, A., Mao, M., Noy, N.F., Stuckenschmidt, H. (eds.) Proceedings of the 7th International Workshop on Ontology Matching, Boston, MA, USA, November 11, 2012. CEUR Workshop Proceedings, vol. 946. CEUR-WS.org (2012), `http://ceur-ws.org/Vol-946/om2012_Tpaper4.pdf`

[197] Hertling, S., Paulheim, H.: Wikimatch results for OEAI 2012. In: Shvaiko, P., Euzenat, J., Kementsietsidis, A., Mao, M., Noy, N.F., Stuckenschmidt, H. (eds.) Proceedings of the 7th International Workshop on Ontology Matching, Boston, MA, USA, November 11, 2012. CEUR Workshop Proceedings, vol. 946. CEUR-WS.org (2012), `http://ceur-ws.org/Vol-946/oaei12_paper15.pdf`

[198] Hertling, S., Paulheim, H.: Webisalod: Providing hypernymy relations extracted from the web as linked open data. In: d'Amato, C., Fernández, M., Tamma, V.A.M., Lécué, F., Cudré-Mauroux, P., Sequeda, J.F., Lange, C., Heflin, J. (eds.) The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part II. Lecture Notes in Computer Science, vol. 10588, pp. 111–119. Springer (2017). `https://doi.org/10.1007/978-3-319-68204-4_11`

[199] Hertling, S., Paulheim, H.: Dbkwik: A consolidated knowledge graph from thousands of wikis. In: Wu, X., Ong, Y., Aggarwal, C.C., Chen, H. (eds.) 2018 IEEE International Conference on Big Knowledge, ICBK 2018, Singapore, November 17-18, 2018. pp. 17–24. IEEE Computer Society (2018). `https://doi.org/10.1109/ICBK.2018.00011`

[200] Hertling, S., Paulheim, H.: DOME results for OAEI 2018. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Pro-

ceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018. CEUR Workshop Proceedings, vol. 2288, pp. 144–151. CEUR-WS.org (2018), `http://ceur-ws.org/Vol-2288/oaei18_paper5.pdf`

[201] Hertling, S., Paulheim, H.: DOME results for OAEI 2019. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 14th International Workshop on Ontology Matching co-located with the 18th International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 26, 2019. CEUR Workshop Proceedings, vol. 2536, pp. 123–130. CEUR-WS.org (2019), `http://ceur-ws.org/Vol-2536/oaei19_paper6.pdf`

[202] Hertling, S., Paulheim, H.: The knowledge graph track at OAEI - gold standards, baselines, and the golden hammer bias. In: Harth, A., Kirrane, S., Ngomo, A.N., Paulheim, H., Rula, A., Gentile, A.L., Haase, P., Cochez, M. (eds.) The Semantic Web - 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31-June 4, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12123, pp. 343–359. Springer (2020). `https://doi.org/10.1007/978-3-030-49461-2_20`

[203] Hertling, S., Portisch, J., Paulheim, H.: MELT - matching evaluation toolkit. In: Acosta, M., Cudré-Mauroux, P., Maleshkova, M., Pellegrini, T., Sack, H., Sure-Vetter, Y. (eds.) Semantic Systems. The Power of AI and Knowledge Graphs - 15th International Conference, SEMANTiCS 2019, Karlsruhe, Germany, September 9-12, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11702, pp. 231–245. Springer (2019). `https://doi.org/10.1007/978-3-030-33220-4_17`

[204] Hertling, S., Portisch, J., Paulheim, H.: Supervised ontology and instance matching with MELT. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 15th International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020), Virtual conference (originally planned to be in Athens, Greece), November 2, 2020. CEUR Workshop Proceedings, vol. 2788, pp. 60–71. CEUR-WS.org (2020), `http://ceur-ws.org/Vol-2788/om2020_LTpaper6.pdf`

[205] Hertling, S., Portisch, J., Paulheim, H.: Matching with transformers in MELT. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 16th International Workshop on Ontol-

ogy Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), Virtual conference, October 25, 2021. CEUR Workshop Proceedings, vol. 3063, pp. 13–24. CEUR-WS.org (2021), `http://ceur-ws.org/Vol-3063/om2021_LTpaper2.pdf`

[206] Hertling, S., Portisch, J., Paulheim, H.: KERMIT - A transformer-based approach for knowledge graph matching. CoRR **abs/2204.13931** (2022). `https://doi.org/10.48550/arXiv.2204.13931`

[207] Hess, G.N., Iochpe, C., Castano, S.: Geographic ontology matching with ig-match. In: Papadias, D., Zhang, D., Kollios, G. (eds.) Advances in Spatial and Temporal Databases, 10th International Symposium, SSTD 2007, Boston, MA, USA, July 16-18, 2007, Proceedings. Lecture Notes in Computer Science, vol. 4605, pp. 185–202. Springer (2007). `https://doi.org/10.1007/978-3-540-73540-3_11`

[208] Hill, F., Reichart, R., Korhonen, A.: Simlex-999: Evaluating semantic models with (genuine) similarity estimation. Comput. Linguistics **41**(4), 665–695 (2015). `https://doi.org/10.1162/COLI_a_00237`

[209] Hitzler, P., Krötzsch, M., Rudolph, S., Sure, Y.: Semantic Web: Grundlagen, chap. 1, pp. 9–14. eXamen.press, Springer, Berlin, 1st edn. (2008)

[210] Hitzler, P., Krötzsch, M., Rudolph, S., Sure, Y.: Semantic Web: Grundlagen. eXamen.press, Springer, Berlin, 1st edn. (2008)

[211] Hitzler, P., Krötzsch, M., Rudolph, S., Sure, Y.: Semantic Web: Grundlagen, chap. 3, pp. 36–88. eXamen.press, Springer, Berlin, 1st edn. (2008)

[212] Hitzler, P., Krötzsch, M., Rudolph, S., Sure, Y.: Semantic Web: Grundlagen, chap. 5, pp. 126–158. eXamen.press, Springer, Berlin, 1st edn. (2008)

[213] Hitzler, P., Krötzsch, M., Rudolph, S., Sure, Y.: Semantic Web: Grundlagen, chap. 7, pp. 202–241. eXamen.press, Springer, Berlin, 1st edn. (2008)

[214] Hnatkowska, B., Kozierkiewicz, A., Pietranik, M.: Fuzzy based approach to ontology relations alignment. In: 30th IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2021, Luxembourg, July 11-14, 2021. pp. 1–7. IEEE (2021). `https://doi.org/10.1109/FUZZ45933.2021.9494564`

[215] Hoffart, J., Seufert, S., Nguyen, D.B., Theobald, M., Weikum, G.: KORE: keyphrase overlap relatedness for entity disambiguation. In: Chen, X., Lebanon, G., Wang, H., Zaki, M.J. (eds.) 21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI,

USA, October 29 - November 02, 2012. pp. 545–554. ACM (2012). https://doi.org/10.1145/2396761.2396832

[216] Hofmann, A., Perchani, S., Portisch, J., Hertling, S., Paulheim, H.: Dbkwik: Towards knowledge graph creation from thousands of wikis. In: Nikitina, N., Song, D., Fokoue, A., Haase, P. (eds.) Proceedings of the ISWC 2017 Posters & Demonstrations and Industry Tracks co-located with 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 23rd - to - 25th, 2017. CEUR Workshop Proceedings, vol. 1963. CEUR-WS.org (2017), http://ceur-ws.org/Vol-1963/paper540.pdf

[217] Holmes, D.O., McCabe, M.C.: Improving precision and recall for soundex retrieval. In: 2002 International Symposium on Information Technology (ITCC 2002), 8-10 April 2002, Las Vegas, NV, USA. pp. 22–27. IEEE Computer Society (2002). https://doi.org/10.1109/ITCC.2002.1000354

[218] Holter, O.M., Myklebust, E.B., Chen, J., Jiménez-Ruiz, E.: Embedding OWL ontologies with owl2vec. In: Suárez-Figueroa, M.C., Cheng, G., Gentile, A.L., Guéret, C., Keet, C.M., Bernstein, A. (eds.) Proceedings of the ISWC 2019 Satellite Tracks (Posters & Demonstrations, Industry, and Outrageous Ideas) co-located with 18th International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 26-30, 2019. CEUR Workshop Proceedings, vol. 2456, pp. 33–36. CEUR-WS.org (2019), http://ceur-ws.org/Vol-2456/paper9.pdf

[219] Horridge, M., Bechhofer, S., Noppens, O.: Igniting the OWL 1.1 touch paper: The OWL API. In: Golbreich, C., Kalyanpur, A., Parsia, B. (eds.) Proceedings of the OWLED 2007 Workshop on OWL: Experiences and Directions, Innsbruck, Austria, June 6-7, 2007. CEUR Workshop Proceedings, vol. 258. CEUR-WS.org (2007), http://ceur-ws.org/Vol-258/paper19.pdf

[220] Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL A Semantic Web Rule Language Combining OWL and RuleML (2004), https://www-old.fmi.uni-sofia.bg/Members/marian/411430437438-43e442-43743d43043d43844f-43743843c43543d-44143543c43544144244a440-2008-2009-44344743543143d430-43343e43443843d430-knowledge-bases/42344743543143d438-43c43044243544043843043b438-course-materials/SWRL%20%20A%20Semantic%20Web%20Rule%20Language%20Combining%20OWL%20and%20RuleM....pdf

[221] Hsu, C.W., Chang, C.C., Lin, C.J.: A practical guide to support vector classification (2003), `http://www.datascienceassn.org/sites/default/files/Practical%20Guide%20to%20Support%20Vector%20Classification.pdf`

[222] Hu, Y., Bai, S., Zou, S., Wang, P.: Lily results for OAEI 2020. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 15th International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020), Virtual conference (originally planned to be in Athens, Greece), November 2, 2020. CEUR Workshop Proceedings, vol. 2788, pp. 194–200. CEUR-WS.org (2020), `http://ceur-ws.org/Vol-2788/oaei20_paper9.pdf`

[223] Ibrahim, S., Fathalla, S., Lehmann, J., Jabeen, H.: Multilingual ontology merging using cross-lingual matching. In: He, J., Purohit, H., Huang, G., Gao, X., Deng, K. (eds.) IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, WI/IAT 2020, Melbourne, Australia, December 14-17, 2020. pp. 113–120. IEEE (2020). `https://doi.org/10.1109/WIIAT50758.2020.00020`

[224] Ichise, R.: Machine learning approach for ontology mapping using multiple concept similarity measures. In: Lee, R.Y. (ed.) 7th IEEE/ACIS International Conference on Computer and Information Science, IEEE/ACIS ICIS 2008, 14-16 May 2008, Portland, Oregon, USA. pp. 340–346. IEEE Computer Society (2008). `https://doi.org/10.1109/ICIS.2008.51`

[225] Idoudi, R., Ettabaâ, K.S., Solaiman, B., Hamrouni, K.: Ontology knowledge mining for ontology alignment. Int. J. Comput. Intell. Syst. **9**(5), 876–887 (2016). `https://doi.org/10.1080/18756891.2016.1237187`

[226] Ivanova, V., Bach, B., Pietriga, E., Lambrix, P.: Alignment cubes: Towards interactive visual exploration and evaluation of multiple ontology alignments. In: d'Amato, C., Fernández, M., Tamma, V.A.M., Lécué, F., Cudré-Mauroux, P., Sequeda, J.F., Lange, C., Heflin, J. (eds.) The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10587, pp. 400–417. Springer (2017). `https://doi.org/10.1007/978-3-319-68288-4_24`

[227] Iyer, V., Agarwal, A., Kumar, H.: Veealign: a supervised deep learning approach to ontology alignment. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz,

E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 15th International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020), Virtual conference (originally planned to be in Athens, Greece), November 2, 2020. CEUR Workshop Proceedings, vol. 2788, pp. 216–224. CEUR-WS.org (2020), `http://ceur-ws.org/Vol-2788/oaei20_paper13.pdf`

[228] Jaccard, P.: Lois de distribution florale dans la zone alpine. Bull Soc Vaudoise Sci Nat **38**, 69–130 (1902)

[229] Jaderberg, M., Dalibard, V., Osindero, S., Czarnecki, W.M., Donahue, J., Razavi, A., Vinyals, O., Green, T., Dunning, I., Simonyan, K., Fernando, C., Kavukcuoglu, K.: Population based training of neural networks. CoRR **abs/1711.09846** (2017), `http://arxiv.org/abs/1711.09846`

[230] Jaiboonlue, P., Nitsuwat, S., Chai, W.N., Luekhong, P., Ruangrajitpakorn, T., Supnithi, T.: A framework of automatic alignment of concept in ontology with confidence score based on inner concept information. In: Supnithi, T., Yamaguchi, T., Buranarach, M. (eds.) Workshop and Poster Proceedings of the 4th Joint International Semantic Technology Conference, JIST 2014, Chiang Mai, Thailand, November 9-11, 2014. CEUR Workshop Proceedings, vol. 1312, pp. 54–68. CEUR-WS.org (2014), `http://ceur-ws.org/Vol-1312/ldop2014_preface.pdf`

[231] Jain, P., Hitzler, P., Sheth, A.P., Verma, K., Yeh, P.Z.: Ontology alignment for linked open data. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) The Semantic Web - ISWC 2010 - 9th International Semantic Web Conference, ISWC 2010, Shanghai, China, November 7-11, 2010, Revised Selected Papers, Part I. Lecture Notes in Computer Science, vol. 6496, pp. 402–417. Springer (2010). `https://doi.org/10.1007/978-3-642-17746-0_26`

[232] Jain, P., Yeh, P.Z., Verma, K., Vasquez, R.G., Damova, M., Hitzler, P., Sheth, A.P.: Contextual ontology alignment of LOD with an upper ontology: A case study with proton. In: Antoniou, G., Grobelnik, M., Simperl, E.P.B., Parsia, B., Plexousakis, D., Leenheer, P.D., Pan, J.Z. (eds.) The Semantic Web: Research and Applications - 8th Extended Semantic Web Conference, ESWC 2011, Heraklion, Crete, Greece, May 29-June 2, 2011, Proceedings, Part I. Lecture Notes in Computer Science, vol. 6643, pp. 80–92. Springer (2011). `https://doi.org/10.1007/978-3-642-21034-1_6`

[233] Jan, S., Li, M., Al-Raweshidy, H.S., Mousavi, A., Qi, M.: Dealing with uncertain entities in ontology alignment using rough sets. IEEE Trans. Syst. Man Cybern. Part C **42**(6), 1600–1612 (2012). https://doi.org/10.1109/TSMCC.2012.2209869

[234] Jan, S., Li, M., Al-Sultany, G., Al-Raweshidy, H.S.: Ontology alignment using rough sets. In: Eighth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2011, 26-28 July 2011, Shanghai, China. pp. 2683–2686. IEEE (2011). https://doi.org/10.1109/FSKD.2011.6020069

[235] Jean-Mary, Y.R., Kabuka, M.R.: ASMOV results for OAEI 2007. In: Shvaiko, P., Euzenat, J., Giunchiglia, F., He, B. (eds.) Proceedings of the 2nd International Workshop on Ontology Matching (OM-2007) Collocated with the 6th International Semantic Web Conference (ISWC-2007) and the 2nd Asian Semantic Web Conference (ASWC-2007), Busan, Korea, November 11, 2007. CEUR Workshop Proceedings, vol. 304. CEUR-WS.org (2007), http://ceur-ws.org/Vol-304/paper12.pdf

[236] Ji, S., Pan, S., Cambria, E., Marttinen, P., Yu, P.S.: A survey on knowledge graphs: Representation, acquisition, and applications. IEEE Trans. Neural Networks Learn. Syst. **33**(2), 494–514 (2022). https://doi.org/10.1109/TNNLS.2021.3070843

[237] Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. In: Chen, K., Huang, C., Sproat, R. (eds.) Proceedings of the 10th Research on Computational Linguistics International Conference, ROCLING 1997, Taipei, Taiwan, August 1997. pp. 19–33. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP) (1997), https://aclanthology.org/O97-1002/

[238] Jiang, Y., Wang, X., Zheng, H.: A semantic similarity measure based on information distance for ontology alignment. Inf. Sci. **278**, 76–87 (2014). https://doi.org/10.1016/j.ins.2014.03.021

[239] Jiménez-Ruiz, E.: Logmap family participation in the OAEI 2020. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 15th International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020), Virtual conference (originally planned to be in Athens, Greece), November 2, 2020. CEUR Workshop Proceedings, vol. 2788, pp.

201–203. CEUR-WS.org (2020), http://ceur-ws.org/Vol-2788/oaei20_paper10.pdf

[240] Jiménez-Ruiz, E.: Logmap family participation in the OAEI 2021. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), Virtual conference, October 25, 2021. CEUR Workshop Proceedings, vol. 3063, pp. 175–177. CEUR-WS.org (2021), http://ceur-ws.org/Vol-3063/oaei21_paper10.pdf

[241] Jiménez-Ruiz, E., Grau, B.C.: Logmap: Logic-based and scalable ontology matching. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N.F., Blomqvist, E. (eds.) The Semantic Web - ISWC 2011 - 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part I. Lecture Notes in Computer Science, vol. 7031, pp. 273–288. Springer (2011). https://doi.org/10.1007/978-3-642-25073-6_18

[242] Jiménez-Ruiz, E., Grau, B.C., Cross, V.: Logmap family participation in the OAEI 2017. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Proceedings of the 12th International Workshop on Ontology Matching co-located with the 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 21, 2017. CEUR Workshop Proceedings, vol. 2032, pp. 153–157. CEUR-WS.org (2017), http://ceur-ws.org/Vol-2032/oaei17_paper7.pdf

[243] Jiménez-Ruiz, E., Grau, B.C., Cross, V.: Logmap family participation in the OAEI 2018. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018. CEUR Workshop Proceedings, vol. 2288, pp. 187–191. CEUR-WS.org (2018), http://ceur-ws.org/Vol-2288/oaei18_paper11.pdf

[244] Jiménez-Ruiz, E., Grau, B.C., Cross, V.V.: Logmap family participation in the OAEI 2016. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O., Ichise, R. (eds.) Proceedings of the 11th International Workshop on Ontology Matching co-located with the 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 18, 2016. CEUR Workshop Proceedings, vol. 1766, pp. 185–189. CEUR-WS.org (2016), http://ceur-ws.org/Vol-1766/oaei16_paper9.pdf

[245] Jiménez-Ruiz, E., Grau, B.C., Solimando, A., Cross, V.V.: Logmap family results for OAEI 2015. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Proceedings of the 10th International Workshop on Ontology Matching collocated with the 14th International Semantic Web Conference (ISWC 2015), Bethlehem, PA, USA, October 12, 2015. CEUR Workshop Proceedings, vol. 1545, pp. 171–175. CEUR-WS.org (2015), `http://ceur-ws.org/Vol-1545/oaei15_paper10.pdf`

[246] Jiménez-Ruiz, E., Grau, B.C., Xia, W., Solimando, A., Chen, X., Cross, V.V., Gong, Y., Zhang, S., Chennai-Thiagarajan, A.: Logmap family results for OAEI 2014. In: Shvaiko, P., Euzenat, J., Mao, M., Jiménez-Ruiz, E., Li, J., Ngonga, A. (eds.) Proceedings of the 9th International Workshop on Ontology Matching collocated with the 13th International Semantic Web Conference (ISWC 2014), Riva del Garda, Trentino, Italy, October 20, 2014. CEUR Workshop Proceedings, vol. 1317, pp. 126–134. CEUR-WS.org (2014), `http://ceur-ws.org/Vol-1317/oaei14_paper4.pdf`

[247] Jiménez-Ruiz, E., Hassanzadeh, O., Efthymiou, V., Chen, J., Srinivas, K.: Semtab 2019: Resources to benchmark tabular data to knowledge graph matching systems. In: Harth, A., Kirrane, S., Ngomo, A.N., Paulheim, H., Rula, A., Gentile, A.L., Haase, P., Cochez, M. (eds.) The Semantic Web - 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31-June 4, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12123, pp. 514–530. Springer (2020). `https://doi.org/10.1007/978-3-030-49461-2_30`

[248] Jiménez-Ruiz, E., Morant, A., Grau, B.C.: Logmap results for OAEI 2011. In: Shvaiko, P., Euzenat, J., Heath, T., Quix, C., Mao, M., Cruz, I.F. (eds.) Proceedings of the 6th International Workshop on Ontology Matching, Bonn, Germany, October 24, 2011. CEUR Workshop Proceedings, vol. 814. CEUR-WS.org (2011), `http://ceur-ws.org/Vol-814/oaei11_paper8.pdf`

[249] Jiménez-Ruiz, E., Saveta, T., Zamazal, O., Hertling, S., Röder, M., Fundulaki, I., Ngomo, A.N., Sherif, M.A., Annane, A., Bellahsene, Z., Yahia, S.B., Diallo, G., Faria, D., Kachroudi, M., Khiat, A., Lambrix, P., Li, H., Mackeprang, M., Mohammadi, M., Rybinski, M., Balasubramani, B.S., Trojahn, C.: Introducing the HOBBIT platform into the ontology alignment evaluation campaign. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Proceedings of the 13th International Work-

shop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018. CEUR Workshop Proceedings, vol. 2288, pp. 49–60. CEUR-WS.org (2018), http://ceur-ws.org/Vol-2288/om2018_LTpaper5.pdf

[250] Jin, T., Fu, Y., Lin, X., Liu, Q., Cui, Z.: A method of ontology mapping based on instance. In: Liao, B., Pan, J., Jain, L.C., Liao, M., Noda, H., Ho, A.T.S. (eds.) 3rd International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007), Kaohsiung, Taiwan, 26-28 November 2007, Proceedings. pp. 145–148. IEEE Computer Society (2007). https://doi.org/10.1109/IIH-MSP.2007.31

[251] Jin, T., Fu, Y., Ling, X., Liu, Q., Cui, Z.: A method of ontology mapping based on subtree kernel. In: Yu, F., Li, R., Liao, G. (eds.) Proceedings of the Workshop on Intelligent Information Technology Application, IITA 2007, Zhang Jiajie, China, December 2-3, 2007. pp. 70–73. IEEE Computer Society (2007). https://doi.org/10.1109/IITA.2007.25

[252] Jirkovský, V., Ichise, R.: MAPSOM: user involvement in ontology matching. In: Kim, W., Ding, Y., Kim, H. (eds.) Semantic Technology - Third Joint International Conference, JIST 2013, Seoul, South Korea, November 28-30, 2013, Revised Selected Papers. Lecture Notes in Computer Science, vol. 8388, pp. 348–363. Springer (2013). https://doi.org/10.1007/978-3-319-06826-8_26

[253] Kachroudi, M., Diallo, G., Yahia, S.B.: OAEI 2017 results of KEPLER. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Proceedings of the 12th International Workshop on Ontology Matching co-located with the 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 21, 2017. CEUR Workshop Proceedings, vol. 2032, pp. 138–145. CEUR-WS.org (2017), http://ceur-ws.org/Vol-2032/oaei17_paper5.pdf

[254] Kachroudi, M., Diallo, G., Yahia, S.B.: KEPLER at OAEI 2018. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018. CEUR Workshop Proceedings, vol. 2288, pp. 173–178. CEUR-WS.org (2018), http://ceur-ws.org/Vol-2288/oaei18_paper9.pdf

[255] Kachroudi, M., Moussa, E.B., Zghal, S., Yahia, S.B.: LDOA results for OAEI 2011. In: Shvaiko, P., Euzenat, J., Heath, T., Quix, C., Mao, M., Cruz, I.F. (eds.) Proceedings of the 6th International Workshop on Ontology Matching, Bonn, Germany, October 24, 2011. CEUR Workshop Proceedings, vol. 814. CEUR-WS.org (2011), `http://ceur-ws.org/Vol-814/oaei11_paper6.pdf`

[256] Kachroudi, M., Yahia, S.B.: Dealing with direct and indirect ontology alignment. J. Data Semant. **7**(4), 237–252 (2018). `https://doi.org/10.1007/s13740-018-0098-y`

[257] Kachroudi, M., Zghal, S., Yahia, S.B.: Bridging the multilingualism gap in ontology alignment. Int. J. Metadata Semant. Ontologies **9**(3), 252–262 (2014). `https://doi.org/10.1504/IJMSO.2014.063139`

[258] Kalfoglou, Y., Hu, B.: CROSI mapping system (CMS) - result of the 2005 ontology alignment contest. In: Ashpole, B., Ehrig, M., Euzenat, J., Stuckenschmidt, H. (eds.) Integrating Ontologies '05, Proceedings of the K-CAP 2005 Workshop on Integrating Ontologies, Banff, Canada, October 2, 2005. CEUR Workshop Proceedings, vol. 156. CEUR-WS.org (2005), `http://ceur-ws.org/Vol-156/paper12.pdf`

[259] Kammoun, A., Diallo, G.: Servomap results for OAEI 2013. In: Shvaiko, P., Euzenat, J., Srinivas, K., Mao, M., Jiménez-Ruiz, E. (eds.) Proceedings of the 8th International Workshop on Ontology Matching co-located with the 12th International Semantic Web Conference (ISWC 2013), Sydney, Australia, October 21, 2013. CEUR Workshop Proceedings, vol. 1111, pp. 169–176. CEUR-WS.org (2013), `http://ceur-ws.org/Vol-1111/oaei13_paper10.pdf`

[260] Karam, N., Khiat, A., Algergawy, A., Sattler, M., Weiland, C., Schmidt, M.: Matching biodiversity and ecology ontologies: challenges and evaluation results. Knowl. Eng. Rev. **35**, e9 (2020). `https://doi.org/10.1017/S0269888920000132`

[261] Karim, M.R., Cochez, M., Jares, J.B., Uddin, M., Beyan, O.D., Decker, S.: Drug-drug interaction prediction based on knowledge graph embeddings and convolutional-lstm network. In: Shi, X.M., Buck, M., Ma, J., Veltri, P. (eds.) Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, BCB 2019, Niagara Falls, NY, USA, September 7-10, 2019. pp. 113–123. ACM (2019). `https://doi.org/10.1145/3307339.3342161`

[262] Kaza, S., Chen, H.: Evaluating ontology mapping techniques: An experiment in public safety information sharing. Decis. Support Syst. **45**(4), 714–728 (2008). `https://doi.org/10.1016/j.dss.2007.12.007`

[263] Kazemi, S.M., Goel, R., Jain, K., Kobyzev, I., Sethi, A., Forsyth, P., Poupart, P.: Relational representation learning for dynamic (knowledge) graphs: A survey. CoRR **abs/1905.11485** (2019), `http://arxiv.org/abs/1905.11485`

[264] Kejriwal, M., Szekely, P.A.: Supervised typing of big graphs using semantic embeddings. In: Groppe, S., Gruenwald, L. (eds.) Proceedings of The International Workshop on Semantic Big Data, SBD@SIGMOD 2017, Chicago, IL, USA, May 19, 2017. pp. 3:1–3:6. ACM (2017). `https://doi.org/10.1145/3066911.3066918`

[265] Khan, W.A., Amin, M.B., Khattak, A.M., Hussain, M., Lee, S.: System for parallel heterogeneity resolution (sphere) results for OAEI 2013. In: Shvaiko, P., Euzenat, J., Srinivas, K., Mao, M., Jiménez-Ruiz, E. (eds.) Proceedings of the 8th International Workshop on Ontology Matching co-located with the 12th International Semantic Web Conference (ISWC 2013), Sydney, Australia, October 21, 2013. CEUR Workshop Proceedings, vol. 1111, pp. 184–189. CEUR-WS.org (2013), `http://ceur-ws.org/Vol-1111/oaei13_paper12.pdf`

[266] Kheder, N., Diallo, G.: Servombi at OAEI 2015. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Proceedings of the 10th International Workshop on Ontology Matching collocated with the 14th International Semantic Web Conference (ISWC 2015), Bethlehem, PA, USA, October 12, 2015. CEUR Workshop Proceedings, vol. 1545, pp. 200–207. CEUR-WS.org (2015), `http://ceur-ws.org/Vol-1545/oaei15_paper15.pdf`

[267] Khiat, A.: Crolom: cross-lingual ontology matching system results for OAEI 2016. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O., Ichise, R. (eds.) Proceedings of the 11th International Workshop on Ontology Matching co-located with the 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 18, 2016. CEUR Workshop Proceedings, vol. 1766, pp. 146–152. CEUR-WS.org (2016), `http://ceur-ws.org/Vol-1766/oaei16_paper3.pdf`

[268] Khiat, A., Benaissa, M.: AOT / AOTL results for OAEI 2014. In: Shvaiko, P., Euzenat, J., Mao, M., Jiménez-Ruiz, E., Li, J., Ngonga, A. (eds.) Pro-

ceedings of the 9th International Workshop on Ontology Matching collocated with the 13th International Semantic Web Conference (ISWC 2014), Riva del Garda, Trentino, Italy, October 20, 2014. CEUR Workshop Proceedings, vol. 1317, pp. 113–119. CEUR-WS.org (2014), `http://ceur-ws.org/Vol-1317/oaei14_paper2.pdf`

[269] Khiat, A., Benaissa, M.: Insmt / insmtl results for OAEI 2014 instance matching. In: Shvaiko, P., Euzenat, J., Mao, M., Jiménez-Ruiz, E., Li, J., Ngonga, A. (eds.) Proceedings of the 9th International Workshop on Ontology Matching collocated with the 13th International Semantic Web Conference (ISWC 2014), Riva del Garda, Trentino, Italy, October 20, 2014. CEUR Workshop Proceedings, vol. 1317, pp. 120–125. CEUR-WS.org (2014), `http://ceur-ws.org/Vol-1317/oaei14_paper3.pdf`

[270] Khiat, A., Mackeprang, M.: I-match and ontoidea results for OAEI 2017. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Proceedings of the 12th International Workshop on Ontology Matching co-located with the 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 21, 2017. CEUR Workshop Proceedings, vol. 2032, pp. 135–137. CEUR-WS.org (2017), `http://ceur-ws.org/Vol-2032/oaei17_paper4.pdf`

[271] Khiat, A., Ouhiba, E.A., Belfedhal, M.A., Zoua, C.E.: Simcat results for OAEI 2016. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O., Ichise, R. (eds.) Proceedings of the 11th International Workshop on Ontology Matching co-located with the 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 18, 2016. CEUR Workshop Proceedings, vol. 1766, pp. 217–221. CEUR-WS.org (2016), `http://ceur-ws.org/Vol-1766/oaei16_paper14.pdf`

[272] Kim, J., Jang, M., Ha, Y., Sohn, J., Lee, S.: Moa: OWL ontology merging and alignment tool for the semantic web. In: Ali, M., Esposito, F. (eds.) Innovations in Applied Artificial Intelligence, 18th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE 2005, Bari, Italy, June 22-24, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3533, pp. 722–731. Springer (2005). `https://doi.org/10.1007/11504894_100`

[273] Kim, W., Seo, J.: Classifying schematic and data heterogeneity in multidatabase systems. Computer **24**(12), 12–18 (1991). `https://doi.org/10.1109/2.116884`

[274] Kim, W., Park, S., Bang, S., Lee, S.: An ontology mapping algorithm between heterogeneous product classification taxonomies. In: Shvaiko, P., Euzenat, J., Noy, N.F., Stuckenschmidt, H., Benjamins, V.R., Uschold, M. (eds.) Proceedings of the 1st International Workshop on Ontology Matching (OM-2006) Collocated with the 5th International Semantic Web Conference (ISWC-2006), Athens, Georgia, USA, November 5, 2006. CEUR Workshop Proceedings, vol. 225. CEUR-WS.org (2006), `http://ceur-ws.org/Vol-225/paper24.pdf`

[275] Kiren, T., Shoaib, M.: A novel ontology matching approach using key concepts. Aslib J. Inf. Manag. **68**(1), 99–111 (2015). `https://doi.org/10.1108/AJIM-04-2015-0054`

[276] Kiu, C., Lee, C.: Ontology mapping and merging through ontodna for learning object reusability. J. Educ. Technol. Soc. **9**(3), 27–42 (2006), `http://www.ifets.info/abstract.php?art_id=651`

[277] Klein, M.: Combining and relating ontologies: an analysis of problems and solutions. In: Gómez-Pérez, A., Gruninger, M., Stuckenschmidt, H., Uschold, M. (eds.) Proceedings of the IJCAI-01 Workshop on Ontologies and Information Sharing Seattle, USA, August 4-5, 2001. CEUR Workshop Proceedings, vol. 47. CEUR-WS.org (2001), `http://ceur-ws.org/Vol-47/klein.pdf`

[278] Knorr, L., Portisch, J.: Fine-tom matcher results for OAEI 2021. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), Virtual conference, October 25, 2021. CEUR Workshop Proceedings, vol. 3063, pp. 144–151. CEUR-WS.org (2021), `http://ceur-ws.org/Vol-3063/oaei21_paper6.pdf`

[279] Kolyvakis, P., Kalousis, A., Kiritsis, D.: Deepalignment: Unsupervised ontology matching with refined word vectors. In: Walker, M.A., Ji, H., Stent, A. (eds.) Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers). pp. 787–798. Association for Computational Linguistics (2018). `https://doi.org/10.18653/v1/n18-1072`

[280] Kolyvakis, P., Kalousis, A., Smith, B., Kiritsis, D.: Biomedical ontology alignment: an approach based on representation learning. J. Biomed.

Semant. **9**(1), 21:1–21:20 (2018). https://doi.org/10.1186/s13326-018-0187-8

[281] Kossack, D., Borg, N., Knorr, L., Portisch, J.: TOM matcher results for OAEI 2021. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), Virtual conference, October 25, 2021. CEUR Workshop Proceedings, vol. 3063, pp. 193–198. CEUR-WS.org (2021), http://ceur-ws.org/Vol-3063/oaei21_paper13.pdf

[282] Kotis, K., Katasonov, A., Leino, J.: Automsv2 results for OAEI 2012. In: Shvaiko, P., Euzenat, J., Kementsietsidis, A., Mao, M., Noy, N.F., Stuckenschmidt, H. (eds.) Proceedings of the 7th International Workshop on Ontology Matching, Boston, MA, USA, November 11, 2012. CEUR Workshop Proceedings, vol. 946. CEUR-WS.org (2012), http://ceur-ws.org/Vol-946/oaei12_paper2.pdf

[283] Kotis, K., Valarakos, A.G., Vouros, G.A.: AUTOMS: automated ontology mapping through synthesis of methods. In: Shvaiko, P., Euzenat, J., Noy, N.F., Stuckenschmidt, H., Benjamins, V.R., Uschold, M. (eds.) Proceedings of the 1st International Workshop on Ontology Matching (OM-2006) Collocated with the 5th International Semantic Web Conference (ISWC-2006), Athens, Georgia, USA, November 5, 2006. CEUR Workshop Proceedings, vol. 225. CEUR-WS.org (2006), http://ceur-ws.org/Vol-225/paper8.pdf

[284] Koukourikos, A., Karampiperis, P., Stoitsis, G.: Cross-language ontology alignment utilizing machine translation models. In: Garoufallou, E., Greenberg, J. (eds.) Metadata and Semantics Research - 7th Research Conference, MTSR 2013, Thessaloniki, Greece, November 19-22, 2013. Proceedings. Communications in Computer and Information Science, vol. 390, pp. 75–86. Springer (2013). https://doi.org/10.1007/978-3-319-03437-9_9

[285] Koukourikos, A., Vouros, G.A., Karkaletsis, V.: SYNTHESIS: results for the ontology alignment evaluation initiative (OAEI) 2013. In: Shvaiko, P., Euzenat, J., Srinivas, K., Mao, M., Jiménez-Ruiz, E. (eds.) Proceedings of the 8th International Workshop on Ontology Matching co-located with the 12th International Semantic Web Conference (ISWC 2013), Sydney, Australia, October 21, 2013. CEUR Workshop Proceedings, vol. 1111, pp.

190–196. CEUR-WS.org (2013), http://ceur-ws.org/Vol-1111/oaei13_paper13.pdf

[286] Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955)

[287] Kumar, S.K., Harding, J.A.: Ontology mapping using description logic and bridging axioms. Comput. Ind. **64**(1), 19–28 (2013). https://doi.org/10.1016/j.compind.2012.09.004

[288] Laadhar, A., Ghozzi, F., Megdiche, I., Ravat, F., Teste, O., Gargouri, F.: Pomap results for OAEI 2017. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Proceedings of the 12th International Workshop on Ontology Matching co-located with the 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 21, 2017. CEUR Workshop Proceedings, vol. 2032, pp. 171–177. CEUR-WS.org (2017), http://ceur-ws.org/Vol-2032/oaei17_paper10.pdf

[289] Laadhar, A., Ghozzi, F., Megdiche, I., Ravat, F., Teste, O., Gargouri, F.: OAEI 2018 results of pomap++. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018. CEUR Workshop Proceedings, vol. 2288, pp. 192–196. CEUR-WS.org (2018), http://ceur-ws.org/Vol-2288/oaei18_paper12.pdf

[290] Laadhar, A., Ghozzi, F., Megdiche, I., Ravat, F., Teste, O., Gargouri, F.: The impact of imbalanced training data on local matching learning of ontologies. In: Abramowicz, W., Corchuelo, R. (eds.) Business Information Systems - 22nd International Conference, BIS 2019, Seville, Spain, June 26-28, 2019, Proceedings, Part I. Lecture Notes in Business Information Processing, vol. 353, pp. 162–175. Springer (2019). https://doi.org/10.1007/978-3-030-20485-3_13

[291] Laadhar, A., Ghozzi, F., Megdiche, I., Ravat, F., Teste, O., Gargouri, F.: Pomap++ results for OAEI 2019: Fully automated machine learning approach for ontology matching. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 14th International Workshop on Ontology Matching co-located with the 18th International Semantic Web Conference (ISWC 2019), Auckland, New

Zealand, October 26, 2019. CEUR Workshop Proceedings, vol. 2536, pp. 169–174. CEUR-WS.org (2019), http://ceur-ws.org/Vol-2536/oaei19_paper13.pdf

[292] Lachtim, F.A., de Carvalho Moura, A.M., Cavalcanti, M.C.: Ontology matching for dynamic publication in semantic portals. J. Braz. Comput. Soc. **15**(1), 27–43 (2009). https://doi.org/10.1590/S0104-65002009000100004

[293] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: ALBERT: A lite BERT for self-supervised learning of language representations. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net (2020), https://openreview.net/forum?id=H1eA7AEtvS

[294] Lassila, O., McGuinness, D.: The role of frame-based representation on the semantic web. Linköping Electronic Articles in Computer and Information Science **6**(5), 2001 (2001)

[295] Lavrac, N., Skrlj, B., Robnik-Sikonja, M.: Propositionalization and embeddings: two sides of the same coin. Mach. Learn. **109**(7), 1465–1507 (2020). https://doi.org/10.1007/s10994-020-05890-8

[296] Le, B.T., Dieng-Kuntz, R., Gandon, F.: On ontology matching problems - for building a corporate semantic web in a multi-communities organization. In: ICEIS 2004, Proceedings of the 6th International Conference on Enterprise Information Systems, Porto, Portugal, April 14-17, 2004. pp. 236–243 (2004)

[297] Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014. JMLR Workshop and Conference Proceedings, vol. 32, pp. 1188–1196. JMLR.org (2014), http://proceedings.mlr.press/v32/le14.html

[298] Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J.: Biobert: a pre-trained biomedical language representation model for biomedical text mining. Bioinform. **36**(4), 1234–1240 (2020). https://doi.org/10.1093/bioinformatics/btz682

[299] Lee, M.D., Pincombe, B., Welsh, M.: An Empirical Evaluation of Models of Text Document Similarity. Proceedings of the Annual Meeting of the

Cognitive Science Society **7**(7), 1254–1529 (2005), https://hdl.handle.net/2440/28910

[300] Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., Bizer, C.: Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web **6**(2), 167–195 (2015). https://doi.org/10.3233/SW-140134

[301] Lehmberg, O.: Web table integration and profiling for knowledge base augmentation. Ph.D. thesis, University of Mannheim, Germany (2019), https://madoc.bib.uni-mannheim.de/52346/

[302] Lenat, D., Guha, R.: Building large knowledge-based systems : representation and inference in the Cyc project. Addison-Wesley Pub. Co, Reading, Mass (1989)

[303] Lera, I., Juiz, C., Puigjaner, R.: Quick ontology mapping algorithm for distributed environments. In: Arabnia, H.R., Marsh, A. (eds.) Proceedings of the 2008 International Conference on Semantic Web & Web Services, SWWS 2008, July 14-17, 2008, Las Vegas, Nevada, USA. pp. 107–113. CSREA Press (2008)

[304] Levenshtein, V.I., et al.: Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet physics doklady. vol. 10, pp. 707–710. Soviet Union (1966)

[305] Li, G.: Improving biomedical ontology matching using domain-specific word embeddings. In: Emrouznejad, A., Chou, J.R. (eds.) CSAE 2020: The 4th International Conference on Computer Science and Application Engineering, Sanya, China, October 20-22, 2020. pp. 120:1–120:5. ACM (2020). https://doi.org/10.1145/3424978.3425102

[306] Li, S., Hu, H., Hu, X.: An ontology mapping method based on tree structure. In: 2006 International Conference on Semantics, Knowledge and Grid (SKG 2006), 1-3 November 2006, Guilin, China. p. 87. IEEE Computer Society (2006). https://doi.org/10.1109/SKG.2006.26

[307] Li, Y., Li, J., Zhang, D., Tang, J.: Result of ontology alignment with rimom at oaei'06. In: Shvaiko, P., Euzenat, J., Noy, N.F., Stuckenschmidt, H., Benjamins, V.R., Uschold, M. (eds.) Proceedings of the 1st International Workshop on Ontology Matching (OM-2006) Collocated with the 5th International Semantic Web Conference (ISWC-2006), Athens, Georgia, USA,

November 5, 2006. CEUR Workshop Proceedings, vol. 225. CEUR-WS.org (2006), `http://ceur-ws.org/Vol-225/paper17.pdf`

[308] Li, Y., Zhong, Q., Li, J., Tang, J.: Result of ontology alignment with rimom at oaei'07. In: Shvaiko, P., Euzenat, J., Giunchiglia, F., He, B. (eds.) Proceedings of the 2nd International Workshop on Ontology Matching (OM-2007) Collocated with the 6th International Semantic Web Conference (ISWC-2007) and the 2nd Asian Semantic Web Conference (ASWC-2007), Busan, Korea, November 11, 2007. CEUR Workshop Proceedings, vol. 304. CEUR-WS.org (2007), `http://ceur-ws.org/Vol-304/paper20.pdf`

[309] Li, Y., Li, J., Suhara, Y., Wang, J., Hirota, W., Tan, W.: Deep entity matching: Challenges and opportunities. ACM J. Data Inf. Qual. **13**(1), 1:1–1:17 (2021). `https://doi.org/10.1145/3431816`

[310] Liao, J., Liu, X., Zhu, X., Xu, T., Wang, J., Sun, H.: Ontology alignment by combining lexical analysis with consequences from reasoners. In: Goble, C.A., Chen, P.P., Zhang, J. (eds.) 2012 IEEE 19th International Conference on Web Services, Honolulu, HI, USA, June 24-29, 2012. pp. 288–294. IEEE Computer Society (2012). `https://doi.org/10.1109/ICWS.2012.57`

[311] Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J.E., Stoica, I.: Tune: A research platform for distributed model selection and training. CoRR **abs/1807.05118** (2018), `http://arxiv.org/abs/1807.05118`

[312] Lima, B., Faria, D., Couto, F.M., Cruz, I.F., Pesquita, C.: OAEI 2020 results for AML and AMLC. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 15th International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020), Virtual conference (originally planned to be in Athens, Greece), November 2, 2020. CEUR Workshop Proceedings, vol. 2788, pp. 154–160. CEUR-WS.org (2020), `http://ceur-ws.org/Vol-2788/oaei20_paper3.pdf`

[313] Lin, D.: An information-theoretic definition of similarity. In: Shavlik, J.W. (ed.) Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998), Madison, Wisconsin, USA, July 24-27, 1998. pp. 296–304. Morgan Kaufmann (1998)

[314] Lin, F., Krizhanovsky, A.: Multilingual ontology matching based on wiktionary data accessible via SPARQL endpoint. In: Kalinichenko, L.A., Sychev, A., Stupnikov, S.A. (eds.) Proceedings of the 13th All-Russian Scientific Conference "Digital libraries: Advanced Methods and Technologies,

Digital Collections", RCDL 2011, Voronezh, Russia, October 19-22, 2011. CEUR Workshop Proceedings, vol. 803, pp. 1–8. CEUR-WS.org (2011), `http://ceur-ws.org/Vol-803/paper1.pdf`

[315] Lin, F., Sandkuhl, K.: A survey of exploiting wordnet in ontology matching. In: Bramer, M. (ed.) Artificial Intelligence in Theory and Practice II, IFIP 20th World Computer Congress, TC 12: IFIP AI 2008 Stream, September 7-10, 2008, Milano, Italy. IFIP, vol. 276, pp. 341–350. Springer (2008). `https://doi.org/10.1007/978-0-387-09695-7_33`

[316] Lin, W., Hu, C., Li, Y., Cheng, X.: Similarity-based ontology mapping in material science domain. In: Krempels, K., Stocker, A. (eds.) WEBIST 2013 - Proceedings of the 9th International Conference on Web Information Systems and Technologies, Aachen, Germany, 8-10 May, 2013. pp. 265–269. SciTePress (2013)

[317] Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Bonet, B., Koenig, S. (eds.) Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA. pp. 2181–2187. AAAI Press (2015), `http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9571`

[318] Lindberg, D.A., Humphreys, B.L., McCray, A.T.: The unified medical language system. Yearbook of Medical Informatics **2**(01), 41–51 (1993)

[319] Ling, W., Dyer, C., Black, A.W., Trancoso, I.: Two/too simple adaptations of word2vec for syntax problems. In: Mihalcea, R., Chai, J.Y., Sarkar, A. (eds.) NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015. pp. 1299–1304. The Association for Computational Linguistics (2015). `https://doi.org/10.3115/v1/n15-1142`

[320] Liu, B.: Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. Data-centric systems and applications, Springer, Heidelberg ; New York, 2 edn. (2011)

[321] Liu, J., Zheng, B., Luo, L., Zhou, J., Zhang, Y., Yu, Z.: Ontology representation and mapping of common fuzzy knowledge. Neurocomputing **215**, 184–195 (2016). `https://doi.org/10.1016/j.neucom.2016.01.114`

[322] Liu, L., Yang, F., Zhang, P., Wu, J., Hu, L.: Svm-based ontology matching approach. Int. J. Autom. Comput. **9**(3), 306–314 (2012). https://doi.org/10.1007/s11633-012-0649-x

[323] Liu, X., Cheng, B., Liao, J., Barnaghi, P.M., Wan, L., Wang, J.: OMI-DL: an ontology matching framework. IEEE Trans. Serv. Comput. **9**(4), 580–593 (2016). https://doi.org/10.1109/TSC.2015.2410794

[324] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized BERT pretraining approach. CoRR **abs/1907.11692** (2019), http://arxiv.org/abs/1907.11692

[325] Löbner, S.: Semantik : Eine Einführung. De Gruyter, Berlin, Germany Boston, Massachusetts (2015)

[326] Locoro, A., David, J., Euzenat, J.: Context-based matching: Design of a flexible framework and experiment. J. Data Semant. **3**(1), 25–46 (2014). https://doi.org/10.1007/s13740-013-0019-z

[327] Lohmann, S., Negru, S., Haag, F., Ertl, T.: Visualizing ontologies with VOWL. Semantic Web **7**(4), 399–419 (2016). https://doi.org/10.3233/SW-150200

[328] Lu, Z.: ontomatch: A probabilistic architecture for ontology matching. In: Peng, Y., Kou, G., Ko, F.I.S., Zeng, Y., Kwack, K. (eds.) Proceedings of the 3rd International Conference on Information Sciences and Interaction Sciences (ICIS 2010), Chengdu, China, June 23-25, 2010. pp. 174–180. IEEE (2010). https://doi.org/10.1109/ICICIS.2010.5534737

[329] Lütke, A.: Anygraphmatcher submission to the OAEI knowledge graph challenge 2019. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 14th International Workshop on Ontology Matching co-located with the 18th International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 26, 2019. CEUR Workshop Proceedings, vol. 2536, pp. 86–93. CEUR-WS.org (2019), http://ceur-ws.org/Vol-2536/oaei19_paper1.pdf

[330] Lv, Q., Jiang, C., Li, H.: Solving ontology meta-matching problem through an evolutionary algorithm with approximate evaluation indicators and adaptive selection pressure. IEEE Access **9**, 3046–3064 (2021). https://doi.org/10.1109/ACCESS.2020.3047875

[331] Löbner, S.: Semantik: Eine Einführung, chap. 2, pp. 21–47. De Gruyter Studienbuch, de Gruyter Mouton, Berlin, 2nd edition edn. (2015)

[332] Mansar, Y., Kang, J., Maarouf, I.E.: The finsim-2 2021 shared task: Learning semantic similarities for the financial domain. In: Leskovec, J., Grobelnik, M., Najork, M., Tang, J., Zia, L. (eds.) Companion of The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021. pp. 288–292. ACM / IW3C2 (2021). `https://doi.org/10.1145/3442442.3451381`

[333] Mao, M., Peng, Y., Spring, M.: Ontology mapping: as a binary classification problem. Concurr. Comput. Pract. Exp. **23**(9), 1010–1025 (2011). `https://doi.org/10.1002/cpe.1633`

[334] Mascardi, V., Locoro, A., Rosso, P.: Automatic ontology matching via upper ontologies: A systematic evaluation. IEEE Trans. Knowl. Data Eng. **22**(5), 609–623 (2010). `https://doi.org/10.1109/TKDE.2009.154`

[335] McCrae, J.P., Aguado de Cea, G., Buitelaar, P., Cimiano, P., Declerck, T., Gómez-Pérez, A., Gracia, J., Hollink, L., Montiel-Ponsoda, E., Spohr, D., Wunner, T.: Interchanging lexical resources on the semantic web. Lang. Resour. Evaluation **46**(4), 701–719 (2012). `https://doi.org/10.1007/s10579-012-9182-3`

[336] Meilicke, C.: Alignment incoherence in ontology matching. Ph.D. thesis, University of Mannheim (2011), `https://ub-madoc.bib.uni-mannheim.de/29351`

[337] Meilicke, C., Garcia-Castro, R., Freitas, F., van Hage, W.R., Montiel-Ponsoda, E., de Azevedo, R.R., Stuckenschmidt, H., Sváb-Zamazal, O., Svátek, V., Tamilin, A., dos Santos, C.T., Wang, S.: Multifarm: A benchmark for multilingual ontology matching. J. Web Semant. **15**, 62–68 (2012). `https://doi.org/10.1016/j.websem.2012.04.001`

[338] Meilicke, C., Stuckenschmidt, H.: Analyzing mapping extraction approaches. In: Shvaiko, P., Euzenat, J., Giunchiglia, F., He, B. (eds.) Proceedings of the 2nd International Workshop on Ontology Matching (OM-2007) Collocated with the 6th International Semantic Web Conference (ISWC-2007) and the 2nd Asian Semantic Web Conference (ASWC-2007), Busan, Korea, November 11, 2007. CEUR Workshop Proceedings, vol. 304. CEUR-WS.org (2007), `http://ceur-ws.org/Vol-304/paper3.pdf`

[339] Melo, A., Paulheim, H.: Synthesizing knowledge graphs for link and type prediction benchmarking. In: Blomqvist, E., Maynard, D., Gangemi, A.,

Hoekstra, R., Hitzler, P., Hartig, O. (eds.) The Semantic Web - 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28 - June 1, 2017, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10249, pp. 136–151 (2017). `https://doi.org/10.1007/978-3-319-58068-5_9`

[340] Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: Dbpedia spotlight: shedding light on the web of documents. In: Ghidini, C., Ngomo, A.N., Lindstaedt, S.N., Pellegrini, T. (eds.) Proceedings the 7th International Conference on Semantic Systems, I-SEMANTICS 2011, Graz, Austria, September 7-9, 2011. pp. 1–8. ACM International Conference Proceeding Series, ACM (2011). `https://doi.org/10.1145/2063518.2063519`

[341] Merriam-Webster: Semantic, `http://web.archive.org/web/20220502040648/https://www.merriam-webster.com/dictionary/semantic`, accessed: 2022-05-02

[342] Merriam-Webster: Syntax. `http://web.archive.org/web/20220412012821/https://www.merriam-webster.com/dictionary/syntax`, accessed: 2022-04-12

[343] Meyer, C.M., Gurevych, I.: Worth its weight in gold or yet another resource - A comparative study of wiktionary, openthesaurus and germanet. In: Gelbukh, A.F. (ed.) Computational Linguistics and Intelligent Text Processing, 11th International Conference, CICLing 2010, Iasi, Romania, March 21-27, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6008, pp. 38–49. Springer (2010). `https://doi.org/10.1007/978-3-642-12116-6_4`

[344] Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Bengio, Y., LeCun, Y. (eds.) 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings (2013), `http://arxiv.org/abs/1301.3781`

[345] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C.J.C., Bottou, L., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States. pp. 3111–3119 (2013), `https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html`

[346] Miller, G.A.: Wordnet: A lexical database for english. Commun. ACM **38**(11), 39–41 (1995). `https://doi.org/10.1145/219717.219748`

[347] Moghaddam, F.B., Draschner, C., Lehmann, J., Jabeen, H.: Literal2feature: An automatic scalable RDF graph feature extractor. In: Alam, M., Groth, P., de Boer, V., Pellegrini, T., Pandit, H.J., Montiel-Ponsoda, E., Rodríguez-Doncel, V., McGillivray, B., Meroño-Peñuela, A. (eds.) Further with Knowledge Graphs - Proceedings of the 17th International Conference on Semantic Systems, SEMANTiCS 2017, Amsterdam, The Netherlands, September 6-9, 2021. Studies on the Semantic Web, vol. 53, pp. 74–88. IOS Press (2021). `https://doi.org/10.3233/SSW210036`

[348] Mohammadi, M., Atashin, A.A., Hofman, W., Tan, Y.: Comparison of ontology alignment systems across single matching task via the mcnemar's test. ACM Trans. Knowl. Discov. Data **12**(4), 51:1–51:18 (2018). `https://doi.org/10.1145/3193573`

[349] Mohammadi, M., Hofman, W., Tan, Y.: SANOM results for OAEI 2018. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018. CEUR Workshop Proceedings, vol. 2288, pp. 205–209. CEUR-WS.org (2018), `http://ceur-ws.org/Vol-2288/oaei18_paper14.pdf`

[350] Mongiello, M., Totaro, R.: Automatic ontology mapping for agent communication in an e-commerce environment. In: Bauknecht, K., Pröll, B., Werthner, H. (eds.) E-Commerce and Web Technologies: 6th International Conference, EC-Web 2005, Copenhagen, Denmark, August 23-26, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3590, pp. 21–30. Springer (2005). `https://doi.org/10.1007/11545163_3`

[351] Monych, M., Portisch, J., Hladik, M., Paulheim, H.: Deskmatcher. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 15th International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020), Virtual conference (originally planned to be in Athens, Greece), November 2, 2020. CEUR Workshop Proceedings, vol. 2788, pp. 181–186. CEUR-WS.org (2020), `http://ceur-ws.org/Vol-2788/oaei20_paper7.pdf`

[352] Mountasser, I., Ouhbi, B., Frikh, B.: Hybrid large-scale ontology matching strategy on big data environment. In: Anderst-Kotsis, G. (ed.) Proceedings of the 18th International Conference on Information Integration and Web-based Applications and Services, iiWAS 2016, Singapore, November 28-30, 2016. pp. 282–287. ACM (2016). `https://doi.org/10.1145/3011141.3011185`

[353] Mukkala, L., Arvo, J., Lehtonen, T., Knuutila, T., et al.: Current state of ontology matching. A Survey of Ontology and Schema Matching. University of Turku Technical Reports **4** (2015)

[354] Müller, A.C., Paulheim, H.: Towards combining ontology matchers via anomaly detection. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Proceedings of the 10th International Workshop on Ontology Matching collocated with the 14th International Semantic Web Conference (ISWC 2015), Bethlehem, PA, USA, October 12, 2015. CEUR Workshop Proceedings, vol. 1545, pp. 40–44. CEUR-WS.org (2015), `http://ceur-ws.org/Vol-1545/om2015_TSpaper2.pdf`

[355] Mungall, C.J., Torniai, C., Gkoutos, G.V., Lewis, S.E., Haendel, M.A.: Uberon, an integrative multi-species anatomy ontology. Genome biology **13**(1), 1–20 (2012), `https://www.doi.org/10.1186/gb-2012-13-1-r5`

[356] Murphy, K.P.: Machine Learning: A Probabilistic Perspective, chap. 5, pp. 149–189. Adaptive Computation and Machine Learning Series, MIT Press, Cambridge, MA (2012)

[357] Murphy, L.: Semantic relations and the lexicon : antonymy, synonymy, and other paradigms. Cambridge University Press, Cambridge, UK New York (2003), `https://www.doi.org/10.1017/CBO9780511486494`

[358] Murphy, L.: Semantic relations and the lexicon : antonymy, synonymy, and other paradigms. Cambridge University Press, Cambridge, UK New York (2003), `https://www.doi.org/10.1017/CBO9780511486494`

[359] Nagy, M., Vargas-Vera, M., Motta, E.: Dssim-ontology mapping with uncertainty. In: Shvaiko, P., Euzenat, J., Noy, N.F., Stuckenschmidt, H., Benjamins, V.R., Uschold, M. (eds.) Proceedings of the 1st International Workshop on Ontology Matching (OM-2006) Collocated with the 5th International Semantic Web Conference (ISWC-2006), Athens, Georgia, USA,

November 5, 2006. CEUR Workshop Proceedings, vol. 225. CEUR-WS.org (2006), http://ceur-ws.org/Vol-225/paper10.pdf

[360] Nagy, M., Vargas-Vera, M., Motta, E.: Uncertainty handling in the context of ontology mapping for question-answering. In: Honavar, V.G., Finin, T. (eds.) Semantic Web for Collaborative Knowledge Acquisition, Papers from the 2006 AAAI Fall Symposium, Washington, DC, USA, October 13-15, 2006. AAAI Technical Report, vol. FS-06-06, pp. 48–55. AAAI Press (2006), https://www.aaai.org/Library/Symposia/Fall/2006/fs06-06-007.php

[361] Nations, D.: Is Web 3.0 Really a Thing? (2018), https://www.lifewire.com/what-is-web-3-0-3486623, accessed: 2022-05-10

[362] Navigli, R., Ponzetto, S.P.: Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artif. Intell. **193**, 217–250 (2012). https://doi.org/10.1016/j.artint.2012.07.001

[363] Neutel, S., de Boer, M.H.T.: Towards automatic ontology alignment using BERT. In: Martin, A., Hinkelmann, K., Fill, H., Gerber, A., Lenat, D., Stolle, R., van Harmelen, F. (eds.) Proceedings of the AAAI 2021 Spring Symposium on Combining Machine Learning and Knowledge Engineering (AAAI-MAKE 2021), Stanford University, Palo Alto, California, USA, March 22-24, 2021. CEUR Workshop Proceedings, vol. 2846. CEUR-WS.org (2021), http://ceur-ws.org/Vol-2846/paper28.pdf

[364] Ngo, D., Bellahsene, Z.: YAM++ : A multi-strategy based approach for ontology matching task. In: ten Teije, A., Völker, J., Handschuh, S., Stuckenschmidt, H., d'Aquin, M., Nikolov, A., Aussenac-Gilles, N., Hernandez, N. (eds.) Knowledge Engineering and Knowledge Management - 18th International Conference, EKAW 2012, Galway City, Ireland, October 8-12, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7603, pp. 421–425. Springer (2012). https://doi.org/10.1007/978-3-642-33876-2_38

[365] Ngo, D., Bellahsene, Z.: YAM++ results for OAEI 2012. In: Shvaiko, P., Euzenat, J., Kementsietsidis, A., Mao, M., Noy, N.F., Stuckenschmidt, H. (eds.) Proceedings of the 7th International Workshop on Ontology Matching, Boston, MA, USA, November 11, 2012. CEUR Workshop Proceedings, vol. 946. CEUR-WS.org (2012), http://ceur-ws.org/Vol-946/oaei12_paper16.pdf

[366] Ngo, D., Bellahsene, Z., Coletta, R.: A generic approach for combining linguistic and context profile metrics in ontology matching. In: Meersman, R., Dillon, T.S., Herrero, P., Kumar, A., Reichert, M., Qing, L., Ooi, B.C., Damiani, E., Schmidt, D.C., White, J., Hauswirth, M., Hitzler, P., Mohania, M.K. (eds.) On the Move to Meaningful Internet Systems: OTM 2011 - Confederated International Conferences: CoopIS, DOA-SVI, and ODBASE 2011, Hersonissos, Crete, Greece, October 17-21, 2011, Proceedings, Part II. Lecture Notes in Computer Science, vol. 7045, pp. 800–807. Springer (2011). `https://doi.org/10.1007/978-3-642-25106-1_27`

[367] Ngo, D., Bellahsene, Z., Coletta, R.: YAM++ results for OAEI 2011. In: Shvaiko, P., Euzenat, J., Heath, T., Quix, C., Mao, M., Cruz, I.F. (eds.) Proceedings of the 6th International Workshop on Ontology Matching, Bonn, Germany, October 24, 2011. CEUR Workshop Proceedings, vol. 814. CEUR-WS.org (2011), `http://ceur-ws.org/Vol-814/oaei11_paper17.pdf`

[368] Ngomo, A.C.N., Röder, M.: Hobbit: Holistic benchmarking for big linked data. ERCIM News **2016**(105), 46–67 (2016), `https://svn.aksw.org/papers/2016/ESWC_HOBBIT_EUNetworking/public.pdf`

[369] Nguyen, T.T.A., Conrad, S.: Ontology matching using multiple similarity measures. In: Fred, A.L.N., Dietz, J.L.G., Aveiro, D., Liu, K., Filipe, J. (eds.) KDIR 2015 - Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, part of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015), Volume 1, Lisbon, Portugal, November 12-14, 2015. pp. 603–611. SciTePress (2015). `https://doi.org/10.5220/0005615606030611`

[370] Nickel, M., Tresp, V., Kriegel, H.: A three-way model for collective learning on multi-relational data. In: Getoor, L., Scheffer, T. (eds.) Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011. pp. 809–816. Omnipress (2011), `https://icml.cc/2011/papers/438_icmlpaper.pdf`

[371] Niedbala, S.: Owl-ctxmatch in the OAEI 2006 alignment contest. In: Shvaiko, P., Euzenat, J., Noy, N.F., Stuckenschmidt, H., Benjamins, V.R., Uschold, M. (eds.) Proceedings of the 1st International Workshop on Ontology Matching (OM-2006) Collocated with the 5th International Semantic Web Conference (ISWC-2006), Athens, Georgia, USA, November 5,

2006. CEUR Workshop Proceedings, vol. 225. CEUR-WS.org (2006), http://ceur-ws.org/Vol-225/paper15.pdf

[372] Niles, I., Pease, A.: Towards a Standard Upper Ontology. In: Proceedings of the international conference on Formal Ontology in Information Systems - Volume 2001. pp. 2–9. ACM Press (2001)

[373] Niles, I., Pease, A.: Towards a standard upper ontology. In: 2nd International Conference on Formal Ontology in Information Systems, FOIS 2001, Ogunquit, Maine, USA, October 17-19, 2001, Proceedings. pp. 2–9. ACM (2001). https://doi.org/10.1145/505168.505170

[374] Nkisi-Orji, I., Wiratunga, N., Massie, S., Hui, K., Heaven, R.: Ontology alignment based on word embedding and random forest classification. In: ECML PKDD 2018, Proceedings, Part I. pp. 557–572 (2018). https://doi.org/10.1007/978-3-030-10925-7_34

[375] Oberle, D.: How ontologies benefit enterprise applications. Semantic Web **5**(6), 473–491 (2014). https://doi.org/10.3233/SW-130114

[376] Ochieng, P., Kyanda, S.: A statistically-based ontology matching tool. Distributed Parallel Databases **36**(1), 195–217 (2018). https://doi.org/10.1007/s10619-017-7206-0

[377] Osman, I., Yahia, S.B., Diallo, G.: Ontology integration: Approaches and challenging issues. Inf. Fusion **71**, 38–63 (2021). https://doi.org/10.1016/j.inffus.2021.01.007

[378] Otero-Cerdeira, L., Rodríguez-Martínez, F.J., Gómez-Rodríguez, A.: Ontology matching: A literature review. Expert Syst. Appl. **42**(2), 949–971 (2015). https://doi.org/10.1016/j.eswa.2014.08.032

[379] Pan, R., Ding, Z., Yu, Y., Peng, Y.: A bayesian network approach to ontology mapping. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) The Semantic Web - ISWC 2005, 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3729, pp. 563–577. Springer (2005). https://doi.org/10.1007/11574620_41

[380] Park, S., Kim, W.: Ontology mapping between heterogeneous product taxonomies in an electronic commerce environment. Int. J. Electron. Commer. **12**(2), 69–87 (2007). https://doi.org/10.2753/JEC1086-4415120203

[381] Park, S., Kim, W., Lee, S., Bang, S.: Product matching through ontology mapping in comparison shopping. In: Kotsis, G., Taniar, D., Pardede, E., Ibrahim, I.K. (eds.) iiWAS'2006 - The Eighth International Conference on Information Integration and Web-based Applications Services, 4-6 December 2006, Yogyakarta, Indonesia. books@ocg.at, vol. 214, pp. 39–49. Austrian Computer Society (2006)

[382] Paulheim, H.: Ontology-based application integration. Springer Science & Business Media (2011), `https://doi.org/10.1007/978-1-4614-1430-8`

[383] Paulheim, H.: Wesee-match results for OEAI 2012. In: Shvaiko, P., Euzenat, J., Kementsietsidis, A., Mao, M., Noy, N.F., Stuckenschmidt, H. (eds.) Proceedings of the 7th International Workshop on Ontology Matching, Boston, MA, USA, November 11, 2012. CEUR Workshop Proceedings, vol. 946. CEUR-WS.org (2012), `http://ceur-ws.org/Vol-946/oaei12_paper14.pdf`

[384] Paulheim, H.: Knowledge graph refinement: A survey of approaches and evaluation methods. Semantic Web **8**(3), 489–508 (2017). `https://doi.org/10.3233/SW-160218`

[385] Paulheim, H., Hertling, S.: Wesee-match results for OAEI 2013. In: Shvaiko, P., Euzenat, J., Srinivas, K., Mao, M., Jiménez-Ruiz, E. (eds.) Proceedings of the 8th International Workshop on Ontology Matching co-located with the 12th International Semantic Web Conference (ISWC 2013), Sydney, Australia, October 21, 2013. CEUR Workshop Proceedings, vol. 1111, pp. 197–202. CEUR-WS.org (2013), `http://ceur-ws.org/Vol-1111/oaei13_paper14.pdf`

[386] Pavlick, E., Rastogi, P., Ganitkevitch, J., Durme, B.V., Callison-Burch, C.: PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers. pp. 425–430. The Association for Computer Linguistics (2015). `https://doi.org/10.3115/v1/p15-2070`

[387] Peeters, R., Bizer, C., Glavas, G.: Intermediate training of BERT for product matching. In: Piai, F., Firmani, D., Crescenzi, V., Angelis, A.D., Dong,

X.L., Mazzei, M., Merialdo, P., Srivastava, D. (eds.) Proceedings of the 2nd International Workshop on Challenges and Experiences from Data Integration to Knowledge Graphs co-located with 46th International Conference on Very Large Data Bases, DI2KG@VLDB 2020, Tokyo, Japan, August 31, 2020. CEUR Workshop Proceedings, vol. 2726. CEUR-WS.org (2020), http://ceur-ws.org/Vol-2726/paper1.pdf

[388] Pellegrino, M.A., Altabba, A., Garofalo, M., Ristoski, P., Cochez, M.: Geval: A modular and extensible evaluation framework for graph embedding techniques. In: Harth, A., Kirrane, S., Ngomo, A.N., Paulheim, H., Rula, A., Gentile, A.L., Haase, P., Cochez, M. (eds.) The Semantic Web - 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31-June 4, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12123, pp. 565–582. Springer (2020). https://doi.org/10.1007/978-3-030-49461-2_33

[389] Pellegrino, M.A., Cochez, M., Garofalo, M., Ristoski, P.: A configurable evaluation framework for node embedding techniques. In: Hitzler, P., Kirrane, S., Hartig, O., de Boer, V., Vidal, M., Maleshkova, M., Schlobach, S., Hammar, K., Lasierra, N., Stadtmüller, S., Hose, K., Verborgh, R. (eds.) The Semantic Web: ESWC 2019 Satellite Events - ESWC 2019 Satellite Events, Portorož, Slovenia, June 2-6, 2019, Revised Selected Papers. Lecture Notes in Computer Science, vol. 11762, pp. 156–160. Springer (2019). https://doi.org/10.1007/978-3-030-32327-1_31

[390] Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Moschitti, A., Pang, B., Daelemans, W. (eds.) Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL. pp. 1532–1543. ACL (2014). https://doi.org/10.3115/v1/d14-1162

[391] Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. In: Macskassy, S.A., Perlich, C., Leskovec, J., Wang, W., Ghani, R. (eds.) The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014. pp. 701–710. ACM (2014). https://doi.org/10.1145/2623330.2623732

[392] Petrov, P., Krachunov, M., Vassilev, D.: A semi-automated approach for anatomical ontology mapping. J. Integr. Bioinform. **10**(2) (2013). https://doi.org/10.2390/biecoll-jib-2013-221

[393] Pirrò, G., Ruffolo, M., Talia, D.: An algorithm for discovering ontology mappings in P2P systems. In: Lovrek, I., Howlett, R.J., Jain, L.C. (eds.) Knowledge-Based Intelligent Information and Engineering Systems, 12th International Conference, KES 2008, Zagreb, Croatia, September 3-5, 2008, Proceedings, Part II. Lecture Notes in Computer Science, vol. 5178, pp. 631–641. Springer (2008). `https://doi.org/10.1007/978-3-540-85565-1_78`

[394] Pirrò, G., Talia, D.: Ufome: A user friendly ontology mapping environment. In: Semeraro, G., Sciascio, E.D., Morbidoni, C., Stoermer, H. (eds.) Proceedings of the 4th Italian Semantic Web Workshop, Dipartimento di Informatica - Universita' degli Studi di Bari - Italy, 18-20 December, 2007. CEUR Workshop Proceedings, vol. 314. CEUR-WS.org (2007), `http://ceur-ws.org/Vol-314/22.pdf`

[395] Plag, I., Lappe, S., Braun, M., Schramm, M.: Introduction to English Linguistics, chap. 5, pp. 145–179. De Gruyter Mouton textbook, De Gruyter Mouton, Berlin, third, revised and enlarged edition edn. (2015)

[396] Portisch, J.: Towards matching of domain-specific schemas using general-purpose external background knowledge. In: Harth, A., Presutti, V., Troncy, R., Acosta, M., Polleres, A., Fernández, J.D., Parreira, J.X., Hartig, O., Hose, K., Cochez, M. (eds.) The Semantic Web: ESWC 2020 Satellite Events - ESWC 2020 Satellite Events, Heraklion, Crete, Greece, May 31 - June 4, 2020, Revised Selected Papers. Lecture Notes in Computer Science, vol. 12124, pp. 270–279. Springer (2020). `https://doi.org/10.1007/978-3-030-62327-2_42`

[397] Portisch, J., Costa, G., Stefani, K., Kreplin, K., Hladik, M., Paulheim, H.: Ontology matching through absolute orientation of embedding spaces. CoRR **abs/2204.04040** (2022). `https://doi.org/10.48550/arXiv.2204.04040`

[398] Portisch, J., Fallatah, O., Neumaier, S., Jaradeh, M.Y., Polleres, A.: Challenges of linking organizational information in open government data to knowledge graphs. In: Keet, C.M., Dumontier, M. (eds.) Knowledge Engineering and Knowledge Management - 22nd International Conference, EKAW 2020, Bolzano, Italy, September 16-20, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12387, pp. 271–286. Springer (2020). `https://doi.org/10.1007/978-3-030-61244-3_19`

[399] Portisch, J., Heist, N., Paulheim, H.: Knowledge graph embedding for data mining vs. knowledge graph embedding for link prediction - two sides of the same coin? Semantic Web **13**(3), 399–422 (2022). `https://doi.org/10.3233/SW-212892`

[400] Portisch, J., Hertling, S., Paulheim, H.: Visual analysis of ontology matching results with the MELT dashboard. In: Harth, A., Presutti, V., Troncy, R., Acosta, M., Polleres, A., Fernández, J.D., Parreira, J.X., Hartig, O., Hose, K., Cochez, M. (eds.) The Semantic Web: ESWC 2020 Satellite Events - ESWC 2020 Satellite Events, Heraklion, Crete, Greece, May 31 - June 4, 2020, Revised Selected Papers. Lecture Notes in Computer Science, vol. 12124, pp. 186–190. Springer (2020). `https://doi.org/10.1007/978-3-030-62327-2_32`

[401] Portisch, J., Hladik, M., Paulheim, H.: Evaluating ontology matchers on real-world financial services data models. In: Alam, M., Usbeck, R., Pellegrini, T., Sack, H., Sure-Vetter, Y. (eds.) Proceedings of the Posters and Demo Track of the 15th International Conference on Semantic Systems co-located with 15th International Conference on Semantic Systems (SEMANTiCS 2019), Karlsruhe, Germany, September 9th - to - 12th, 2019. CEUR Workshop Proceedings, vol. 2451. CEUR-WS.org (2019), `http://ceur-ws.org/Vol-2451/paper-22.pdf`

[402] Portisch, J., Hladik, M., Paulheim, H.: Wiktionary matcher. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 14th International Workshop on Ontology Matching co-located with the 18th International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 26, 2019. CEUR Workshop Proceedings, vol. 2536, pp. 181–188. CEUR-WS.org (2019), `http://ceur-ws.org/Vol-2536/oaei19_paper15.pdf`

[403] Portisch, J., Hladik, M., Paulheim, H.: Alod2vec matcher results for OAEI 2020. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 15th International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020), Virtual conference (originally planned to be in Athens, Greece), November 2, 2020. CEUR Workshop Proceedings, vol. 2788, pp. 147–153. CEUR-WS.org (2020), `http://ceur-ws.org/Vol-2788/oaei20_paper2.pdf`

[404] Portisch, J., Hladik, M., Paulheim, H.: Kgvec2go - knowledge graph embeddings as a service. In: Calzolari, N., Béchet, F., Blache, P., Choukri,

K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S. (eds.) Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020. pp. 5641–5647. European Language Resources Association (2020), https://aclanthology.org/2020.lrec-1.692/

[405] Portisch, J., Hladik, M., Paulheim, H.: Rdf2vec light - A lightweight approachfor knowledge graph embeddings. In: Taylor, K.L., Gonçalves, R.S., Lécué, F., Yan, J. (eds.) Proceedings of the ISWC 2020 Demos and Industry Tracks: From Novel Ideas to Industrial Practice co-located with 19th International Semantic Web Conference (ISWC 2020), Globally online, November 1-6, 2020 (UTC). CEUR Workshop Proceedings, vol. 2721, pp. 79–84. CEUR-WS.org (2020), http://ceur-ws.org/Vol-2721/paper520.pdf

[406] Portisch, J., Hladik, M., Paulheim, H.: Background knowledge in schema matching: Strategy vs. data. In: Hotho, A., Blomqvist, E., Dietze, S., Fokoue, A., Ding, Y., Barnaghi, P.M., Haller, A., Dragoni, M., Alani, H. (eds.) The Semantic Web - ISWC 2021 - 20th International Semantic Web Conference, ISWC 2021, Virtual Event, October 24-28, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12922, pp. 287–303. Springer (2021). https://doi.org/10.1007/978-3-030-88361-4_17

[407] Portisch, J., Hladik, M., Paulheim, H.: Finmatcher at finsim-2: Hypernym detection in the financial services domain using knowledge graphs. In: Leskovec, J., Grobelnik, M., Najork, M., Tang, J., Zia, L. (eds.) Companion of The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021. pp. 293–297. ACM / IW3C2 (2021). https://doi.org/10.1145/3442442.3451382

[408] Portisch, J., Hladik, M., Paulheim, H.: Background knowledge in ontology matching: A survey. Semantic Web (2022), in print

[409] Portisch, J., Paulheim, H.: Alod2vec matcher. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018. CEUR Workshop Proceedings, vol. 2288, pp. 132–137. CEUR-WS.org (2018), http://ceur-ws.org/Vol-2288/oaei18_paper3.pdf

[410] Portisch, J., Paulheim, H.: Wiktionary matcher results for OAEI 2020. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn,

C. (eds.) Proceedings of the 15th International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020), Virtual conference (originally planned to be in Athens, Greece), November 2, 2020. CEUR Workshop Proceedings, vol. 2788, pp. 225–232. CEUR-WS.org (2020), `http://ceur-ws.org/Vol-2788/oaei20_paper14.pdf`

[411] Portisch, J., Paulheim, H.: Alod2vec matcher results for OAEI 2021. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), Virtual conference, October 25, 2021. CEUR Workshop Proceedings, vol. 3063, pp. 117–123. CEUR-WS.org (2021), `http://ceur-ws.org/Vol-3063/oaei21_paper2.pdf`

[412] Portisch, J., Paulheim, H.: Putting rdf2vec in order. In: Seneviratne, O., Pesquita, C., Sequeda, J., Etcheverry, L. (eds.) Proceedings of the ISWC 2021 Posters, Demos and Industry Tracks: From Novel Ideas to Industrial Practice co-located with 20th International Semantic Web Conference (ISWC 2021), Virtual Conference, October 24-28, 2021. CEUR Workshop Proceedings, vol. 2980. CEUR-WS.org (2021), `http://ceur-ws.org/Vol-2980/paper352.pdf`

[413] Portisch, J., Paulheim, H.: Wiktionary matcher results for OAEI 2021. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), Virtual conference, October 25, 2021. CEUR Workshop Proceedings, vol. 3063, pp. 199–206. CEUR-WS.org (2021), `http://ceur-ws.org/Vol-3063/oaei21_paper14.pdf`

[414] Portisch, J., Paulheim, H.: The DLCC node classification benchmark for analyzing knowledge graph embeddings. In: International Semantic Web Conference (2022), under review

[415] Portisch, J., Paulheim, H.: Rdf2vec variants and DL classes. Semantic Web Journal (2022), to be submitted

[416] Portisch, J., Paulheim, H.: Walk this way! entity walks and property walks for rdf2vec. CoRR **abs/2204.02777** (2022). `https://doi.org/10.48550/arXiv.2204.02777`

[417] Portisch, J.P.: Automatic schema matching utilizing hypernymy relations extracted from the web (2018), https://madoc.bib.uni-mannheim.de/52029/

[418] Postel, H.J.: Die Kölner Phonetik. Ein Verfahren zur Identifizierung von Personenname auf der Grundlage der Gestaltanalyse. IBM-Nachrichten **19**, 925–931 (1969)

[419] Pour, M.A.N., Algergawy, A., Amardeilh, F., Amini, R., Fallatah, O., Faria, D., Fundulaki, I., Harrow, I., Hertling, S., Hitzler, P., Huschka, M., Ibanescu, L., Jiménez-Ruiz, E., Karam, N., Laadhar, A., Lambrix, P., Li, H., Li, Y., Michel, F., Nasr, E., Paulheim, H., Pesquita, C., Portisch, J., Roussey, C., Saveta, T., Shvaiko, P., Splendiani, A., Trojahn, C., Vatascinová, J., Yaman, B., Zamazal, O., Zhou, L.: Results of the ontology alignment evaluation initiative 2021. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), Virtual conference, October 25, 2021. CEUR Workshop Proceedings, vol. 3063, pp. 62–108. CEUR-WS.org (2021), http://ceur-ws.org/Vol-3063/oaei21_paper0.pdf

[420] Princeton University: Princeton University | WordNet | doorhandle (2010), http://wordnetweb.princeton.edu/perl/webwn?o2=&o0=1&o8=1&o1=1&o7=&o5=&o9=&o6=&o3=&o4=&s=doorhandle, accessed: 2022-05-10

[421] Purvis, K.: Fachwörterbuch Versicherungswirtschaft und Recht: Englisch / Deutsch - Deutsch / Englisch. C.H.Beck & Vahlen, Heidelberg, Germany (2021)

[422] Quix, C., Gal, A., Sagi, T., Kensche, D.: An integrated matching system geromesuite and SMB: results for OAEI 2010. In: Shvaiko, P., Euzenat, J., Giunchiglia, F., Stuckenschmidt, H., Mao, M., Cruz, I.F. (eds.) Proceedings of the 5th International Workshop on Ontology Matching (OM-2010), Shanghai, China, November 7, 2010. CEUR Workshop Proceedings, vol. 689. CEUR-WS.org (2010), http://ceur-ws.org/Vol-689/oaei10_paper7.pdf

[423] Quix, C., Geisler, S., Kensche, D., Li, X.: Results of geromesuite for OAEI 2008. In: Shvaiko, P., Euzenat, J., Giunchiglia, F., Stuckenschmidt, H. (eds.) Proceedings of the 3rd International Workshop on Ontology Matching

(OM-2008) Collocated with the 7th International Semantic Web Conference (ISWC-2008), Karlsruhe, Germany, October 26, 2008. CEUR Workshop Proceedings, vol. 431. CEUR-WS.org (2008), `http://ceur-ws.org/Vol-431/oaei08_paper6.pdf`

[424] Quix, C., Geisler, S., Kensche, D., Li, X.: Results of geromesuite for OAEI 2009. In: Shvaiko, P., Euzenat, J., Giunchiglia, F., Stuckenschmidt, H., Noy, N.F., Rosenthal, A. (eds.) Proceedings of the 4th International Workshop on Ontology Matching (OM-2009) collocated with the 8th International Semantic Web Conference (ISWC-2009) Chantilly, USA, October 25, 2009. CEUR Workshop Proceedings, vol. 551. CEUR-WS.org (2009), `http://ceur-ws.org/Vol-551/oaei09_paper6.pdf`

[425] Quix, C., Roy, P., Kensche, D.: Automatic selection of background knowledge for ontology matching. In: Virgilio, R.D., Giunchiglia, F., Tanca, L. (eds.) Proceedings of the International Workshop on Semantic Web Information Management, SWIM 2011, Athens, Greece, June 12, 2011. p. 5. ACM (2011). `https://doi.org/10.1145/1999299.1999304`

[426] Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB J. **10**(4), 334–350 (2001). `https://doi.org/10.1007/s007780100057`

[427] Rajeswari, V., Kavitha, M., Varughese, D.K.: A weighted graph-oriented ontology matching algorithm for enhancing ontology mapping and alignment in semantic web. Soft Comput. **23**(18), 8661–8676 (2019). `https://doi.org/10.1007/s00500-019-04148-3`

[428] Real, F.J.Q., Bella, G., McNeill, F., Bundy, A.: Using domain lexicon and grammar for ontology matching. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 15th International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020), Virtual conference (originally planned to be in Athens, Greece), November 2, 2020. CEUR Workshop Proceedings, vol. 2788, pp. 1–12. CEUR-WS.org (2020), `http://ceur-ws.org/Vol-2788/om2020_LTpaper1.pdf`

[429] Rebstock, M., Fengel, J., Paulheim, H.: Ontologies-based Business Integration. Springer (2008), `https://doi.org/10.1007/978-3-540-75230-1`

[430] Refoufi, A., Benarab, A.: A robust approach to the ontology matching problem. In: Kettani, M.E.Y.E., Djeddi, C., Siddiqi, I., Hannad, Y., Pal,

S.K. (eds.) Proceedings of the 2nd Mediterranean Conference on Pattern Recognition and Artificial Intelligence, MedPRAI 2018, Rabat, Morocco, March 27-28, 2018. pp. 76–83. ACM (2018). `https://doi.org/10.1145/3177148.3180086`

[431] Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. pp. 45–50. ELRA, Valletta, Malta (May 2010), `http://is.muni.cz/publication/884893/en`

[432] Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019. pp. 3980–3990. Association for Computational Linguistics (2019). `https://doi.org/10.18653/v1/D19-1410`

[433] Resnik, P.: Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. J. Artif. Intell. Res. **11**, 95–130 (1999). `https://doi.org/10.1613/jair.514`

[434] Reynaud, C., Safar, B.: When usual structural alignment techniques don't apply. In: Shvaiko, P., Euzenat, J., Noy, N.F., Stuckenschmidt, H., Benjamins, V.R., Uschold, M. (eds.) Proceedings of the 1st International Workshop on Ontology Matching (OM-2006) Collocated with the 5th International Semantic Web Conference (ISWC-2006), Athens, Georgia, USA, November 5, 2006. CEUR Workshop Proceedings, vol. 225. CEUR-WS.org (2006), `http://ceur-ws.org/Vol-225/paper18.pdf`

[435] Riemer, N.: Introducing Semantics, chap. 1, pp. 1–44. Cambridge Introductions to Language and Linguistics, Cambridge University Press, Cambridge; New York (2010)

[436] Riemer, N.: Introducing Semantics, chap. 2, pp. 45–86. Cambridge Introductions to Language and Linguistics, Cambridge University Press, Cambridge; New York (2010)

[437] Riemer, N.: Introducing Semantics, chap. 5, pp. 135–172. Cambridge Introductions to Language and Linguistics, Cambridge University Press, Cambridge; New York (2010)

[438] Ristoski, P.: Exploiting semantic web knowledge graphs in data mining. Ph.D. thesis, University of Mannheim, Germany (2018), `http://d-nb.info/1190653079`

[439] Ristoski, P., Paulheim, H.: A comparison of propositionalization strategies for creating features from linked open data. In: Tiddi, I., d'Aquin, M., Jay, N. (eds.) Proceedings of the 1st Workshop on Linked Data for Knowledge Discovery co-located with European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2014), Nancy, France, September 19th, 2014. CEUR Workshop Proceedings, vol. 1232. CEUR-WS.org (2014), `http://ceur-ws.org/Vol-1232/paper1.pdf`

[440] Ristoski, P., Paulheim, H.: Rdf2vec: RDF graph embeddings for data mining. In: Groth, P., Simperl, E., Gray, A.J.G., Sabou, M., Krötzsch, M., Lécué, F., Flöck, F., Gil, Y. (eds.) The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9981, pp. 498–514 (2016). `https://doi.org/10.1007/978-3-319-46523-4_30`

[441] Ristoski, P., Paulheim, H.: Semantic web in data mining and knowledge discovery: A comprehensive survey. J. Web Semant. **36**, 1–22 (2016). `https://doi.org/10.1016/j.websem.2016.01.001`

[442] Ristoski, P., Rosati, J., Noia, T.D., Leone, R.D., Paulheim, H.: Rdf2vec: RDF graph embeddings and their applications. Semantic Web **10**(4), 721–752 (2019). `https://doi.org/10.3233/SW-180317`

[443] Ristoski, P., de Vries, G.K.D., Paulheim, H.: A collection of benchmark datasets for systematic evaluations of machine learning on the semantic web. In: Groth, P., Simperl, E., Gray, A.J.G., Sabou, M., Krötzsch, M., Lécué, F., Flöck, F., Gil, Y. (eds.) The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9982, pp. 186–194 (2016). `https://doi.org/10.1007/978-3-319-46547-0_20`

[444] Rossi, A., Barbosa, D., Firmani, D., Matinata, A., Merialdo, P.: Knowledge graph embedding for link prediction: A comparative analysis. ACM Trans. Knowl. Discov. Data **15**(2), 14:1–14:49 (2021). `https://doi.org/10.1145/3424672`

[445] Roussille, P., Megdiche, I., Teste, O., Trojahn, C.: Boosting holistic ontology matching: Generating graph clique-based relaxed reference alignments for holistic evaluation. In: Faron-Zucker, C., Ghidini, C., Napoli, A., Toussaint, Y. (eds.) Knowledge Engineering and Knowledge Management - 21st International Conference, EKAW 2018, Nancy, France, November 12-16, 2018, Proceedings. Lecture Notes in Computer Science, vol. 11313, pp. 355–369. Springer (2018). https://doi.org/10.1007/978-3-030-03667-6_23

[446] Roussille, P., Megdiche, I., Teste, O., Trojahn, C.: Holontology: results of the 2018 OAEI evaluation campaign. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018. CEUR Workshop Proceedings, vol. 2288, pp. 167–172. CEUR-WS.org (2018), http://ceur-ws.org/Vol-2288/oaei18_paper8.pdf

[447] Rybinski, M., del Mar Roldán García, M., García-Nieto, J., Montes, J.F.A.: Dismatch results for OAEI 2016. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O., Ichise, R. (eds.) Proceedings of the 11th International Workshop on Ontology Matching co-located with the 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 18, 2016. CEUR Workshop Proceedings, vol. 1766, pp. 161–165. CEUR-WS.org (2016), http://ceur-ws.org/Vol-1766/oaei16_paper5.pdf

[448] Röder, M., Ngomo, A.C.N., Strohbach, M.: Deliverable 2.1: Detailed Architecture of the HOBBIT Platform (2016), https://project-hobbit.eu/wp-content/uploads/2016/11/D2.1_Detailed_Architecture_of_the_HOBBIT_Platform.pdf

[449] Sabou, M., d'Aquin, M., Motta, E.: Exploring the semantic web as background knowledge for ontology matching. J. Data Semant. **11**, 156–190 (2008). https://doi.org/10.1007/978-3-540-92148-6_6

[450] Sabou, M., Gracia, J.: Spider: Bringing non-equivalence mappings to OAEI. In: Shvaiko, P., Euzenat, J., Giunchiglia, F., Stuckenschmidt, H. (eds.) Proceedings of the 3rd International Workshop on Ontology Matching (OM-2008) Collocated with the 7th International Semantic Web Conference (ISWC-2008), Karlsruhe, Germany, October 26, 2008. CEUR Work-

shop Proceedings, vol. 431. CEUR-WS.org (2008), `http://ceur-ws.org/Vol-431/oaei08_paper11.pdf`

[451] Sabou, M., Gracia, J., Angeletou, S., d'Aquin, M., Motta, E.: Evaluating the semantic web: A task-based approach. In: Aberer, K., Choi, K., Noy, N.F., Allemang, D., Lee, K., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007. Lecture Notes in Computer Science, vol. 4825, pp. 423–437. Springer (2007). `https://doi.org/10.1007/978-3-540-76298-0_31`

[452] Salzberg, S.: On comparing classifiers: Pitfalls to avoid and a recommended approach. Data Min. Knowl. Discov. **1**(3), 317–328 (1997). `https://doi.org/10.1023/A:1009752403260`

[453] dos Santos, C.T., Moraes, M.C., Quaresma, P., Vieira, R.: Using cooperative agent negotiation for ontology mapping. In: Dunin-Keplicz, B., Omicini, A., Padget, J.A. (eds.) Proceedings of the 4th European Workshop on Multi-Agent Systems EUMAS'06, Lisbon, Portugal, December 14-15, 2006. CEUR Workshop Proceedings, vol. 223. CEUR-WS.org (2006), `http://ceur-ws.org/Vol-223/47.pdf`

[454] dos Santos, C.T., Moraes, M.C., Quaresma, P., Vieira, R.: A cooperative approach for composite ontology mapping. J. Data Semant. **10**, 237–263 (2008). `https://doi.org/10.1007/978-3-540-77688-8_8`

[455] dos Santos, C.T., Quaresma, P., Vieira, R.: An extended value-based argumentation framework for ontology mapping with confidence degrees. In: Rahwan, I., Parsons, S., Reed, C. (eds.) Argumentation in Multi-Agent Systems, 4th International Workshop, ArgMAS 2007, Honolulu, HI, USA, May 15, 2007, Revised Selected and Invited Papers. Lecture Notes in Computer Science, vol. 4946, pp. 132–144. Springer (2007). `https://doi.org/10.1007/978-3-540-78915-4_9`

[456] dos Santos, C.T., Quaresma, P., Vieira, R.: An argumentation framework based on confidence degrees to combine ontology mapping approaches. Int. J. Metadata Semant. Ontologies **3**(2), 142–150 (2008). `https://doi.org/10.1504/IJMSO.2008.021892`

[457] dos Santos, C.T., Quaresma, P., Vieira, R.: An argumentation framework based on strength for ontology mapping. In: Rahwan, I., Moraitis, P.

(eds.) Argumentation in Multi-Agent Systems, Fifth International Workshop, ArgMAS 2008, Estoril, Portugal, May 12, 2008. Revised Selected and Invited Papers. Lecture Notes in Computer Science, vol. 5384, pp. 57–71. Springer (2008). `https://doi.org/10.1007/978-3-642-00207-6_4`

[458] Saussure, F.d., Bally, C., Riedlinger, A., Lommel, H., Ernst, P.: Grundfragen der allgemeinen Sprachwissenschaft, chap. 1 (of part 1), pp. 76–119. De-Gruyter-Studienbuch, de Gruyter, Berlin, 3rd edn. (2001)

[459] Saussure, F.d., Bally, C., Riedlinger, A., Lommel, H., Ernst, P.: Grundfragen der allgemeinen Sprachwissenschaft, chap. 5 (of part 2), pp. 147–151. De-Gruyter-Studienbuch, de Gruyter, Berlin, 3rd edn. (2001)

[460] Saussure, F.d., Bally, C., Riedlinger, A., Lommel, H., Ernst, P.: Grundfragen der allgemeinen Sprachwissenschaft, chap. 6 (of part 2), pp. 152–159. De-Gruyter-Studienbuch, de Gruyter, Berlin, 3rd edn. (2001)

[461] Schadd, F.C., Roos, N.: Maasmatch results for OAEI 2011. In: Shvaiko, P., Euzenat, J., Heath, T., Quix, C., Mao, M., Cruz, I.F. (eds.) Proceedings of the 6th International Workshop on Ontology Matching, Bonn, Germany, October 24, 2011. CEUR Workshop Proceedings, vol. 814. CEUR-WS.org (2011), `http://ceur-ws.org/Vol-814/oaei11_paper9.pdf`

[462] Schadd, F.C., Roos, N.: Coupling of wordnet entries for ontology mapping using virtual documents. In: Shvaiko, P., Euzenat, J., Kementsietsidis, A., Mao, M., Noy, N.F., Stuckenschmidt, H. (eds.) Proceedings of the 7th International Workshop on Ontology Matching, Boston, MA, USA, November 11, 2012. CEUR Workshop Proceedings, vol. 946. CEUR-WS.org (2012), `http://ceur-ws.org/Vol-946/om2012_Tpaper3.pdf`

[463] Schadd, F.C., Roos, N.: A feature selection approach for anchor evaluation in ontology mapping. In: Klinov, P., Mouromtsev, D. (eds.) Knowledge Engineering and the Semantic Web - 5th International Conference, KESW 2014, Kazan, Russia, September 29-October 1, 2014. Proceedings. Communications in Computer and Information Science, vol. 468, pp. 160–174. Springer (2014). `https://doi.org/10.1007/978-3-319-11716-4_14`

[464] Schadd, F.C., Roos, N.: Word-sense disambiguation for ontology mapping: Concept disambiguation using virtual documents and information retrieval techniques. J. Data Semant. **4**(3), 167–186 (2015). `https://doi.org/10.1007/s13740-014-0045-5`

[465] Schmachtenberg, M., Bizer, C., Paulheim, H.: Adoption of the linked data best practices in different topical domains. In: Mika, P., Tudorache, T., Bernstein, A., Welty, C., Knoblock, C.A., Vrandecic, D., Groth, P., Noy, N.F., Janowicz, K., Goble, C.A. (eds.) The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I. Lecture Notes in Computer Science, vol. 8796, pp. 245–260. Springer (2014). https://doi.org/10.1007/978-3-319-11964-9_16

[466] Schriml, L.M., Arze, C., Nadendla, S., Chang, Y.W., Mazaitis, M., Felix, V., Feng, G., Kibbe, W.A.: Disease ontology: a backbone for disease semantic integration. Nucleic Acids Res. **40**(Database-Issue), 940–946 (2012). https://doi.org/10.1093/nar/gkr972

[467] Seitner, J., Bizer, C., Eckert, K., Faralli, S., Meusel, R., Paulheim, H., Ponzetto, S.P.: A large database of hypernymy relations extracted from the web. In: Calzolari, N., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S. (eds.) Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016. European Language Resources Association (ELRA) (2016), http://www.lrec-conf.org/proceedings/lrec2016/summaries/204.html

[468] Sen, S., Fernandes, D., Sarda, N.L.: Optimal learning of ontology mappings from human interactions. In: Meersman, R., Tari, Z. (eds.) On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS, OTM Confederated International Conferences CoopIS, DOA, ODBASE, GADA, and IS 2007, Vilamoura, Portugal, November 25-30, 2007, Proceedings, Part I. Lecture Notes in Computer Science, vol. 4803, pp. 1025–1033. Springer (2007). https://doi.org/10.1007/978-3-540-76848-7_67

[469] Sen, S., Somavarapu, S., Sarda, N.L.: Class structures and lexical similarities of class names for ontology matching. In: Collard, M. (ed.) Ontologies-Based Databases and Information Systems, First and Second VLDB Workshops, ODBIS 2005/2006 Trondheim, Norway, September 2-3, 2005, Seoul, Korea, September 11, 2006, Revised Papers. Lecture Notes in Computer Science, vol. 4623, pp. 18–36. Springer (2006). https://doi.org/10.1007/978-3-540-75474-9_2

[470] Sérasset, G.: Dbnary: Wiktionary as a lemon-based multilingual lexical resource in RDF. Semantic Web **6**(4), 355–361 (2015). `https://doi.org/10.3233/SW-140147`

[471] Sergieh, H.M., Unland, R.: IROM: information retrieval-based ontology matching. In: Declerck, T., Granitzer, M., Grzegorzek, M., Romanelli, M., Rüger, S.M., Sintek, M. (eds.) Semantic Multimedia - 5th International Conference on Semantic and Digital Media Technologies, SAMT 2010. Saarbrücken, Germany, December 1-3, 2010. Revised Selected Papers. Lecture Notes in Computer Science, vol. 6725, pp. 127–142. Springer (2010). `https://doi.org/10.1007/978-3-642-23017-2_9`

[472] Severo, B., dos Santos, C.T., Vieira, R.: VOAR: A visual and integrated ontology alignment environment. In: Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., Piperidis, S. (eds.) Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014. pp. 3671–3677. European Language Resources Association (ELRA) (2014), `http://www.lrec-conf.org/proceedings/lrec2014/summaries/851.html`

[473] Severo, B., Trojahn, C., Vieira, R.: VOAR 3.0 : a configurable environment for manipulating multiple ontology alignments. In: Nikitina, N., Song, D., Fokoue, A., Haase, P. (eds.) Proceedings of the ISWC 2017 Posters & Demonstrations and Industry Tracks co-located with 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 23rd - to - 25th, 2017. CEUR Workshop Proceedings, vol. 1963. CEUR-WS.org (2017), `http://ceur-ws.org/Vol-1963/paper550.pdf`

[474] Shadgara, B., Nejhadia, A.H., Osareha, A.: Ontology alignment using machine learning techniques. International Journal of Computer Science and Information Technology **3** (2011), `http://www.semantic-web-journal.net/sites/default/files/swj132.pdf`

[475] Shen, G., Jin, L., Zhao, Z., Jia, Z., He, W., Huang, Z.: Omreasoner: using reasoner for ontology matching: results for OAEI 2011. In: Shvaiko, P., Euzenat, J., Heath, T., Quix, C., Mao, M., Cruz, I.F. (eds.) Proceedings of the 6th International Workshop on Ontology Matching, Bonn, Germany, October 24, 2011. CEUR Workshop Proceedings, vol. 814. CEUR-WS.org (2011), `http://ceur-ws.org/Vol-814/oaei11_paper13.pdf`

[476] Sheth, A.P., Kashyap, V.: So far (schematically) yet so near (semantically). In: Hsiao, D.K., Neuhold, E.J., Sacks-Davis, R. (eds.) Proceedings of the IFIP WG 2.6 Database Semantics Conference on Interoperable Database Systems (DS-5), Lorne, Victoria, Australia, 16-20 November 1992. IFIP Transactions, vol. A-25, pp. 283–312. North-Holland (1992)

[477] Shi, B., Weninger, T.: Open-world knowledge graph completion. In: McIlraith, S.A., Weinberger, K.Q. (eds.) Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018. pp. 1957–1964. AAAI Press (2018), `https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16055`

[478] Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. J. Data Semant. pp. 146–171 (2005). `https://doi.org/10.1007/11603412_5`

[479] Shvaiko, P., Euzenat, J.: Ten challenges for ontology matching. In: Meersman, R., Tari, Z. (eds.) On the Move to Meaningful Internet Systems: OTM 2008, OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008, Monterrey, Mexico, November 9-14, 2008, Proceedings, Part II. Lecture Notes in Computer Science, vol. 5332, pp. 1164–1182. Springer (2008). `https://doi.org/10.1007/978-3-540-88873-4_18`

[480] Shvaiko, P., Euzenat, J.: Ontology matching: State of the art and future challenges. IEEE Trans. Knowl. Data Eng. **25**(1), 158–176 (2013). `https://doi.org/10.1109/TKDE.2011.253`

[481] Shvaiko, P., Euzenat, J., Giunchiglia, F., He, B. (eds.): Proceedings of the 2nd International Workshop on Ontology Matching (OM-2007) Collocated with the 6th International Semantic Web Conference (ISWC-2007) and the 2nd Asian Semantic Web Conference (ASWC-2007), Busan, Korea, November 11, 2007, CEUR Workshop Proceedings, vol. 304. CEUR-WS.org (2008), `http://ceur-ws.org/Vol-304`

[482] Shvaiko, P., Euzenat, J., Giunchiglia, F., Stuckenschmidt, H. (eds.): Proceedings of the 3rd International Workshop on Ontology Matching (OM-2008) Collocated with the 7th International Semantic Web Conference (ISWC-2008), Karlsruhe, Germany, October 26, 2008, CEUR Workshop

Proceedings, vol. 431. CEUR-WS.org (2008), `http://ceur-ws.org/Vol-431`

[483] Shvaiko, P., Euzenat, J., Giunchiglia, F., Stuckenschmidt, H., Mao, M., Cruz, I.F. (eds.): Proceedings of the 5th International Workshop on Ontology Matching (OM-2010), Shanghai, China, November 7, 2010, CEUR Workshop Proceedings, vol. 689. CEUR-WS.org (2010), `http://ceur-ws.org/Vol-689`

[484] Shvaiko, P., Euzenat, J., Giunchiglia, F., Stuckenschmidt, H., Noy, N.F., Rosenthal, A. (eds.): Proceedings of the 4th International Workshop on Ontology Matching (OM-2009) collocated with the 8th International Semantic Web Conference (ISWC-2009) Chantilly, USA, October 25, 2009, CEUR Workshop Proceedings, vol. 551. CEUR-WS.org (2009), `http://ceur-ws.org/Vol-551`

[485] Shvaiko, P., Euzenat, J., Heath, T., Quix, C., Mao, M., Cruz, I.F. (eds.): Proceedings of the 6th International Workshop on Ontology Matching, Bonn, Germany, October 24, 2011, CEUR Workshop Proceedings, vol. 814. CEUR-WS.org (2011), `http://ceur-ws.org/Vol-814`

[486] Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.): Proceedings of the 10th International Workshop on Ontology Matching collocated with the 14th International Semantic Web Conference (ISWC 2015), Bethlehem, PA, USA, October 12, 2015, CEUR Workshop Proceedings, vol. 1545. CEUR-WS.org (2016), `http://ceur-ws.org/Vol-1545`

[487] Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.): Proceedings of the 12th International Workshop on Ontology Matching co-located with the 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 21, 2017, CEUR Workshop Proceedings, vol. 2032. CEUR-WS.org (2017), `http://ceur-ws.org/Vol-2032`

[488] Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.): Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018, CEUR Workshop Proceedings, vol. 2288. CEUR-WS.org (2018), `http://ceur-ws.org/Vol-2288`

[489] Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O., Ichise, R. (eds.): Proceedings of the 11th International Workshop on Ontology Matching co-located with the 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 18, 2016, CEUR Workshop Proceedings, vol. 1766. CEUR-WS.org (2016), `http://ceur-ws.org/Vol-1766`

[490] Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.): Proceedings of the 14th International Workshop on Ontology Matching co-located with the 18th International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 26, 2019, CEUR Workshop Proceedings, vol. 2536. CEUR-WS.org (2020), `http://ceur-ws.org/Vol-2536`

[491] Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.): Proceedings of the 15th International Workshop on Ontology Matching co-located with the 19th International Semantic Web Conference (ISWC 2020), Virtual conference (originally planned to be in Athens, Greece), November 2, 2020, CEUR Workshop Proceedings, vol. 2788. CEUR-WS.org (2020), `http://ceur-ws.org/Vol-2788`

[492] Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.): Proceedings of the 16th International Workshop on Ontology Matching co-located with the 20th International Semantic Web Conference (ISWC 2021), Virtual conference, October 25, 2021, CEUR Workshop Proceedings, vol. 3063. CEUR-WS.org (2022), `http://ceur-ws.org/Vol-3063`

[493] Shvaiko, P., Euzenat, J., Kementsietsidis, A., Mao, M., Noy, N.F., Stuckenschmidt, H. (eds.): Proceedings of the 7th International Workshop on Ontology Matching, Boston, MA, USA, November 11, 2012, CEUR Workshop Proceedings, vol. 946. CEUR-WS.org (2012), `http://ceur-ws.org/Vol-946`

[494] Shvaiko, P., Euzenat, J., Mao, M., Jiménez-Ruiz, E., Li, J., Ngonga, A. (eds.): Proceedings of the 9th International Workshop on Ontology Matching collocated with the 13th International Semantic Web Conference (ISWC 2014), Riva del Garda, Trentino, Italy, October 20, 2014, CEUR Workshop Proceedings, vol. 1317. CEUR-WS.org (2014), `http://ceur-ws.org/Vol-1317`

[495] Shvaiko, P., Euzenat, J., Noy, N.F., Stuckenschmidt, H., Benjamins, V.R., Uschold, M. (eds.): Proceedings of the 1st International Workshop on Ontology Matching (OM-2006) Collocated with the 5th International Semantic Web Conference (ISWC-2006), Athens, Georgia, USA, November 5, 2006, CEUR Workshop Proceedings, vol. 225. CEUR-WS.org (2006), http://ceur-ws.org/Vol-225

[496] Shvaiko, P., Euzenat, J., Srinivas, K., Mao, M., Jiménez-Ruiz, E. (eds.): Proceedings of the 8th International Workshop on Ontology Matching co-located with the 12th International Semantic Web Conference (ISWC 2013), Sydney, Australia, October 21, 2013, CEUR Workshop Proceedings, vol. 1111. CEUR-WS.org (2013), http://ceur-ws.org/Vol-1111

[497] da Silva, J., Baião, F.A., Revoredo, K.: ALIN results for OAEI 2016. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O., Ichise, R. (eds.) Proceedings of the 11th International Workshop on Ontology Matching co-located with the 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 18, 2016. CEUR Workshop Proceedings, vol. 1766, pp. 130–137. CEUR-WS.org (2016), http://ceur-ws.org/Vol-1766/oaei16_paper1.pdf

[498] da Silva, J., Revoredo, K., Baião, F.A.: ALIN results for OAEI 2018. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018. CEUR Workshop Proceedings, vol. 2288, pp. 117–124. CEUR-WS.org (2018), http://ceur-ws.org/Vol-2288/oaei18_paper1.pdf

[499] Snoek, J., Larochelle, H., Adams, R.P.: Practical bayesian optimization of machine learning algorithms. In: Bartlett, P.L., Pereira, F.C.N., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States. pp. 2960–2968 (2012), https://proceedings.neurips.cc/paper/2012/hash/05311655a15b75fab86956663e1819cd-Abstract.html

[500] Sofronova, R., Biswas, R., Alam, M., Sack, H.: Entity typing based on rdf2vec using supervised and unsupervised methods. In: Harth, A., Presutti, V., Troncy, R., Acosta, M., Polleres, A., Fernández, J.D., Parreira, J.X., Hartig, O., Hose, K., Cochez, M. (eds.) The Semantic Web: ESWC 2020

Satellite Events - ESWC 2020 Satellite Events, Heraklion, Crete, Greece, May 31 - June 4, 2020, Revised Selected Papers. Lecture Notes in Computer Science, vol. 12124, pp. 203–207. Springer (2020). https://doi.org/10.1007/978-3-030-62327-2_35

[501] Son, J.W., Park, S., Park, S.: An ontology alignment based on parse tree kernel for combining structural and semantic information without explicit enumeration of features. In: 2008 IEEE / WIC / ACM International Conference on Web Intelligence, WI 2008, 9-12 December 2008, Sydney, NSW, Australia, Main Conference Proceedings. pp. 468–474. IEEE Computer Society (2008). https://doi.org/10.1109/WIIAT.2008.239

[502] Song, F., Zacharewicz, G., Chen, D.: Multi-strategies ontology alignment aggregated by AHP. In: Graña, M., Toro, C., Posada, J., Howlett, R.J., Jain, L.C. (eds.) Advances in Knowledge-Based and Intelligent Information and Engineering Systems - 16th Annual KES Conference, San Sebastian, Spain, 10-12 September 2012. Frontiers in Artificial Intelligence and Applications, vol. 243, pp. 1583–1592. IOS Press (2012), http://www.booksonline.iospress.nl/Content/View.aspx?piid=32098

[503] Speer, R., Chin, J., Havasi, C.: Conceptnet 5.5: An open multilingual graph of general knowledge. In: Singh, S.P., Markovitch, S. (eds.) Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA. pp. 4444–4451. AAAI Press (2017), http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14972

[504] Spiliopoulos, V., Valarakos, A.G., Vouros, G.A., Karkaletsis, V.: SEMA: results for the ontology alignment contest OAEI 2007. In: Shvaiko, P., Euzenat, J., Giunchiglia, F., He, B. (eds.) Proceedings of the 2nd International Workshop on Ontology Matching (OM-2007) Collocated with the 6th International Semantic Web Conference (ISWC-2007) and the 2nd Asian Semantic Web Conference (ASWC-2007), Busan, Korea, November 11, 2007. CEUR Workshop Proceedings, vol. 304. CEUR-WS.org (2007), http://ceur-ws.org/Vol-304/paper22.pdf

[505] Spohr, D., Hollink, L., Cimiano, P.: A machine learning approach to multilingual and cross-lingual ontology matching. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N.F., Blomqvist, E. (eds.) The Semantic Web - ISWC 2011 - 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part I. Lecture Notes in Computer Science, vol. 7031, pp. 665–680. Springer (2011). https://doi.org/10.1007/978-3-642-25073-6_42

[506] Spyns, P., Meersman, R., Jarrar, M.: Data modelling versus ontology engineering. SIGMOD Rec. **31**(4), 12–17 (2002). `https://doi.org/10.1145/637411.637413`

[507] Staab, S., Studer, R. (eds.): Handbook on Ontologies. International Handbooks on Information Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2 edn. (2009)

[508] Standard Upper Ontology Working Group: Standard Upper Ontology Working Group (SUO WG) - Home Page (2003), `http://web.archive.org/web/20140512225349/http://suo.ieee.org/`, accessed: 2014-05-12

[509] Steenwinckel, B., Vandewiele, G., Bonte, P., Weyns, M., Paulheim, H., Ristoski, P., Turck, F.D., Ongenae, F.: Walk extraction strategies for node embeddings with rdf2vec in knowledge graphs. In: Kotsis, G., Tjoa, A.M., Khalil, I., Moser, B., Mashkoor, A., Sametinger, J., Fensel, A., Gil, J.M., Fischer, L., Czech, G., Sobieczky, F., Khan, S. (eds.) Database and Expert Systems Applications - DEXA 2021 Workshops - BIOKDD, IWCFS, MLKgraphs, AI-CARES, ProTime, AISys 2021, Virtual Event, September 27-30, 2021, Proceedings. Communications in Computer and Information Science, vol. 1479, pp. 70–80. Springer (2021). `https://doi.org/10.1007/978-3-030-87101-7_8`

[510] Steenwinckel, B., Vandewiele, G., Rausch, I., Heyvaert, P., Taelman, R., Colpaert, P., Simoens, P., Dimou, A., Turck, F.D., Ongenae, F.: Facilitating the analysis of COVID-19 literature through a knowledge graph. In: Pan, J.Z., Tamma, V.A.M., d'Amato, C., Janowicz, K., Fu, B., Polleres, A., Seneviratne, O., Kagal, L. (eds.) The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference, Athens, Greece, November 2-6, 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12507, pp. 344–357. Springer (2020). `https://doi.org/10.1007/978-3-030-62466-8_22`

[511] Steinmetz, N., Knuth, M., Sack, H.: Statistical analyses of named entity disambiguation benchmarks. In: Hellmann, S., Filipowska, A., Barrière, C., Mendes, P.N., Kontokostas, D. (eds.) Proceedings of the NLP & DBpedia workshop co-located with the 12th International Semantic Web Conference (ISWC 2013), Sydney, Australia, October 22, 2013. CEUR Workshop Proceedings, vol. 1064. CEUR-WS.org (2013), `http://ceur-ws.org/Vol-1064/Steinmetz_Statistical.pdf`

[512] Straccia, U., Troncy, R.: omap: Results of the ontology alignment contest. In: Ashpole, B., Ehrig, M., Euzenat, J., Stuckenschmidt, H. (eds.) Integrating Ontologies '05, Proceedings of the K-CAP 2005 Workshop on Integrating Ontologies, Banff, Canada, October 2, 2005. CEUR Workshop Proceedings, vol. 156. CEUR-WS.org (2005), `http://ceur-ws.org/Vol-156/paper14.pdf`

[513] Stuart, J.N., Johnston, W.D., Humphreys, B.L.: Relationships in medical subject headings. Relationships in the organization of knowledge. New York: Kluwer Academic Publishers pp. 171–184 (2001), `https://doi.org/10.1007/978-94-015-9696-1_11`

[514] Su, X., Gulla, J.A.: An information retrieval approach to ontology mapping. Data Knowl. Eng. **58**(1), 47–69 (2006). `https://doi.org/10.1016/j.datak.2005.05.012`

[515] Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Williamson, C.L., Zurko, M.E., Patel-Schneider, P.F., Shenoy, P.J. (eds.) Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007. pp. 697–706. ACM (2007). `https://doi.org/10.1145/1242572.1242667`

[516] Sun, Z., Deng, Z., Nie, J., Tang, J.: Rotate: Knowledge graph embedding by relational rotation in complex space. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net (2019), `https://openreview.net/forum?id=HkgEQnRqYQ`

[517] Sure, Y., Corcho, Ó., Euzenat, J., Hughes, T. (eds.): EON 2004, Evaluation of Ontology-based Tools, Proceedings of the 3rd International Workshop on Evaluation of Ontology-based Tools held at the 3rd International Semantic Web Conference ISWC 2004, 7th November 2004, Hiroshima Prince Hotel, Hiroshima, Japan, CEUR Workshop Proceedings, vol. 128. CEUR-WS.org (2005), `http://ceur-ws.org/Vol-128`

[518] Sutanta, E., Wardoyo, R., Mustofa, K., Winarko, E.: Survey: Models and prototypes of schema matching. International Journal of Electrical & Computer Engineering (2088-8708) **6**(3) (2016), `https://www.doi.org/10.11591/ijece.v6i3.pp1011-1022`

[519] Tan, H., Lambrix, P.: A method for recommending ontology alignment strategies. In: Aberer, K., Choi, K., Noy, N.F., Allemang, D., Lee, K., Nixon,

L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007. Lecture Notes in Computer Science, vol. 4825, pp. 494–507. Springer (2007). `https://doi.org/10.1007/978-3-540-76298-0_36`

[520] Tan, H., Lambrix, P.: SAMBO results for the ontology alignment evaluation initiative 2007. In: Shvaiko, P., Euzenat, J., Giunchiglia, F., He, B. (eds.) Proceedings of the 2nd International Workshop on Ontology Matching (OM-2007) Collocated with the 6th International Semantic Web Conference (ISWC-2007) and the 2nd Asian Semantic Web Conference (ASWC-2007), Busan, Korea, November 11, 2007. CEUR Workshop Proceedings, vol. 304. CEUR-WS.org (2007), `http://ceur-ws.org/Vol-304/paper21.pdf`

[521] Tan, P., Steinbach, M.S., Karpatne, A., Kumar, V.: Introduction to Data Mining (Second Edition). Pearson (2019), `https://www-users.cse.umn.edu/%7Ekumar001/dmbook/index.php`

[522] Tang, J., Li, J., Liang, B., Huang, X., Li, Y., Wang, K.: Using bayesian decision for ontology mapping. J. Web Semant. **4**(4), 243–262 (2006). `https://doi.org/10.1016/j.websem.2006.06.001`

[523] Tang, J., Liang, B., Li, J., Wang, K.: Risk minimization based ontology mapping. In: Chi, C., Lam, K. (eds.) Content Computing, Advanced Workshop on Content Computing, AWCC 2004, ZhenJiang, JiangSu, China, November 15-17, 2004, Proceedings. Lecture Notes in Computer Science, vol. 3309, pp. 469–480. Springer (2004). `https://doi.org/10.1007/978-3-540-30483-8_58`

[524] Tang, Y., Wang, P., Pan, Z., Liu, H.: Lily results for OAEI 2018. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018. CEUR Workshop Proceedings, vol. 2288, pp. 179–186. CEUR-WS.org (2018), `http://ceur-ws.org/Vol-2288/oaei18_paper10.pdf`

[525] Tardieu, H., Rochfeld, A., Coletti, R.: La méthode merise, principes et outils. Revue du SCOM anc Bulletin O. et M. Organisation, Méthodes et Automatisation dans les Services Publics Paris **90** (1983)

[526] Thayasivam, U., Doshi, P.: On the utility of wordnet for ontology alignment: Is it really worth it? In: Proceedings of the 5th IEEE International Conference on Semantic Computing (ICSC 2011), Palo Alto, CA, USA, September 18-21, 2011. pp. 267–274. IEEE Computer Society (2011). https://doi.org/10.1109/ICSC.2011.28

[527] Thayasivam, U., Doshi, P.: Optima results for OAEI 2011. In: Shvaiko, P., Euzenat, J., Heath, T., Quix, C., Mao, M., Cruz, I.F. (eds.) Proceedings of the 6th International Workshop on Ontology Matching, Bonn, Germany, October 24, 2011. CEUR Workshop Proceedings, vol. 814. CEUR-WS.org (2011), http://ceur-ws.org/Vol-814/oaei11_paper14.pdf

[528] Thiéblin, É., Haemmerlé, O., Hernandez, N., Trojahn, C.: Survey on complex ontology matching. Semantic Web **11**(4), 689–727 (2020). https://doi.org/10.3233/SW-190366

[529] Thiéblin, É., Haemmerlé, O., Trojahn, C.: Automatic evaluation of complex alignments: An instance-based approach. Semantic Web **12**(5), 767–787 (2021). https://doi.org/10.3233/SW-210437

[530] Tigrine, A.N., Bellahsene, Z., Todorov, K.: LYAM++ results for OAEI 2015. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O. (eds.) Proceedings of the 10th International Workshop on Ontology Matching collocated with the 14th International Semantic Web Conference (ISWC 2015), Bethlehem, PA, USA, October 12, 2015. CEUR Workshop Proceedings, vol. 1545, pp. 176–180. CEUR-WS.org (2015), http://ceur-ws.org/Vol-1545/oaei15_paper11.pdf

[531] Tigrine, A.N., Bellahsene, Z., Todorov, K.: LYAM++ results for OAEI 2016. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O., Ichise, R. (eds.) Proceedings of the 11th International Workshop on Ontology Matching co-located with the 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 18, 2016. CEUR Workshop Proceedings, vol. 1766, pp. 196–200. CEUR-WS.org (2016), http://ceur-ws.org/Vol-1766/oaei16_paper11.pdf

[532] Tigrine, A.N., Bellahsene, Z., Todorov, K.: Selecting optimal background knowledge sources for the ontology matching task. In: Blomqvist, E., Ciancarini, P., Poggi, F., Vitali, F. (eds.) Knowledge Engineering and Knowledge Management - 20th International Conference, EKAW 2016, Bologna, Italy, November 19-23, 2016, Proceedings. Lecture Notes in Computer Sci-

ence, vol. 10024, pp. 651–665 (2016). https://doi.org/10.1007/978-3-319-49004-5_42

[533] Todorov, K., Hudelot, C., Geibel, P.: Fuzzy and cross-lingual ontology matching mediated by background knowledge. In: Bobillo, F., Carvalho, R.N., da Costa, P.C.G., d'Amato, C., Fanizzi, N., Laskey, K.B., Laskey, K.J., Lukasiewicz, T., Nickles, M., Pool, M. (eds.) Uncertainty Reasoning for the Semantic Web III - ISWC International Workshops, URSW 2011-2013, Revised Selected Papers. Lecture Notes in Computer Science, vol. 8816, pp. 142–162. Springer (2014). https://doi.org/10.1007/978-3-319-13413-0_8

[534] Toutanova, K., Chen, D.: Observed versus latent features for knowledge base and text inference. In: Proceedings of the 3rd workshop on continuous vector space models and their compositionality. pp. 57–66 (2015). https://doi.org/10.18653/v1/W15-4007

[535] Toutanova, K., Chen, D., Pantel, P., Poon, H., Choudhury, P., Gamon, M.: Representing text for joint embedding of text and knowledge bases. In: Màrquez, L., Callison-Burch, C., Su, J., Pighin, D., Marton, Y. (eds.) Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015. pp. 1499–1509. The Association for Computational Linguistics (2015). https://doi.org/10.18653/v1/d15-1174

[536] Tran, Q., Ichise, R., Ho, B.: Cluster-based similarity aggregation for ontology matching. In: Shvaiko, P., Euzenat, J., Heath, T., Quix, C., Mao, M., Cruz, I.F. (eds.) Proceedings of the 6th International Workshop on Ontology Matching, Bonn, Germany, October 24, 2011. CEUR Workshop Proceedings, vol. 814. CEUR-WS.org (2011), http://ceur-ws.org/Vol-814/oaei11_paper5.pdf

[537] Trojahn, C., Vieira, R., Schmidt, D., Pease, A., Guizzardi, G.: Foundational ontologies meet ontology matching: A survey. Semantic Web pp. 1–20 (2021), https://www.doi.org/10.3233/SW-210447, preprint

[538] Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: Balcan, M., Weinberger, K.Q. (eds.) Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016. JMLR Workshop and Conference Proceedings, vol. 48, pp. 2071–2080. JMLR.org (2016), http://proceedings.mlr.press/v48/trouillon16.html

[539] Tudorache, T., Vendetti, J., Noy, N.F.: Web-protege: A lightweight OWL ontology editor for the web. In: Dolbear, C., Ruttenberg, A., Sattler, U. (eds.) Proceedings of the Fifth OWLED Workshop on OWL: Experiences and Directions, collocated with the 7th International Semantic Web Conference (ISWC-2008), Karlsruhe, Germany, October 26-27, 2008. CEUR Workshop Proceedings, vol. 432. CEUR-WS.org (2008), `http://ceur-ws.org/Vol-432/owled2008eu_submission_40.pdf`

[540] Uschold, M., Grüninger, M.: Ontologies and semantics for seamless connectivity. SIGMOD Rec. **33**(4), 58–64 (2004). `https://doi.org/10.1145/1041410.1041420`

[541] Vaccari, L., Shvaiko, P., Pane, J., Besana, P., Marchese, M.: An evaluation of ontology matching in geo-service applications. GeoInformatica **16**(1), 31–66 (2012). `https://doi.org/10.1007/s10707-011-0125-8`

[542] Vandewiele, G., Steenwinckel, B., Bonte, P., Weyns, M., Paulheim, H., Ristoski, P., Turck, F.D., Ongenae, F.: Walk extraction strategies for node embeddings with rdf2vec in knowledge graphs. CoRR **abs/2009.04404** (2020), `https://arxiv.org/abs/2009.04404`

[543] Vandewiele, G., Steenwinckel, B., Weyns, M., Bonte, P., Ongenae, F., Turck, F.D.: pyrdf2vec: A python library for rdf2vec (2020), `https://github.com/IBCNServices/pyRDF2Vec`

[544] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. pp. 5998–6008 (2017), `https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html`

[545] Vazquez, R., Swoboda, N.: Combining the semantic web with the web as background knowledge for ontology mapping. In: Meersman, R., Tari, Z. (eds.) On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS, OTM Confederated International Conferences CoopIS, DOA, ODBASE, GADA, and IS 2007, Vilamoura, Portugal, November 25-30, 2007, Proceedings, Part I. Lecture Notes in Computer Science, vol. 4803, pp. 814–831. Springer (2007). `https://doi.org/10.1007/978-3-540-76848-7_55`

[546] Vennesland, A., Aalberg, T.: False-positive reduction in ontology matching based on concepts' domain similarity. In: Méndez, E., Crestani, F., Ribeiro, C., David, G., Lopes, J.C. (eds.) Digital Libraries for Open Knowledge, 22nd International Conference on Theory and Practice of Digital Libraries, TPDL 2018, Porto, Portugal, September 10-13, 2018, Proceedings. Lecture Notes in Computer Science, vol. 11057, pp. 344–348. Springer (2018). `https://doi.org/10.1007/978-3-030-00066-0_37`

[547] Vennesland, A., Gorman, J., Wilson, S., Neumayr, B., Schuetz, C.G.: Automated compliance verification in ATM using principles from ontology matching. In: Aveiro, D., Dietz, J.L.G., Filipe, J. (eds.) Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K 2018, Volume 2: KEOD, Seville, Spain, September 18-20, 2018. pp. 39–50. SciTePress (2018). `https://doi.org/10.5220/0006924000390050`

[548] Visser, P.R., Jones, D.M., Bench-Capon, T.J., Shave, M.: An analysis of ontology mismatches; heterogeneity versus interoperability. In: AAAI 1997 Spring Symposium on Ontological Engineering, Stanford CA., USA. pp. 164–72 (1997)

[549] Volz, J., Bizer, C., Gaedke, M., Kobilarov, G.: Silk - A link discovery framework for the web of data. In: Bizer, C., Heath, T., Berners-Lee, T., Idehen, K. (eds.) Proceedings of the WWW2009 Workshop on Linked Data on the Web, LDOW 2009, Madrid, Spain, April 20, 2009. CEUR Workshop Proceedings, vol. 538. CEUR-WS.org (2009), `http://ceur-ws.org/Vol-538/ldow2009_paper13.pdf`

[550] Vrandecic, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Commun. ACM **57**(10), 78–85 (2014). `https://doi.org/10.1145/2629489`

[551] W3C: Extensible Markup Language (XML) 1.1 (Second Edition) (2006), `https://www.w3.org/TR/xml11/`, accessed: 2022-05-10

[552] W3C: OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition) (2012), `https://www.w3.org/TR/2012/REC-owl2-syntax-20121211/`, accessed: 2022-05-10

[553] W3C: SPARQL 1.1 Protocol. `https://www.w3.org/TR/2013/REC-sparql11-protocol-20130321/` (2013), `https://www.w3.org/TR/2013/REC-sparql11-protocol-20130321/`

[554] W3C: JSON-LD 1.0 (2014), https://www.w3.org/TR/json-ld/, accessed: 2022-05-10

[555] W3C: RDF 1.1 N-Quads (2014), https://www.w3.org/TR/n-quads/, accessed: 2022-05-10

[556] W3C: RDF 1.1 Primer (2014), https://www.w3.org/TR/rdf11-primer/, accessed: 2022-05-10

[557] W3C: RDF 1.1 Turtle (2014), https://www.w3.org/TR/2014/REC-turtle-20140225/, accessed: 2022-05-10

[558] Wang, L.L., Bhagavatula, C., Neumann, M., Lo, K., Wilhelm, C., Ammar, W.: Ontology alignment in the biomedical domain using entity definitions and context. In: Demner-Fushman, D., Cohen, K.B., Ananiadou, S., Tsujii, J. (eds.) Proceedings of the BioNLP 2018 workshop, Melbourne, Australia, July 19, 2018. pp. 47–55. Association for Computational Linguistics (2018). https://doi.org/10.18653/v1/w18-2306

[559] Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: A survey of approaches and applications. IEEE Trans. Knowl. Data Eng. **29**(12), 2724–2743 (2017). https://doi.org/10.1109/TKDE.2017.2754499

[560] Wang, S., Wang, G., Liu, X.: Results of NBJLM for OAEI 2010. In: Shvaiko, P., Euzenat, J., Giunchiglia, F., Stuckenschmidt, H., Mao, M., Cruz, I.F. (eds.) Proceedings of the 5th International Workshop on Ontology Matching (OM-2010), Shanghai, China, November 7, 2010. CEUR Workshop Proceedings, vol. 689. CEUR-WS.org (2010), http://ceur-ws.org/Vol-689/oaei10_paper10.pdf

[561] Wang, X., Haas, L.M., Meliou, A.: Explaining data integration. IEEE Data Eng. Bull. **41**(2), 47–58 (2018), http://sites.computer.org/debull/A18june/p47.pdf

[562] Wang, X., Gao, T., Zhu, Z., Zhang, Z., Liu, Z., Li, J., Tang, J.: KEPLER: A unified model for knowledge embedding and pre-trained language representation. Trans. Assoc. Comput. Linguistics **9**, 176–194 (2021), https://transacl.org/ojs/index.php/tacl/article/view/2447

[563] Wang, Y., Gong, J., Wang, Z., Liang, Y., Lu, N.: Combining two strategies for ontology mapping. In: Cordeiro, J.A.M., Pedrosa, V., Encarnação, B., Filipe, J. (eds.) WEBIST 2006, Proceedings of the Second International Conference on Web Information Systems and Technologies: Internet Technology

/ Web Interface and Applications, Setúbal, Portugal, April 11-13, 2006. pp. 381–386. INSTICC Press (2006)

[564] Wang, Y., Gong, J., Wang, Z., Zhou, C.: A composite approach for ontology mapping. In: Bell, D.A., Hong, J. (eds.) Flexible and Efficient Information Handling, 23rd British National Conference on Databases, BNCOD 23, Belfast, Northern Ireland, UK, July 18-20, 2006, Proceedings. Lecture Notes in Computer Science, vol. 4042, pp. 282–285. Springer (2006). `https://doi.org/10.1007/11788911_31`

[565] Wang, Y., Liu, W., Bell, D.A.: A concept hierarchy based ontology mapping approach. In: Bi, Y., Williams, M. (eds.) Knowledge Science, Engineering and Management, 4th International Conference, KSEM 2010, Belfast, Northern Ireland, UK, September 1-3, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6291, pp. 101–113. Springer (2010). `https://doi.org/10.1007/978-3-642-15280-1_12`

[566] Wang, Y., Liu, X., Chen, J.: A multi-stage strategy for ontology mapping resolution. In: Proceedings of the IEEE International Conference on Information Reuse and Integration, IRI 2008, 13-15 July 2008, Las Vegas, Nevada, USA. pp. 345–350. IEEE Systems, Man, and Cybernetics Society (2008). `https://doi.org/10.1109/IRI.2008.4583055`

[567] Wang, Y.: Problog program based ontology matching. In: Zhang, S., Wirsing, M., Zhang, Z. (eds.) Knowledge Science, Engineering and Management - 8th International Conference, KSEM 2015, Chongqing, China, October 28-30, 2015, Proceedings. Lecture Notes in Computer Science, vol. 9403, pp. 778–783. Springer (2015). `https://doi.org/10.1007/978-3-319-25159-2_72`

[568] Wang, Y., Dong, L., Jiang, X., Ma, X., Li, Y., Zhang, H.: Kg2vec: A node2vec-based vectorization model for knowledge graph. Plos one **16**(3), e0248552 (2021), `https://doi.org/10.1371/journal.pone.0248552`

[569] Wang, Z., Wang, Y., Du, T.: Ontology mapping approach based on parsing graph. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Taipei, Taiwan, October 8-11, 2006. pp. 5309–5313. IEEE (2006). `https://doi.org/10.1109/ICSMC.2006.385152`

[570] Wang, Z., Wang, Y., Zhang, S., Shen, G., Du, T.: Effective large scale ontology mapping. In: Lang, J., Lin, F., Wang, J. (eds.) Knowledge Science, Engineering and Management, First International Conference, KSEM 2006,

Guilin, China, August 5-8, 2006, Proceedings. Lecture Notes in Computer Science, vol. 4092, pp. 454–465. Springer (2006). `https://doi.org/10.1007/11811220_38`

[571] Whetzel, P.L., Noy, N.F., Shah, N.H., Alexander, P.R., Nyulas, C., Tudorache, T., Musen, M.A.: Bioportal: enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications. Nucleic Acids Res. **39**(Web-Server-Issue), 541–545 (2011). `https://doi.org/10.1093/nar/gkr469`

[572] Wiktionary: Semantics, `https://web.archive.org/web/20220506154229/https://en.wiktionary.org/wiki/semantics`, accessed: 2022-05-06

[573] Wiktionary: Syntax, `http://web.archive.org/web/20220421073435/https://en.wiktionary.org/wiki/syntax`, accessed: 2022-04-01

[574] Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.W., da Silva Santos, L.B., Bourne, P.E., et al.: The fair guiding principles for scientific data management and stewardship. Scientific data **3**(1), 1–9 (2016). `https://doi.org/10.1038/sdata.2016.18`

[575] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Brew, J.: Huggingface's transformers: State-of-the-art natural language processing. CoRR **abs/1910.03771** (2019), `http://arxiv.org/abs/1910.03771`

[576] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T.L., Gugger, S., Drame, M., Lhoest, Q., Rush, A.M.: Transformers: State-of-the-art natural language processing. In: Liu, Q., Schlangen, D. (eds.) Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020. pp. 38–45. Association for Computational Linguistics (2020). `https://doi.org/10.18653/v1/2020.emnlp-demos.6`

[577] Wrigley, S.N., Garcia-Castro, R., Nixon, L.J.B.: Semantic evaluation at large scale (SEALS). In: Mille, A., Gandon, F., Misselis, J., Rabinovich, M., Staab, S. (eds.) Proceedings of the 21st World Wide Web Conference, WWW 2012,

Lyon, France, April 16-20, 2012 (Companion Volume). pp. 299–302. ACM (2012). https://doi.org/10.1145/2187980.2188033

[578] Wu, J., Wang, Y.: IAOM: an integrated automatic ontology mapping approach towards knowledge integration. In: Li, K., Fei, M., Irwin, G.W., Ma, S. (eds.) Bio-Inspired Computational Intelligence and Applications, International Conference on Life System Modeling and Simulation, LSMS 2007, Shanghai, China, September 14-17, 2007, Proceedings. Lecture Notes in Computer Science, vol. 4688, pp. 502–509. Springer (2007). https://doi.org/10.1007/978-3-540-74769-7_54

[579] Wu, J., Lv, J., Guo, H., Ma, S.: Daeom: A deep attentional embedding approach for biomedical ontology matching. Applied Sciences **10**(21), 7909 (2020)

[580] Wu, Z., Palmer, M.S.: Verb semantics and lexical selection. In: Pustejovsky, J. (ed.) 32nd Annual Meeting of the Association for Computational Linguistics, 27-30 June 1994, New Mexico State University, Las Cruces, New Mexico, USA, Proceedings. pp. 133–138. Morgan Kaufmann Publishers / ACL (1994). https://doi.org/10.3115/981732.981751

[581] Xia, H., Zheng, X., Hu, X., Tian, Y.: Multi-strategy ontology mapping based on stacking method. In: Chen, Y., Deng, H., Zhang, D., Xiao, Y. (eds.) Sixth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2009, Tianjin, China, 14-16 August 2009, 6 Volumes. pp. 262–269. IEEE Computer Society (2009). https://doi.org/10.1109/FSKD.2009.103

[582] Xie, C., Chekol, M.W., Spahiu, B., Cai, H.: Leveraging structural information in ontology matching. In: Barolli, L., Takizawa, M., Enokido, T., Jara, A.J., Bocchi, Y. (eds.) 30th IEEE International Conference on Advanced Information Networking and Applications, AINA 2016, Crans-Montana, Switzerland, 23-25 March, 2016. pp. 1108–1115. IEEE Computer Society (2016). https://doi.org/10.1109/AINA.2016.64

[583] Xie, R., Liu, Z., Jia, J., Luan, H., Sun, M.: Representation learning of knowledge graphs with entity descriptions. In: Schuurmans, D., Wellman, M.P. (eds.) Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA. pp. 2659–2665. AAAI Press (2016), http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12216

[584] Xu, M.: Understanding graph embedding methods and their applications. SIAM Rev. **63**(4), 825–853 (2021). https://doi.org/10.1137/20M1386062

[585] Xue, X.: Optimizing ontology alignments through NSGA-II using an aggregation strategy and a mapping extraction approach. In: Pan, J., Lee, I., Huang, H., Yang, C. (eds.) 2015 International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIH-MSP 2015, Adelaide, Australia, September 23-25, 2015. pp. 349–352. IEEE (2015). https://doi.org/10.1109/IIH-MSP.2015.100

[586] Xue, X.: Efficient ontology meta-matching using alignment prescreening approach and gaussian random field model assisted NSGA-II. J. Inf. Hiding Multim. Signal Process. **8**(5), 1062–1068 (2017), http://bit.kuas.edu.tw/%7Ejihmsp/2017/vol8/JIH-MSP-2017-05-010.pdf

[587] Xue, X., Chen, J.: Optimizing ontology alignment through hybrid population-based incremental learning algorithm. Memetic Comput. **11**(2), 209–217 (2019). https://doi.org/10.1007/s12293-018-0255-8

[588] Xue, X., Chen, J.: Optimizing sensor ontology alignment through compact co-firefly algorithm. Sensors **20**(7), 2056 (2020). https://doi.org/10.3390/s20072056

[589] Xue, X., Liu, J.: Collaborative ontology matching based on compact interactive evolutionary algorithm. Knowl. Based Syst. **137**, 94–103 (2017). https://doi.org/10.1016/j.knosys.2017.09.017

[590] Xue, X., Liu, J., Tsai, P., Zhan, X., Ren, A.: Optimizing ontology alignment by using compact genetic algorithm. In: 11th International Conference on Computational Intelligence and Security, CIS 2015, Shenzhen, China, December 19-20, 2015. pp. 231–234. IEEE Computer Society (2015). https://doi.org/10.1109/CIS.2015.64

[591] Xue, X., Pan, J.: A segment-based approach for large-scale ontology matching. Knowl. Inf. Syst. **52**(2), 467–484 (2017). https://doi.org/10.1007/s10115-016-1018-9

[592] Xue, X., Tsai, P., Zhang, L.: Using parallel compact evolutionary algorithm for optimizing ontology alignment. In: Pan, J., Lin, J.C., Wang, C., Jiang, X.H. (eds.) Genetic and Evolutionary Computing - Proceedings of

the Tenth International Conference on Genetic and Evolutionary Computing, ICGEC 2016, November 7-9, 2016, Fuzhou City, Fujian Province, China. Advances in Intelligent Systems and Computing, vol. 536, pp. 157–165 (2016). https://doi.org/10.1007/978-3-319-48490-7_19

[593] Xue, X., Wang, Y.: Ontology alignment based on instance using NSGA-II. J. Inf. Sci. **41**(1), 58–70 (2015). https://doi.org/10.1177/0165551514550142

[594] Xue, X., Wang, Y.: Optimizing ontology alignments through a memetic algorithm using both matchfmeasure and unanimous improvement ratio. Artif. Intell. **223**, 65–81 (2015). https://doi.org/10.1016/j.artint.2015.03.001

[595] Xue, X., Wang, Y., Hao, W.: Using MOEA/D for optimizing ontology alignments. Soft Comput. **18**(8), 1589–1601 (2014). https://doi.org/10.1007/s00500-013-1165-9

[596] Xue, X., Wang, Y., Hao, W.: Optimizing ontology alignments by using NSGA-II. Int. Arab J. Inf. Technol. **12**(2), 175–181 (2015), http://iajit.org/index.php?option=com_content&task=blog category&id=97&Itemid=373

[597] Xue, X., Wang, Y., Hao, W., Hou, J.: Optimizing ontology alignments through NSGA-II without using reference alignment. Comput. Informatics **33**(4), 857–876 (2014), http://www.cai.sk/ojs/index.php/cai/article/view/1159

[598] Xue, X., Wang, Y., Ren, A.: Optimizing ontology alignment through memetic algorithm based on partial reference alignment. Expert Syst. Appl. **41**(7), 3213–3222 (2014). https://doi.org/10.1016/j.eswa.2013.11.021

[599] Xue, X., Yang, C., Jiang, C., Tsai, P., Mao, G., Zhu, H.: Optimizing ontology alignment through linkage learning on entity correspondences. Complex. **2021**, 5574732:1–5574732:12 (2021). https://doi.org/10.1155/2021/5574732

[600] Yang, B., Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1412.6575

[601] Yang, F.: Variable weight semantic graph-based ontology mapping method. Expert Syst. J. Knowl. Eng. **36**(5) (2019). `https://doi.org/10.1111/exsy.12337`

[602] Yang, Y., Cer, D., Ahmad, A., Guo, M., Law, J., Constant, N., Ábrego, G.H., Yuan, S., Tar, C., Sung, Y., Strope, B., Kurzweil, R.: Multilingual universal sentence encoder for semantic retrieval. In: Celikyilmaz, A., Wen, T. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020. pp. 87–94. Association for Computational Linguistics (2020). `https://doi.org/10.18653/v1/2020.acl-demos.12`

[603] Yao, L., Mao, C., Luo, Y.: KG-BERT: BERT for knowledge graph completion. CoRR **abs/1909.03193** (2019), `http://arxiv.org/abs/1909.03193`

[604] Zago, M.G.: Why the Web 3.0 Matters and you should know about it (2018), `https://medium.com/@matteozago/why-the-web-3-0-matters-and-you-should-know-about-it-a5851d63c949`, accessed: 2022-05-10

[605] Zaveri, A., Dumontier, M.: Ontology mapping for life science linked data. In: Tao, C., Jiang, G., Song, D., Heflin, J., Schilder, F. (eds.) Proceedings of the First International Workshop on Biomedical Data Integration and Discovery (BMDID 2016) co-located with The 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 17, 2016. CEUR Workshop Proceedings, vol. 1709. CEUR-WS.org (2016), `http://ceur-ws.org/Vol-1709/BMDID_2016_paper_11.pdf`

[606] Zhang, L., Ding, S., Tang, S.: An applicable approach to ontology mapping based on knowledge base. In: First IITA International Joint Conference on Artificial Intelligence, Hainan Island, China, 25-26 April 2009. pp. 211–214. IEEE Computer Society (2009). `https://doi.org/10.1109/JCAI.2009.29`

[607] Zhang, L., Ren, J., Li, X.: OIM-SM: A method for ontology integration based on semantic mapping. J. Intell. Fuzzy Syst. **32**(3), 1983–1995 (2017). `https://doi.org/10.3233/JIFS-161553`

[608] Zhang, M., Duan, Z., Zhao, C.: Semi-automatically annotating data semantics to web services using ontology mapping. In: Proceedings of the 12th International Conference on CSCW in Design, CSCWD 2008, April 16-18, 2008, Nanyang Hotel, Xi'an Jiaotong University, Xi'an, China.

pp. 470–475. IEEE (2008). https://doi.org/10.1109/CSCWD.2008.4537024

[609] Zhang, S., Bodenreider, O.: NLM anatomical ontology alignment system. results of the 2006 ontology alignment contest. In: Shvaiko, P., Euzenat, J., Noy, N.F., Stuckenschmidt, H., Benjamins, V.R., Uschold, M. (eds.) Proceedings of the 1st International Workshop on Ontology Matching (OM-2006) Collocated with the 5th International Semantic Web Conference (ISWC-2006), Athens, Georgia, USA, November 5, 2006. CEUR Workshop Proceedings, vol. 225. CEUR-WS.org (2006), http://ceur-ws.org/Vol-225/paper14.pdf

[610] Zhang, S., Bodenreider, O.: Hybrid alignment strategy for anatomical ontologies: Results of the 2007 ontology alignment contest. In: Shvaiko, P., Euzenat, J., Giunchiglia, F., He, B. (eds.) Proceedings of the 2nd International Workshop on Ontology Matching (OM-2007) Collocated with the 6th International Semantic Web Conference (ISWC-2007) and the 2nd Asian Semantic Web Conference (ASWC-2007), Busan, Korea, November 11, 2007. CEUR Workshop Proceedings, vol. 304. CEUR-WS.org (2007), http://ceur-ws.org/Vol-304/paper11.pdf

[611] Zhang, Y., Panangadan, A.V., Prasanna, V.K.: UFOM: unified fuzzy ontology matching. In: Joshi, J., Bertino, E., Thuraisingham, B.M., Liu, L. (eds.) Proceedings of the 15th IEEE International Conference on Information Reuse and Integration, IRI 2014, Redwood City, CA, USA, August 13-15, 2014. pp. 787–794. IEEE Computer Society (2014). https://doi.org/10.1109/IRI.2014.7051969

[612] Zhang, Y., Wang, X., Lai, S., He, S., Liu, K., Zhao, J., Lv, X.: Ontology matching with word embeddings. In: Sun, M., Liu, Y., Zhao, J. (eds.) Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data - 13th China National Conference, CCL 2014, and Second International Symposium, NLP-NABD 2014, Wuhan, China, October 18-19, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8801, pp. 34–45. Springer (2014). https://doi.org/10.1007/978-3-319-12277-9_4

[613] Zhao, M., Zhang, S.: Identifying and validating ontology mappings by formal concept analysis. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., Hassanzadeh, O., Ichise, R. (eds.) Proceedings of the 11th International Workshop on Ontology Matching co-located with the 15th

International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 18, 2016. CEUR Workshop Proceedings, vol. 1766, pp. 61–72. CEUR-WS.org (2016), http://ceur-ws.org/Vol-1766/om2016_Tpaper6.pdf

[614] Zheng, D., Song, X., Ma, C., Tan, Z., Ye, Z., Dong, J., Xiong, H., Zhang, Z., Karypis, G.: DGL-KE: training knowledge graph embeddings at scale. In: Huang, J., Chang, Y., Cheng, X., Kamps, J., Murdock, V., Wen, J., Liu, Y. (eds.) Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020. pp. 739–748. ACM (2020). https://doi.org/10.1145/3397271.3401172

[615] Zheng, Q., Shao, C., Li, J., Wang, Z., Hu, L.: Rimom2013 results for OAEI 2013. In: Shvaiko, P., Euzenat, J., Srinivas, K., Mao, M., Jiménez-Ruiz, E. (eds.) Proceedings of the 8th International Workshop on Ontology Matching co-located with the 12th International Semantic Web Conference (ISWC 2013), Sydney, Australia, October 21, 2013. CEUR Workshop Proceedings, vol. 1111, pp. 161–168. CEUR-WS.org (2013), http://ceur-ws.org/Vol-1111/oaei13_paper9.pdf

[616] Zhong, Q., Li, H., Li, J., Xie, G.T., Tang, J., Zhou, L., Pan, Y.: A gauss function based approach for unbalanced ontology matching. In: Çetintemel, U., Zdonik, S.B., Kossmann, D., Tatbul, N. (eds.) Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2009, Providence, Rhode Island, USA, June 29 - July 2, 2009. pp. 669–680. ACM (2009). https://doi.org/10.1145/1559845.1559915

[617] Zhu, H., Xue, X., Jiang, C., Ren, H.: Multiobjective sensor ontology matching technique with user preference metrics. Wirel. Commun. Mob. Comput. **2021**, 5594553:1–5594553:9 (2021). https://doi.org/10.1155/2021/5594553

[618] Zouaq, A., Martel, F.: What is the schema of your knowledge graph?: leveraging knowledge graph embeddings and clustering for expressive taxonomy learning. In: Groppe, S., Gruenwald, L., Presutti, V. (eds.) Proceedings of The International Workshop on Semantic Big Data, SBD@SIGMOD 2020, Portland, Oregon, USA, June 19, 2020. pp. 6:1–6:6. ACM (2020). https://doi.org/10.1145/3391274.3393637