information

MDPI

Article

Language-Agnostic Relation Extraction from Abstracts in Wikis

Nicolas Heist, Sven Hertling and Heiko Paulheim *

Data and Web Science Group, University of Mannheim, Mannheim 68131, Germany;
nico@informatik.uni-mannheim.de (N.H.); sven@informatik.uni-mannheim.de (S.H.)
* Correspondence: heiko@informatik.uni-mannheim.de

Received: 5 February 2018; Accepted: 28 March 2018; Published: 29 March 2018

Abstract: Large-scale knowledge graphs, such as DBpedia, Wikidata, or YAGO, can be enhanced by relation extraction from text, using the data in the knowledge graph as training data, i.e., using *distant supervision*. While most existing approaches use language-specific methods (usually for English), we present a language-agnostic approach that exploits background knowledge from the graph instead of language-specific techniques and builds machine learning models only from language-independent features. We demonstrate the extraction of relations from Wikipedia abstracts, using the twelve largest language editions of Wikipedia. From those, we can extract 1.6 M new relations in DBpedia at a level of precision of 95%, using a RandomForest classifier trained only on language-independent features. We furthermore investigate the similarity of models for different languages and show an exemplary geographical breakdown of the information extracted. In a second series of experiments, we show how the approach can be transferred to DBkWik, a knowledge graph extracted from thousands of Wikis. We discuss the challenges and first results of extracting relations from a larger set of Wikis, using a less formalized knowledge graph.

Keywords: relation extraction; knowledge graphs; Wikipedia; DBpedia; DBkWik; Wiki farms

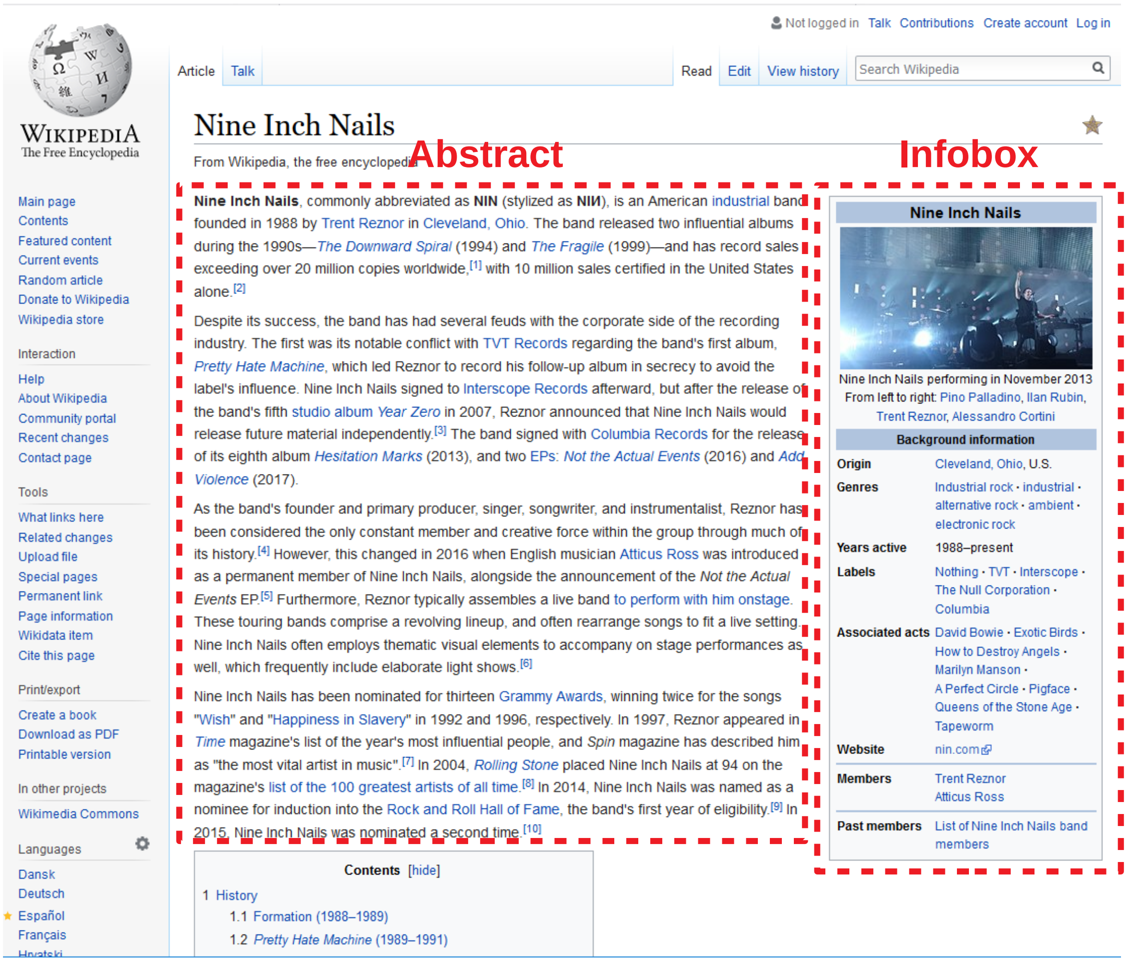
1. Introduction

Large-scale knowledge graphs, such as DBpedia [1], Freebase [2], Wikidata [3], or YAGO [4], are usually built using heuristic extraction methods, by exploiting crowd-sourcing processes, or both [5]. These approaches can help creating large-scale public cross-domain knowledge graphs, but are prone both to errors as well as incompleteness. Therefore, over the last years, various methods for refining those knowledge graphs have been developed [6].

Many of those knowledge graphs are built from semi-structured input. Most prominently, DBpedia and YAGO are built from structured parts in DBpedia, e.g., infoboxes and categories. Recently, the DBpedia extraction approach has been transferred to Wikis in general as well [7]. For those approaches, each page in a Wiki usually corresponds to one entity in the knowledge graph, therefore, this Wiki page can be considered a useful source of additional information about that entity. For filling missing relations (e.g., the missing birthplace of a person), *relation extraction* methods are proposed, which can be used to fill the relation based on the Wiki page's text.

Most methods for relation extraction work on text and thus usually have at least one component which is explicitly specific for the language at hand (e.g., stemming, POS tagging, dependency parsing), like, e.g., [8–10], or implicitly exploits some characteristics of that language [11]. Thus, adapting those methods to work with texts in different natural languages is usally not a straight forward process.

In this paper, we pursue an approach to extract knowledge from abstracts from Wikis. While a *Wiki* generally is a collaborative content editing platform [12], we consider everything a Wiki that has a scope (which can range from very broad, as in the case of Wikipedia, to very specific) and content pages that are interlinked, where each page has a specific topic. Practically, we limit ourselves to installations of the MediaWiki platform [13], which is the most wide-spread Wiki platform [14], although implementations for other platforms would be possible. As an abstract, we consider the contents of the Wiki page that appear in a Wiki before the first structuring element (e.g., a headline or a table of contents), as depicted in Figure 1.



Nine Inch Nails

From Wikipedia, the free encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

In other projects
Wikimedia Commons

Languages
Dansk
Deutsch
★ Español
Français
Hrvatski

Nine Inch Nails, commonly abbreviated as **NIN** (stylized as **NIИ**), is an American industrial band founded in 1988 by Trent Reznor in Cleveland, Ohio. The band released two influential albums during the 1990s—*The Downward Spiral* (1994) and *The Fragile* (1999)—and has record sales exceeding over 20 million copies worldwide,[1] with 10 million sales certified in the United States alone.[2]

Despite its success, the band has had several feuds with the corporate side of the recording industry. The first was its notable conflict with TVT Records regarding the band's first album, *Pretty Hate Machine*, which led Reznor to record his follow-up album in secrecy to avoid the label's influence. Nine Inch Nails signed to Interscope Records afterward, but after the release of the band's fifth studio album *Year Zero* in 2007, Reznor announced that Nine Inch Nails would release future material independently.[3] The band signed with Columbia Records for the release of its eighth album *Hesitation Marks* (2013), and two EPs: *Not the Actual Events* (2016) and *Add Violence* (2017).

As the band's founder and primary producer, singer, songwriter, and instrumentalist, Reznor has been considered the only constant member and creative force within the group through much of its history.[4] However, this changed in 2016 when English musician Atticus Ross was introduced as a permanent member of Nine Inch Nails, alongside the announcement of the *Not the Actual Events* EP.[5] Furthermore, Reznor typically assembles a live band to perform with him onstage. These touring bands comprise a revolving lineup, and often rearrange songs to fit a live setting. Nine Inch Nails often employs thematic visual elements to accompany on stage performances as well, which frequently include elaborate light shows.[6]

Nine Inch Nails has been nominated for thirteen Grammy Awards, winning twice for the songs "Wish" and "Happiness in Slavery" in 1992 and 1996, respectively. In 1997, Reznor appeared in *Time* magazine's list of the year's most influential people, and *Spin* magazine has described him as "the most vital artist in music".[7] In 2004, *Rolling Stone* placed Nine Inch Nails at 94 on the magazine's list of the 100 greatest artists of all time.[8] In 2014, Nine Inch Nails was named as a nominee for induction into the Rock and Roll Hall of Fame, the band's first year of eligibility.[9] In 2015, Nine Inch Nails was nominated a second time.[10]

Contents [hide]

Nine Inch Nails	
Nine Inch Nails performing in November 2013 From left to right: Pino Palladino, Ilan Rubin, Trent Reznor, Alessandro Cortini	
Background information	
Origin	Cleveland, Ohio, U.S.
Genres	Industrial rock · industrial · alternative rock · ambient · electronic rock
Years active	1988–present
Labels	Nothing · TVT · Interscope · The Null Corporation · Columbia
Associated acts	David Bowie · Exotic Birds · How to Destroy Angels · Marilyn Manson · A Perfect Circle · Pigface · Queens of the Stone Age · Tapeworm
Website	nin.com
Members	Trent Reznor Atticus Ross
Past members	List of Nine Inch Nails band members

Figure 1. An example Wikipedia page. As the abstract, we consider the beginning of a Web page before the first structuring element (here: the table of contents).

In this paper, we propose a language-agnostic approach. Instead of knowledge about the language, we take background knowledge from the a knowledge graph into account. With that, we try to discover certain patterns in how Wiki abstracts are written. For example, in many cases, any genre mentioned in the abstract about a band is usually a genre of that band, the first city mentioned in an abstract about a person is that person's birthplace, and so on. Examples for such patterns are shown in Table 1, which shows patterns observed on 10 randomly sampled pages of cities, movies, and persons from the English Wikipedia.

In that case, the linguistic assumptions that we make about a language at hand are quite minimal. In fact, we only assume that for each Wiki, there are certain ways to structure an abstract of a given type of entity, in terms of what aspect is mentioned where (e.g., the birth place is the first place mentioned when talking about a person). Thus, the approach can be considered as language-independent (see [11] for an in-depth discussion).

Table 1. Examples for patterns observed on pages for cities, movies, and persons. Those patterns were manually found on ten random pages about cities, movies, and persons.

Category	Pattern	No. Pages
City	*Region mentioned in first sentence* → `dbo:isPartOf`	9/10
City	*Country mentioned in first sentence* → `dbo:country`	4/10
Movie	*Actor mentioned in first two sentences* → `dbo:actor`	8/10
Movie	*First director mentioned* → `dbo:director`	7/10
Person	*First place mentioned* → `dbo:birthplace`	5/10
Person	*Country mentioned in first sentence* → `dbo:nationality`	3/10

The choice for Wiki abstracts as a corpus mitigates one of the common sources of errors in the relation extraction process, i.e., the entity linking. When creating a knowledge graph from Wikis, an unambiguous 1:1 mapping between entities in the knowledge graph and Wiki pages is created. Moreover, the percentage of wrong page links is usually marginal (i.e., below 1%) [15], and particularly below the error rate made by current entity linking tools [16,17], thus, the corpus at hand can be directly exploited for relation extraction without the need for an upfront potentially noisy entity linking step.

By applying the exact same pipeline without any modifications to the twelve largest languages of Wikipedia, which encompass languages from different language families, we demonstrate that such patterns can be extracted from Wikipedia abstracts in arbitrary languages. We show that it is possible to extract valuable information by combining the information extracted from different languages, and we show that some patterns even exist across languages. Furthermore, we demonstrate the usage of the approach on a less structured knowledge graph created from several thousands of Wikis, i.e., *DBkWik*.

The rest of this paper is structured as follows. In Section 2, we review related work. We introduce our approach in Section 3, and discuss various experiments on DBpedia in Section 4 and on DBkWik in Section 5. We conclude with a summary and an outlook on future work.

A less extensive version of this article has already been published as a conference paper [18]. This article extends the original in various directions, most prominently, it discusses the evidence of language-independent patterns in different Wikipedia languages, and it shows the applicability of the approach to a less well structured knowledge graph extracted from a very large number of Wikis, i.e., DBkWik [7].

2. Related Work

Various approaches have been proposed for relation extraction from text, in particular from Wikipedia. In this paper, we particularly deal with *closed* relation extraction, i.e., extracting new instantiations for relations that are defined a priori (by considering the schema of the knowledge graph at hand, or the set of relations contained therein).

Using the categorization introduced in [6], the approach proposed in this paper is an *external* one, as it uses Wikipedia as an external resource in addition to the knowledge graph itself. While internal approaches for relation prediction in knowledge graphs exist as well, using, e.g., association rule mining, tensor factorization, or graph embeddings, we restrict ourselves to comparing the proposed approach to other external approaches.

Most of the approaches in the literature make more or less heavy use of language-specific techniques. Distant supervision is proposed by [19] as a means to relation extraction for Freebase from Wikipedia texts. The approach uses a mixture of lexical and syntactic features, where the latter are highly language-specific. A similar approach is proposed for DBpedia in [20]. Like the Freebase-centric approach, it uses quite a few language-specific techniques, such as POS tagging and lemmatization. While those two approaches use Wikipedia as a corpus, [21] compare that corpus to a corpus of news texts, showing that the usage of Wikipedia leads to higher quality results.

Nguyen et al. [22] introduce an approach for mining relations from Wikipedia articles which exploits similarities of dependency trees for extracting new relation instances. In [23], the similarity of dependency trees is also exploited for clustering pairs of concepts with similar dependency trees. The construction of those dependency trees is highly language specific, and consequently, both approaches are evaluated on the English Wikipedia only.

An approach closely related to the one discussed in this paper is *iPopulator* [24], which uses Conditional Random Fields to extract patterns for infobox values in Wikipedia abstracts. Similarly, *Kylin* [25] uses Conditional Random Fields to extract relations from Wikipedia articles and general Web pages. Similarly to the approach proposed in this paper, *PORE* [26] uses information on neighboring entities in a sentence to train a support vector machine classifier for the extraction of four different relations. The papers only report results for English language texts.

Truly language-agnostic approaches are scarce. In [27], a multi-lingual approach for open relation extraction is introduced, which uses Google translate to produce English language translations of the corpus texts in a preprocessing step, and hence exploits externalized linguistic knowledge. In the recent past, some approaches based on deep learning have been proposed which are reported to or would in theory also work on multi-lingual text [28–31]. They have the advantages that (a) they can compensate for shortcomings in the entity linking step when using arbitrary text and (b) that explicit linguistic feature engineering is replaced by implicit feature construction in deep neural networks. In contrast to those works, we work with a specific set of texts, i.e., Wikipedia abstracts. Here, we can assume that the entity linking is mostly free from noise (albeit not complete), and directly exploit knowledge from the knowledge graph at hand, i.e., in our case, DBpedia.

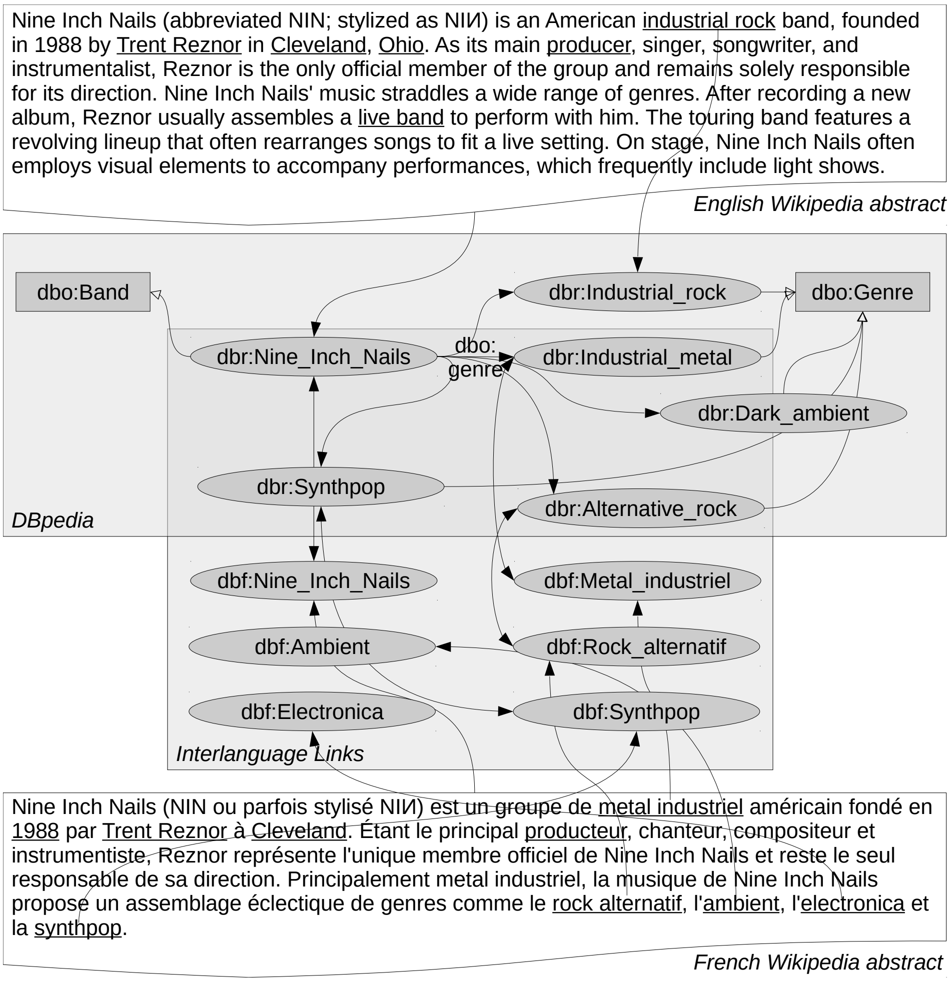
In contrast to most of those works, the approach discussed in this paper works on Wikipedia abstracts in *arbitrary* languages, which we demonstrate in an evaluation using the twelve largest language editions of Wikipedia. While, to the best of our knowledge, most of the approaches discussed above are only evaluated on one or at maximum two languages, this is the first approach to be evaluated on a larger variety of languages. Furthermore, while many works in the area evaluate their results using only on one underlying knowledge graph [6], we show results with two differently created knowledge graphs and two different corpora.

3. Approach

Our aim is to identify and exploit typical patterns in Wiki abstracts. As a running example in this section, we use the *genre* relation which may hold between a music artist and a music genre, and our example depict the DBpedia knowledge graph and the various language editions of Wikipedia.

DBpedia is created from Wikipeda using an *ontology* edited in a collaborative process, and a set of *mappings* that link infoboxes used in Wikipedia to that ontology, which are also crowd sourced. During the extraction process, an entity is created for each page in Wikipedia, and the mappings are used to create statements about those entities. The result is a knowledge graph for multiple language editions of Wikipedia [1].

Figure 2 depicts this example with both an English and a French Wikipedia abstract. As our aim is to mine relations for the canonical DBpedia, extracted from the (largest) English language Wikipedia, we inspect all links in the abstract which have a corresponding entity in the main DBpedia knowledge graph created from the English Wikipedia. For this work, we use the 2014 version of DBpedia [32], which was the most recent release available at the time the experiments were conducted. All statements made in this paper about the size etc. of DBpedia correspond to that version. For other languages, we take one intermediate step via the interlanguage links in Wikipedia, which are extracted as a part of DBpedia [1]. Note that in our experiments, the goal is to enhance the canonical, i.e., English, DBpedia using texts in *multiple* languages. If the goal was to extend a knowledge graph in a given language (e.g., French), we could omit the links to the canonical DBpedia and work directly with the French knowledge graph. Apart from omitting the links, the approach would be identical.



Nine Inch Nails (abbreviated NIN; stylized as NIИ) is an American industrial rock band, founded in 1988 by Trent Reznor in Cleveland, Ohio. As its main producer, singer, songwriter, and instrumentalist, Reznor is the only official member of the group and remains solely responsible for its direction. Nine Inch Nails' music straddles a wide range of genres. After recording a new album, Reznor usually assembles a live band to perform with him. The touring band features a revolving lineup that often rearranges songs to fit a live setting. On stage, Nine Inch Nails often employs visual elements to accompany performances, which frequently include light shows.

Nine Inch Nails (NIN ou parfois stylisé NIИ) est un groupe de metal industriel américain fondé en 1988 par Trent Reznor à Cleveland. Étant le principal producteur, chanteur, compositeur et instrumentiste, Reznor représente l'unique membre officiel de Nine Inch Nails et reste le seul responsable de sa direction. Principalement metal industriel, la musique de Nine Inch Nails propose un assemblage éclectique de genres comme le rock alternatif, l'ambient, l'electronica et la synthpop.

Figure 2. Approach illustrated with extraction from English (**above**) and French (**below**) Wikipedia abstract.

3.1. Overall Approach

Our focus are relations between entities, i.e., *object properties* in OWL [33]. *Datatype properties* relating an entity and a literal value (e.g., a number or a date) are out of scope here. Entities in a knowledge graph may, but need not have a type (e.g., an entity may be of type *Person*, *Place*, etc. or not have a type at all, due to incompleteness or other reasons). For object relations, the ontology may define an explicit domain and range, i.e., types of entities that are allowed in the subject and object position. For example, the relation *genre* is used to relate artists to genres.

Furthermore, we assume that each entity has a corresponding Wiki page that describes the entity—for knowledge graphs created from Wikis, this is usually the Wiki page from which the entity was created. We consider a pair of Wiki pages p_0 and p_1, where p_1 is linked from the abstract of p_0, as a *candidate* for a relation R if the corresponding DBpedia entities e_0 and e_1 have types that match the domain and range of R. In that case, $R(e_0, e_1)$ is considered a candidate axiom to be included in the knowledge graph. In the example in Figure 2, given that the *genre* relation holds between musical artists and genres, and the involved entities are of the matching types, one candidate each is generated from both the English and the French DBpedia (Prefixes used in this paper: `dbr=http://dbpedia.org/`, `dbf=http://fr.dbpedia.org/`, `dbo=http://dbpedia.org/ontology/`).

We expect that candidates contain a lot of false positives. For example, for the *birthplace* relation holding between a person and a city, all cities linked from the person's corresponding Wiki page would be considered candidates. However, cities may be referred to for various different reasons in an abstract about a person (e.g., they may be their death place, the city of their alma mater, etc.). Thus, we require additional evidence to decide whether a candidate actually represents a valid instantiation of a relation.

To make that decision, we train a *machine learning model*. For each abstract of a Wiki page for which a given relation is present in the knowledge graph, we use the *partial completeness assumption* [34] or *local closed world assumption* [35], i.e., we consider the relation to be complete. Hence, all candidates for the relation created from the abstract which are contained in the knowledge graph are considered as *positive* training examples, all those which are not contained are considered as *negative* training examples. In the example in Figure 2, *Industrial Rock* would be considered a positive example for the relation *genre*, whereas the genre *Rock*, if it were linked in the abstract, would be considered a negative example, since it is not linked as a genre in the underlying knowledge graph.

3.2. Feature Engineering

For training a classifier, both positive and negative examples need to be described by *features*. Table 2 sums up the features used by the classifiers proposed in this paper.

We use features related to the actual candidates found in the abstract (i.e., entities whose type matches the range of the relation at hand), i.e., the total number of candidates in the abstract (F00) and the candidate's sentence (F01), the position of the candidate w.r.t. all other candidates in the abstract (F02) and the candidate's sentence (F03), as well as the position of the candidate's sentence in the abstract (F07). The same is done for all entities, be it candidates or not (F04,F05,F06). Since all of those measures yield positive integers, they are normalized to $(0, 1]$ by using their inverse.

Further features taken into account are the existence of a back link from the candidate's page to the abstract's page (F08), and the vector of all the candidate's types in the knowledge graph's ontology (FXX). The subject's types are not utilized. For DBpedia and DBkWik (and unlike YAGO), they only exist if the subject has an infobox, which would make the approach infeasible to use for long tail entities for which the Wiki page does not come with an infobox. Table 3 depicts the translated feature table for the French Wikipedia abstract depicted in Figure 2. In this example, there are five candidates (i.e., entities of type `Genre`), three of which are also contained in the DBpedia knowledge graph (i.e., they serve as true positives).

With the help of a feature vector representation, it is possible to learn fine-grained classification models, such as *The first three genres mentioned in the first or second sentence of a band abstract are genres of that band.*

Table 2. List of features used by the classifier.

ID	Name	Range	ID	Name	Range
F00	NumberOfCandidates	(0, 1]	F05	EntityPosition	(0, 1]
F01	CandidatesInSentence	(0, 1]	F06	EntityPositionInSentence	(0, 1]
F02	CandidatePosition	(0, 1]	F07	SentencePosition	(0, 1]
F03	CandidatePositionInSentence	(0, 1]	F08	BackLink	T/F
F04	EntitiesInSentence	(0, 1]	TXX	InstanceTypes	T/F

Table 3. Example Feature Representation.

Instance	F00	F01	F02	F03	F04	F05	F06	F07	F08	`T:Genre`	`T:Place`	`T:Band`	...	Correct
Industrial_Metal	0.2	1.0	1.0	1.0	0.25	1.0	1.0	1.0	1.0	true	false	false	...	true
Alternative_Rock	0.2	0.2	0.5	1.0	0.25	0.2	1.0	0.3	1.0	true	false	false	...	true
Ambient_music	0.2	0.2	0.3	0.5	0.25	0.1	0.5	0.3	0.0	true	false	false	...	false
Electronica	0.2	0.2	0.25	0.25	0.3	0.1	0.3	0.3	0.0	true	false	false	...	false
Synthpop	0.2	0.2	0.2	0.25	0.25	0.1	0.25	0.3	0.0	true	false	false	...	true

3.3. Machine Learning Algorithms

Initially, we experimented with a set of five classification algorithms, i.e., Naive Bayes, RIPPER [36], Random Forest (RF) [37], Neural Networks [38] and Support Vector Machines (SVM) [39]. For all those classifiers, we used the implementation in *RapidMiner* [40] and, for the preliminary evaluation, all classifiers were used in their standard setup.

In order to choose a classifier for the subsequent experiments, we conducted an initial study on DBpedia and the English Wikipedia: for the five classifiers above, we used samples of size 50,000 from the ten most frequent relations in DBpedia, the corresponding English language abstracts, and performed an experiment in ten-fold cross validation. The results are depicted in Table 4. We can observe that the best results in terms of F-measure are achieved by Random Forests, which has been selected as the classifier to use in the subsequent experiments.

Table 4. Pre-Study results on DBpedia with five machine learning algorithms.

Relation	Naive Bayes			Rand.For.			RIPPER			Neural Net			SVM		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
dbo:birthPlace	0.69	0.65	0.67	0.69	0.76	0.72	0.72	0.73	0.72	0.61	0.75	0.67	0.72	0.74	0.73
dbo:family	0.55	0.93	0.69	0.87	0.83	0.85	0.85	0.83	0.84	0.77	0.83	0.80	0.87	0.83	0.85
dbo:deathPlace	0.42	0.30	0.35	0.51	0.30	0.38	0.64	0.18	0.28	0.61	0.19	0.29	0.66	0.20	0.31
dbo:producer	0.35	0.55	0.43	0.35	0.14	0.20	0.47	0.04	0.07	0.23	0.10	0.14	0.48	0.05	0.09
dbo:writer	0.55	0.61	0.58	0.62	0.55	0.58	0.64	0.54	0.59	0.52	0.51	0.51	0.67	0.53	0.59
dbo:subsequentWork	0.11	0.21	0.14	0.35	0.10	0.16	0.42	0.02	0.04	0.21	0.07	0.11	0.61	0.06	0.11
dbo:previousWork	0.18	0.43	0.25	0.39	0.18	0.25	0.59	0.05	0.09	0.57	0.08	0.14	0.60	0.10	0.17
dbo:artist	0.94	0.94	0.94	0.94	0.95	0.94	0.95	0.96	0.95	0.95	0.86	0.90	0.95	0.89	0.92
dbo:nationality	0.76	0.90	0.82	0.76	0.92	0.83	0.77	0.91	0.83	0.72	0.81	0.76	0.77	0.92	0.84
dbo:formerTeam	0.79	0.74	0.76	0.85	0.88	0.86	0.85	0.88	0.86	0.82	0.77	0.79	0.85	0.89	0.87
Average	0.53	0.63	0.56	0.63	0.56	0.58	0.69	0.51	0.53	0.60	0.50	0.51	0.72	0.52	0.55

Furthermore, we compared the machine learning approach to four simple baselines using the same setup:

Baseline 1 The first entity with a matching type is classified as a positive relation, all others as negative.
Baseline 2 All entities with a matching type are classified as positive relations.
Baseline 3 The first entity with a matching ingoing edge is classified as a positive relation. For example, when trying to extract relations for `dbo:birthPlace`, the first entity which already has one ingoing edge of type `dbo:birthPlace` would be classified as positive.
Baseline 4 All entities with a matching ingoing edge are classified as positive relations.

The results of the baseline evaluations are depicted in Table 5. We can observe that in terms of F-measure, they are outperformed by RandomForest. Although the margin seems small, the baseline approaches usually have a high recall, but low precision. In fact, none of them reaches a precision above 0.5, which means that by applying such approaches, at least half of the relations inserted into a knowledge graph would be noise.

Table 5. Results of the four baselines on DBpedia.

Relation	Baseline 1			Baseline 2			Baseline 3			Baseline 4		
	P	R	F	P	R	F	P	R	F	P	R	F
dbo:birthPlace	0.47	0.99	0.64	0.46	1.00	0.63	0.49	0.98	0.66	0.48	0.99	0.65
dbo:family	0.18	0.85	0.30	0.17	1.00	0.29	0.87	0.84	0.86	0.86	1.00	0.92
dbo:deathPlace	0.28	0.97	0.43	0.27	1.00	0.43	0.30	0.93	0.46	0.30	0.96	0.46
dbo:producer	0.17	0.93	0.29	0.15	1.00	0.26	0.33	0.80	0.47	0.32	0.87	0.46
dbo:writer	0.41	0.69	0.52	0.19	1.00	0.32	0.56	0.59	0.58	0.45	0.86	0.59
dbo:subsequentWork	0.02	1.00	0.04	0.02	1.00	0.04	0.02	0.19	0.03	0.02	0.19	0.03
dbo:previousWork	0.04	1.00	0.08	0.04	1.00	0.08	0.04	0.20	0.06	0.03	0.20	0.06
dbo:artist	0.31	0.99	0.47	0.27	1.00	0.42	0.57	0.87	0.69	0.53	0.87	0.66
dbo:nationality	0.73	0.96	0.83	0.64	1.00	0.78	0.74	0.96	0.84	0.64	1.00	0.78
dbo:formerTeam	0.35	0.72	0.47	0.40	1.00	0.57	0.78	0.70	0.74	0.81	0.98	0.89
Average	0.30	0.91	0.41	0.26	1.00	0.38	0.47	0.71	0.54	0.44	0.79	0.55

4. Experiments on DBpedia

To validate the approach on DBpedia, we conducted different experiments to validate the approach. First, we analyzed the performance of the relation extraction using a RandomForest classifier on the English DBpedia only. The classifier is used in its standard setup, i.e., training 20 trees. The code for reproducing the results is available online [41].

For the evaluation, we follow a two-fold approach: for once, we use a cross-validated silver standard evaluation, where we evaluate how well existing relations can be predicted for instances already present in DBpedia. Since such a silver-standard evaluation can introduce certain biases [6], we additionally validate the findings on a subset of the extracted relations in a manual retrospective evaluation.

In a second set of experiments, we analyze the extraction of relations on the twelve largest language editions of Wikipedia, which at the same time are those with more than 1 M articles, i.e., English, German, Spanish, French, Italian, Dutch, Polish, Russian, Cebuano, Swedish, Vietnamese, and Waray [42]. The datasets of the extracted relations for all languages can be found online [41]. Note that this selection of languages does not only contain Indo-European, but also one Austronesian and two Austroasiatic languages.

In addition, we conduct further analyses. First, we investigate differences of the relations extracted for different languages with respect to topic and locality. For the latter, the hypothesis is that information extracted, e.g., for places from German abstracts is about places in German speaking countries. Finally, with RIPPER being a symbolic learner, we also inspect the models as such to find out whether there are universal patterns for relations that occur in different languages.

For 395 relations that can hold between entities, the ontology underlying the DBpedia knowledge graph [43] defines an explicit domain and range, i.e., the types of objects that are allowed in the subject and object position of this relation. Those were considered in the evaluation.

4.1. Experiments on English Abstracts

In a first set of experiments, we analyzed the performance of our method on English abstracts only. Since we aim at augmenting the DBpedia knowledge graph at a reasonable level of precision, our aim was to learn models which reach a precision of at least 95%, i.e., that add statements with no more than 5% noise to the knowledge graph. Out of the 395 relations under inspection, the RandomForest classifier could learn models with a precision of 95% or higher for 99 relations. For the 99 models that RF could learn with a minimum precision of 95%, the macro (micro) average recall and precision are 31.5% (30.6%) and 98.2% (95.7%), respectively.

Figure 3 depicts the results for the models ordered by precision. It shows that while meaningful models can be learned for more than two thirds of the relations (at varying degrees of precision), there is also quite a large number of relations for which the maximum precision that can be achieved is 0. There are a different reasons for zero precision models, including

- Relations that are expressed as a link between two Wikipedia pages, but use an intermediate RDF node in the knowledge graph. One example is `dbo:careerStation`, which uses an intermediate node of type `dbo:CareerStation` to link an athlete to a club, while the link is directly between the athlete's and the club's Web page in Wikipedia. Since the intermediate node has no counterpart in Wikipedia, there are no training examples for the relation.
- Relations which are defined in the DBpedia ontology, but are not instantiated in the knowledge graph, such as `dbo:fundedBy`. For those relations, no positive training examples exist.
- Relations for which the class defined for the domain or range is empty. For example, `dbo:ideology` has the range `dbo:Ideology`, but that class has no instances in DBpedia.
- Relations which are not important enough to be mentioned in the abstract, such as the `dbo:endingTheme` of a TV series.

- Relations which are often used in disagreement with the underlying ontology, such as `dbo:instrument` (which is often used to link an instrument to a music genre, but intended to be used to link an instrument to a person [44]).

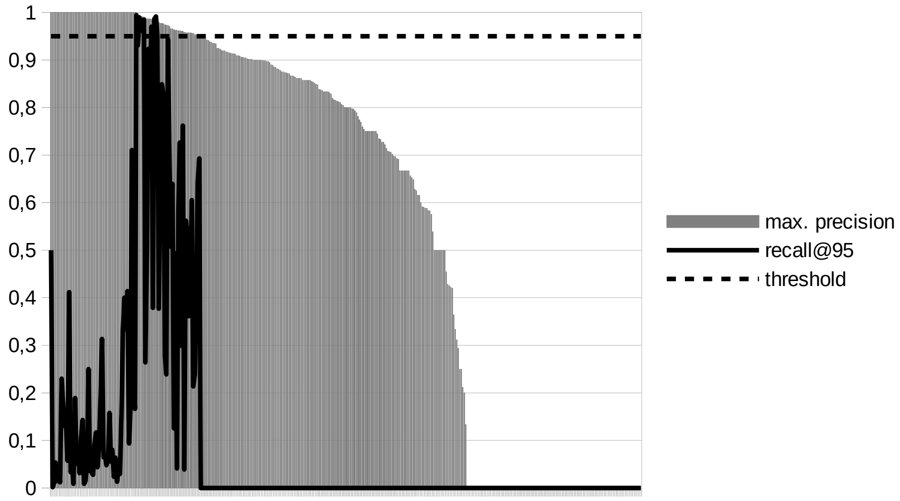


Figure 3. Depiction of precision and recall@95 for the models created, ordered by precision.

The same figure also yields another interesting observations: the highest precision models often have a low recall@95, i.e., it is possible to learn high precision models for some relations, but only at a low recall. Again, there are multiple reasons here, including

- Relations that are not often mentioned in the abstract, but easy to extract, such as the `dbo:coverArtist` of a book.
- Relations that have incomplete information in the abstract, such as the athletes playing for a team (typically, the abstract will mention the most famous, if at all).
- Relations that are overall very infrequent, such as `dbo:confluenceState`.

By applying the 99 models to all candidates, a total of 998,993 new relation instances could be extracted, which corresponds to roughly 5% of all candidates. Figure 4 depicts the 20 relations for which most instances are extracted.

For validating the precision and recall scores computed on the existing relation instances, we sampled each 200 *newly* generated from five relations (i.e., 1,000 in total) and validated them manually. For the selection of entities, we aimed at a wider coverage of common topics (geographic entities, people, books, music works), as well as relations which can be validated fairly well without the need of any specific domain knowledge. The results are depicted in Table 6. The validation confirms the high precision of the learned models.

Table 6. Results of the manual verification of precision and recall scores computed on the existing relation instances. R_e and P_e denotes the recall and precision of the models computed on the existing relation instances, while R_m and P_m denotes those verified by manual computation.

Relation	R_e	P_e	R_m	P_m
dbo:musicalBand	96.2	95.1	87.9	96.7
dbo:author	68.2	95.2	53.4	91.9
dbo:department	64.5	99.5	53.5	93.7
dbo:sourceCountry	98.9	98.0	98.8	97.8
dbo:saint	41.2	100.0	53.3	95.5

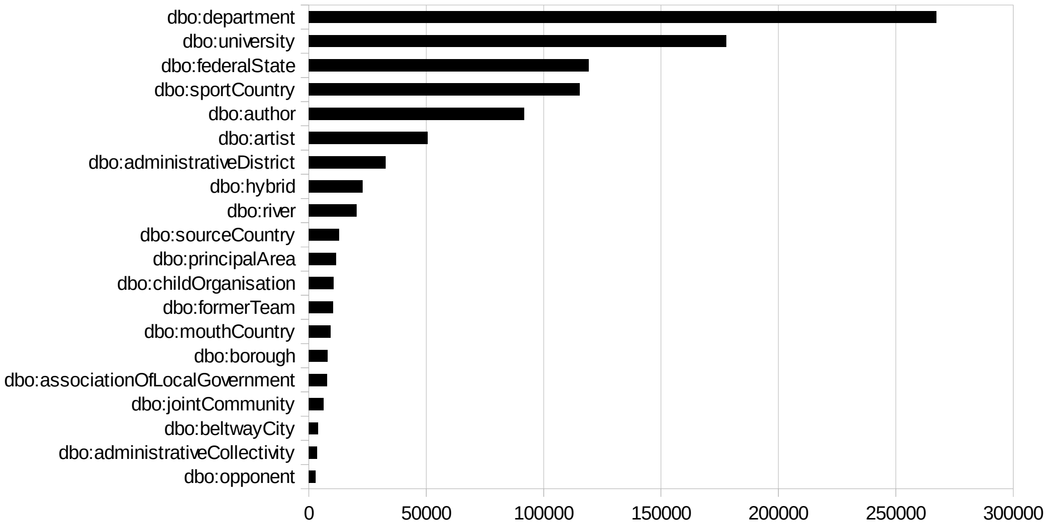


Figure 4. 20 most frequent relations extracted from English abstracts.

4.2. Cross-Lingual Relation Extraction

In the next experiment, we used the RandomForests classifier to extract models for relations for the top 12 languages, as depicted in Table 7. One model is trained per relation and language. The feature extraction and classifier parameterization is the same as in the previous set of experiments.

Table 7. Size of the 12 largest language editions of Wikipedia, and percentage of articles linked to English.

Language	# Entities	% Links to English
English	4,192,414	100.00
Swedish	2,351,544	17.60
German	1,889,351	42.21
Dutch	1,848,249	32.98
French	1,708,934	51.48
Cebuano	1,662,301	5.67
Russian	1,277,074	42.61
Waray	1,259,540	12.77
Italian	1,243,586	55.69
Spanish	1,181,096	54.72
Polish	1,149,530	53.70
Vietnamese	1,141,845	28.68

As a first result, we look at the number of relations for which models can be extracted at 95% precision. While it is possible to learn extraction models for 99 relations at that level of precision for English, that number almost doubles to 187 when using the top twelve languages, as depicted in Figure 5. These results show that it is possible to learn high precision models for relations in other languages for which this is not possible in English.

When extracting new statements (i.e., instantiations of the relations) using those models, our goal is to extract those statements in the canonical DBpedia knowledge graph, as depicted in Figure 2. The number of extracted statements per language, as well as cumulated statements, is depicted in Figure 5.

At first glance, it is obvious that, although a decent number of models can be learned for most languages, the number of statements extracted are on average an order of magnitude smaller than the number of statements that are extracted for English. However, the additional number of extracted relations is considerable: while for English only, there is roughly 1 M relations, 1.6 M relations can be extracted from the top 12 languages, which is an increase of about 60% when stepping from an

English-only to a multi-lingual extraction. The graphs in Figure 5 also shows that the results stabilize after using the seven largest language editions, i.e., we do not expect any significant benefits from adding more languages with smaller Wikipedias to the setup. Furthermore, with smaller Wikipedias, the number of training examples also decreases, which also limits the precision of the models learned.

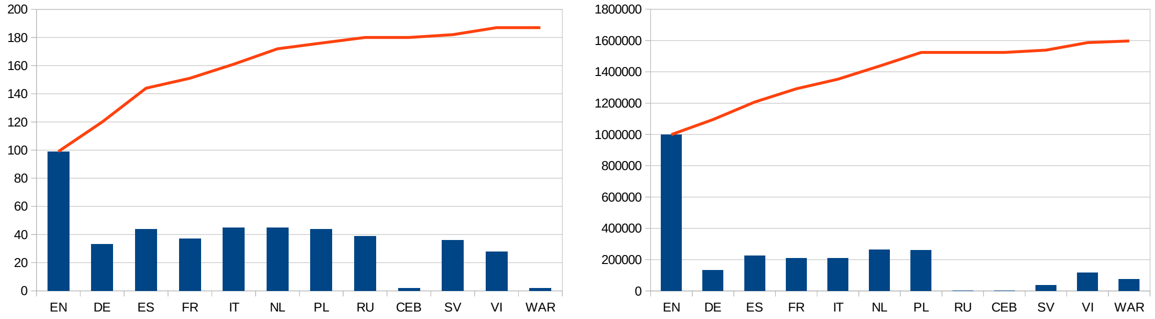


Figure 5. Number of relations (**left**) and statements (**right**) extracted at 95% precision in the top 12 languages. The bars show the number of statements that could be extracted for the given language, the line depicts the accumulated number of statements for the top N languages.

As can be observed in Figure 5, the number of extracted statements is particularly low for Russian and Cebuano. For the latter, the figure shows that only a small number of high quality models can be learned, mostly due to the low number of inter-language links to English, as depicted in Table 7. For the former, the number of high quality models that can be learned is larger, but the models are mostly unproductive, since they are learned for rather exotic relations. In particular, for the top 5 relations in Figure 4, no model is learned for Russian.

It is evident that the number of extracted statements is not proportional to the relative size of the respective Wikipedia, as depicted in Table 7. For example, although the Swedish Wikipedia is more than half the size of the English one, the number of extracted statements from Swedish is by a factor of 28 lower than those extracted from English. At first glance, this may be counter intuitive.

The reason for the number of statements extracted from languages other than English is that we only generate candidates if both the article at hand and the entity linked from that article's abstract have a counterpart in the canonical English DBpedia. However, as can be seen from Table 7, those links to counterparts are rather scarce. For the example of Swedish, the probability of an entity being linked to the English Wikipedia is only 0.176. Thus, the probability for a candidate that both the subject and object are linked to the English Wikipedia is $0.176 \times 0.176 = 0.031$. This is pretty exactly the ratio of statements extracted from Swedish to statements extracted from English (0.036). In fact, the number of extracted statements per language and the squared number of links between the respective language edition and the English Wikipedia have a Pearson correlation coefficient of 0.95. This shows that the low number of statements is mainly an effect of missing inter-language links in Wikipedia, rather than a shortcoming of the approach as such.

If we were interested in extending the coverage of DBpedia not only w.r.t. relations between existing entities, but also adding *new* entities (in particular: entities which only exist in language editions of Wikipedia other than English), then the number of statements would be larger. However, this was not in the focus of this work.

It is remarkable that the three Wikis in our evaluation with the lowest interlinkage degree—i.e., Swedish, Cebuano, and Waray—are known to be created by a bot called *Lsjbot* to a large degree [45]. Obviously, this bot creates a large number of pages, but at the same time fails to create the corresponding interlanguage links.

4.3. Topical and Geographical Analysis by Language

To further analyze the extracted statements, we look at the topical and geographical coverage for the *additional* statements (i.e., statements that are not yet contained in DBpedia) that are extracted for the twelve languages at hand. First, we depict the most frequent relations and subject classes for the statements. The results are depicted in Figures 6 and 7. It can be observed that the majority of statements is related to geographical entities and their relations. The Russian set is an exception, since most extracted relations are about musical works, in contrast to geographic entities, as for the other languages. Furthermore, the English set has the largest fraction of person related facts.

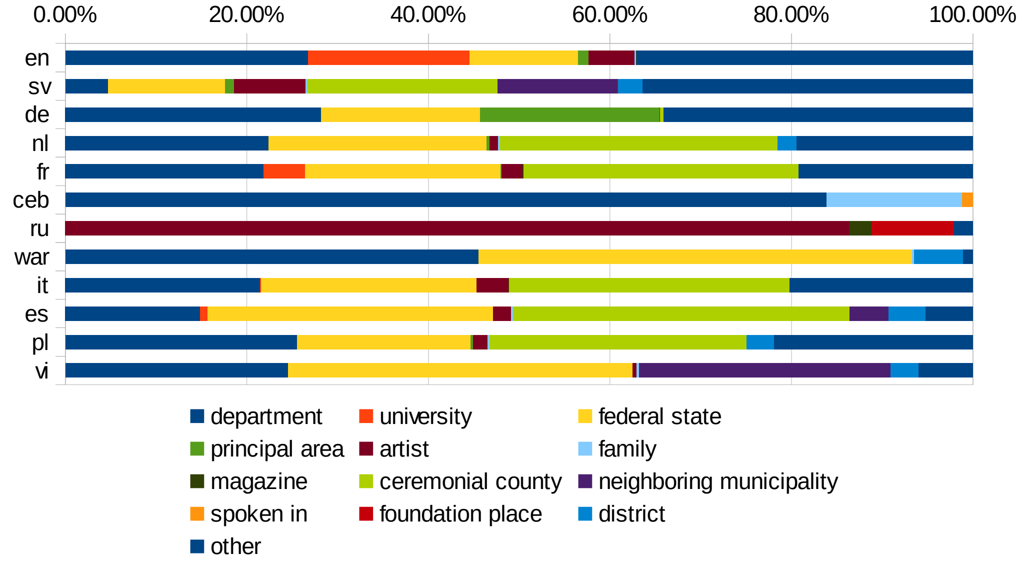


Figure 6. Distribution of relations in the different language extractions.

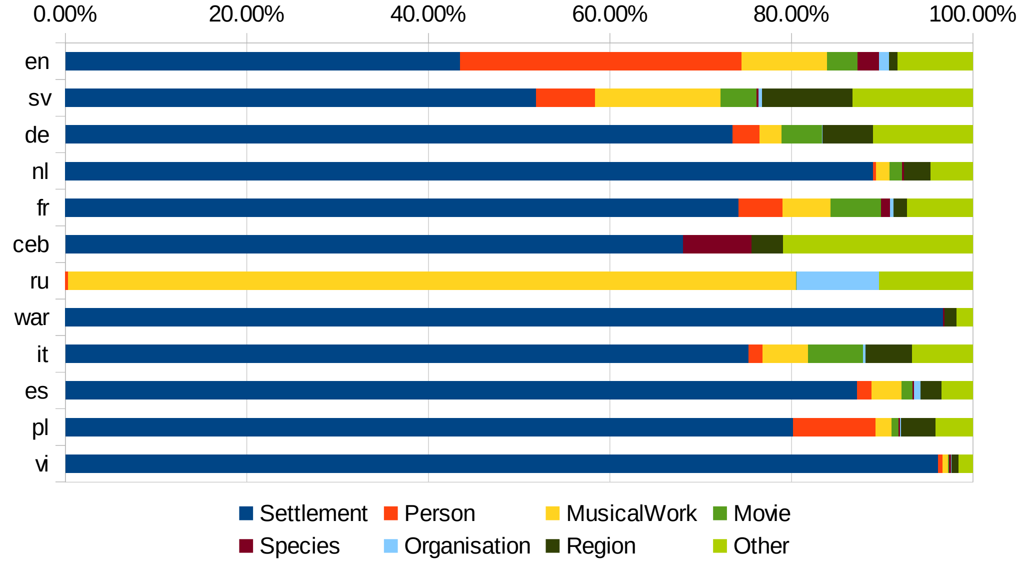


Figure 7. Distribution of subject types in the different language extractions.

We assume that the coverage of Wikipedia in different languages is, to a certain extent, biased towards places, persons, etc. from countries in which the respective language is spoken [46]. Thus, we expect that, e.g., for relations extracted about places, we will observe that the distribution of countries to which entities are related differs for the various language editions.

To validate this hypothesis, we determine the country to which a statement is related as follows: given a statement s in the form

```
s p o .
```

we determine the set of pairs $P_s := < r, c >$ of relations and countries that fulfill

```
s r c .
c a dbo:Country .
```

and

```
o r c .
c a dbo:Country .
```

For all statements S extracted from a language, we sum up the relative number of relations of a country to each statement, i.e., we determine the weight of a country C as

$$w(C) := \sum_{s=1}^{|S|} \frac{|\{< r, c > \in P_s | c = C\}|}{|P_s|} \tag{1}$$

The analysis was conducted using the RapidMiner Linked Open Data Extension [47].

Figure 8 depicts the distributions for the countries. We can observe that while in most cases, facts about US related entities are the majority, only for Polish, entities related to Poland are the most frequent. For Swedish, German, French, Cebuano and Italian, the countries with the largest population speaking those languages (i.e., Sweden, Germany, France, Philippines, and Italy, respectively), are at the second position. For Spanish, Spain is at the second position, despite Mexico and Colombia (rank 11 and 6, respectively) having a larger population. For the other languages, a language-specific effect is not observable: for Dutch, the Netherlands are at rank 8, for Vietnamese, Vietnam is at rank 34, for Waray, the Philippines are at rank 7. For Russian, Russia is on rank 23, preceded by Soviet Union (sic!, rank 15) and Belarus (rank 22).

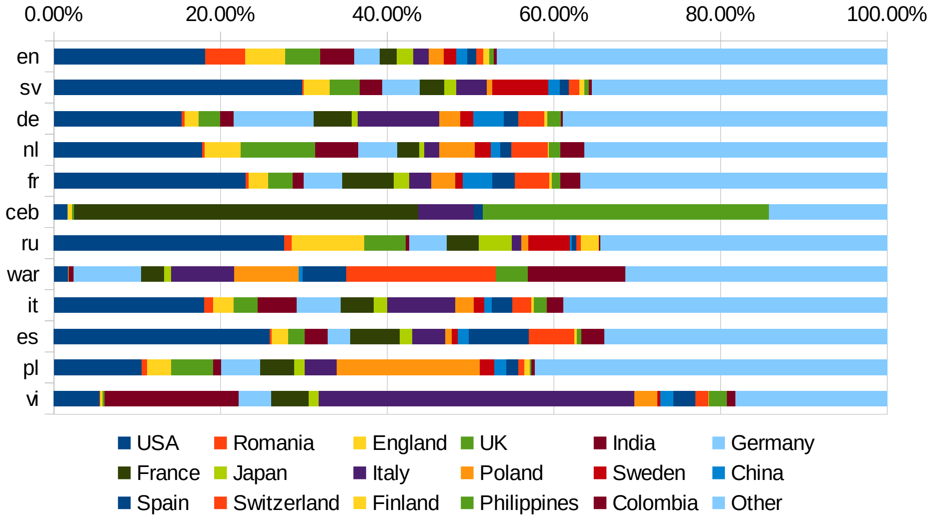


Figure 8. Distribution of locality in the different language extractions.

The results show that despite the dominance of US-related entities, there is a fairly large variety in the geographical coverage of the information extracted. This supports the finding that adding information extracted from multiple Wikipedia language editions helps broadening the coverage of entities.

4.4. Analysis of Cross-Lingual Patterns

For the analysis, we look at the relations which are extracted in *most* languages. Those relations are depicted in Table 8.

Table 8. Most frequently extracted relations across the top 12 languages.

Relation	No. of Languages
department	11
musicalBand	10
family	9
president	9
derivative	8
veneratedIn	7
majorIsland	7
operator	7
guest	6
crosses	6

In order to analyze how similar the relations are, we stepped from Random Forests to a more symbolic learner, i.e., the rule learning algorithm *RIPPER* [36], which had delivered good results in the pre-study as well (see Section 3.3). RIPPER learns only disjunctive rules, where each condition is a simple comparison of an attribute and an atomic value.

To measure the similarity of two models, we use the overlap of conditions in the first rule (which covers the majority of examples) of the different languages. We compute the average probability of a condition to be contained in the first rule of another language model as well and use that average probability as a similarity measure between the RIPPER rule models learned for different languages.

From the twelve relations depicted in Table 8, RIPPER was able to extract a model for more than one language in seven cases. The similarities of the models are depicted in Table 9.

Table 9. Cross-language model similarity for different relations, averaged across the 12 most frequently extracted relations and all language pairs.

Relation	Similarity
crosses	0.286
department	0.320
derivative	0.217
family	0.139
majorIsland	0.000
musicalBand	0.741
operator	0.292

One example with a strong cross-language pattern is the *musicalBand* relation, the first rule is completely identical for six out of ten models learned. This rule is

```
   PositionOfSentence <= 0.5
&& !BackLink
=> false
```

Note that this is a rule for the negative case, i.e., a candidate *not* being a valid statement.

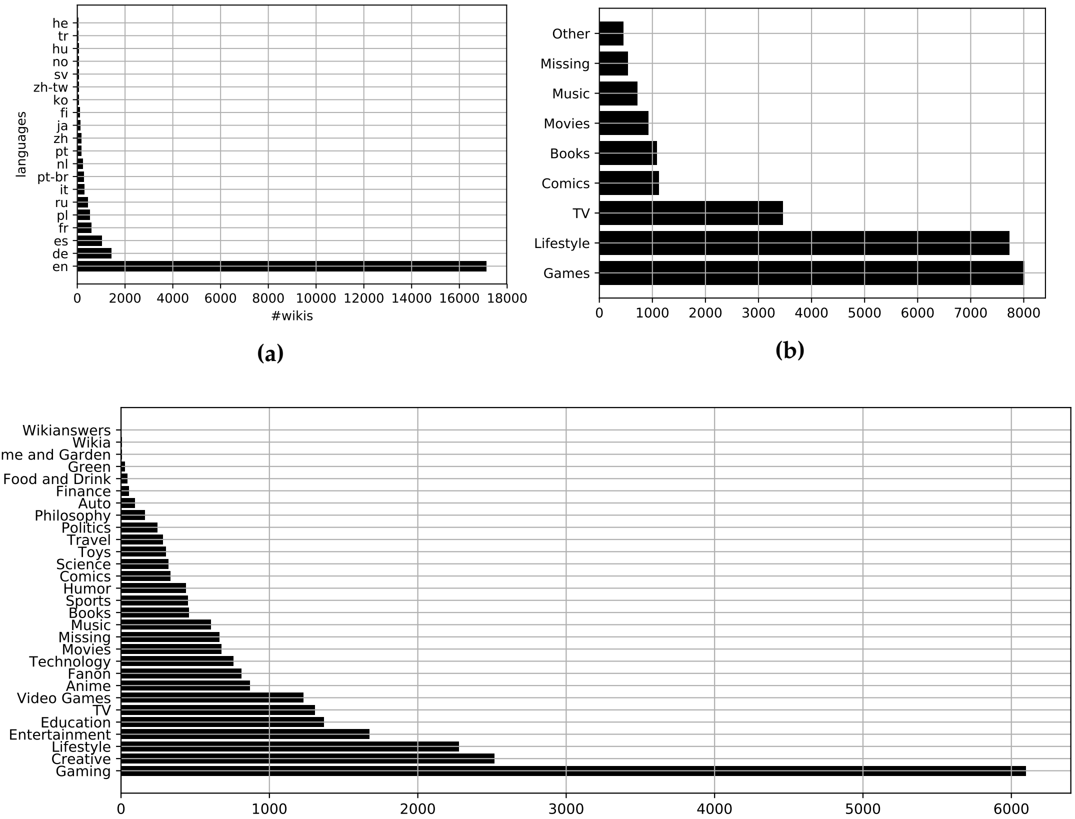
5. Experiments on DBkWik

A second series of experiments was conducted on DBkWik, a knowledge graph created from thousands of Wikis [7]. The idea behind DBkWik is that the DBpedia Extraction Framework [48] is considered a black box which takes a MediaWiki [13] dump as input, and outputs a knowledge graph. MediaWiki is the software platform that Wikipedia runs on, but Wikipedia is by for not the only Wiki which uses MediaWiki: in fact, WikiApiary reports more than 20,000 public installations of MediaWiki [49].

One of those installations is *Fandom powered by Wikia* [50], a popular *Wiki Farm* [51] hosting more than 400,000 individual Wikis with more than 50M individual pages.

5.1. Dataset

The current version of DBkWik [52] is built from Wiki dumps offered by Fandom. The Wikis are in multiple languages, and they come with two orthogonal topical classifications, called *topics* and *hubs*, where the former are more fine grained than the latter. In total, there are 24,032 Wikis that have a dump (since Wiki dumps need to be actively created by the Wiki owner, there is only a small fraction of Wikis that actually come with a dump). For those Wikis, a language and two topical classifications—a broad *hub* and a finer-grained *topic*—are defined. Figure 9 shows a breakdown by language, topic, and hub. It can be observed that the majority of the Wikis is in English, with more than 1000 in Spanish and German, respectively. Topic-wise, games and entertainment related Wikis are predominant.



(a) **(b)**

(c)

Figure 9. Breakdown of the downloaded Wikis by Language, Topic, and Hub. (**a**) Languages; (**b**) Hubs; (**c**) Topics.

In total, we were able to download the dumps for 12,480 Wikis from Fandom, since the links for about half the dumps, which are hosted on Amazon S3, were broken. After those were digested by the DBpedia Extraction Framework, the resulting knowledge graph has 13,258,243 entities and 54,039,699 relation assertions in total. However, these numbers may include duplicates, since entities from different Wikis related to the same real world entity are currently not matched. The dataset we used for the experiments in this paper is available online [53].

Although they are both created by the same piece of software, there are some crucial differences between DBkWik and DBpedia. First, a knowledge graph created from Wikipedia can be considered as being free from duplicates, i.e., there are no two entities in the knowledge graph denoting the same real world entity. Each entity in DBpedia is created from one Wikipedia page, and in Wikipedia, we can assume that for a real world entity (e.g., a person, a movie, etc.), there is exactly one page that has this entity as its main topic, and that different language editions are interlinked so that duplicates in different language Wikipedias can be easily resolved using a `owl:sameAs` link. As a result, DBpedia is practically free from duplicate entities.

When using an arbitrary collection of Wikis, as for DBkWik this assumption is not met—a page for the same entity may exist in multiple Wikis, and links between those Wikis are rare. Hence, the dataset contains an (unknown) number of duplicates, which cannot be trivially resolved. For example, the collection of Wikis that we processed encompasses around 80 Harry Potter Wikis, hence, we can assume the entity *Harry Potter* to appear multiple times in DBkWik.

Second, there is no central ontology, and no mappings between infoboxes and that ontology. Hence, we have to create an ontology on the fly. We do so by assigning class to each infobox type used in each Wiki, and a property to each infobox key used in each Wiki. The resulting ontology is very informal, i.e., it defines only classes and relations, but no additional axioms, in particular no domain and range restrictions. It consists of 71,584 classes and 506,512 relations. Categories, which are used, e.g., by YAGO for creating an ontology [4], are also present in Fandom and extracted into DBkWik, but currently not used for building an ontology. Unlike the category graph extracted from Wikipedia, which is quite strongly connected, the category graph in DBkWik rather consists of many small, unconnected pieces.

Third, that ontology contains duplicates as well, since the class and property names are generated uniquely for each Wiki. For example, when considering the above distributions, there are around 1000 Wikis about books and movies, respectively, so that we can expect several hundred definitions for classes such as *book*, *author*, *movie*, *actor*, etc. and the same holds for relations. Hence, the number of classes and relations above may be somewhat misleading. So far, we have not unified the schema to avoid false matches (e.g., a *track* infobox in a Wiki about music may be different from a *track* infobox in a Wiki about railways), since sophisticated schema or ontology matching [54,55] would be required to fuse the schema.

The DBkWik dataset is also matched to DBpedia, currently using simple string matching methods. A first inspection showed that the mapping works quite well on the schema level, but lacks precision on the instance level [7].

5.2. Setup

In our experiments, we use only the machine learning approach that worked best in the DBpedia experiments, i.e., RandomForests. Unlike the experiments done above, we used *Weka* [56] for the machine learning part in order to be able to directly integrate the learning in the DBpedia Extraction Framework.

Since the features used in our approach require domain and range definitions in the ontology, but the DBkWik ontology is very shallow and informal, as discussed above, we use a simple heuristic inference step to that end, assuming that the most frequently appearing type in the subject and object position are the property's domain and range, similar to the approaches introduced in [57,58]. Furthermore, we ran a simplistic version of the heuristic typing algorithm SDType [59], which creates types for untyped entities in a knowledge graph. This resulted in the typing of 90,791 additional entities (i.e., an increase of 4.6 percent).

5.3. Results

In total, we learned models for 2058 relations, which crossed the lower threshold of 95%. With the help of those models, 311,840 additional relation assertions could added to DBkWik, which were previously not contained in the graph.

Tables 10 and 11 show the 10 Wikis for which the most additional statements could be extracted, and the 10 relations for which we could add the most statements. All of the top 10 Wikis are in English language. It is remarkable that the Wiki for which the approach yields the most additional statements is one from the minor categories according to Figure 9, i.e., education.

Table 10. The 10 Wikis from which most statements could be extracted.

Wiki	Hub	Topic	Models	Statements
military	Lifestyle	Education	21	44,916
wizard101	Games	Gaming	1	28,253
fallout	Games	Gaming	16	11,338
yugi-oh	Games	Anime	24	10,157
yugioh	Games	Anime	24	10,007
prowrestling	Lifestyle	Sports	8	7415
farmville	Games	Gaming	8	5149
rune-scape	Games	Gaming	8	4182
villains	TV	Entertainment	1	4146
runescape	Games	Gaming	8	4143

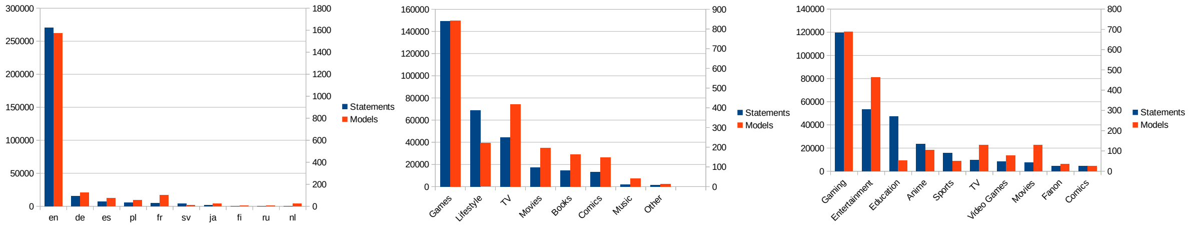
Table 11. The relations with the largest amount of extracted statements.

Property	Statements
vendor	28,273
battlesLabel	14,795
aircraftHelicopterAttack	10,350
type	8015
weakness	7726
sender	6995
rasse	5752
series	5745
factionHostile	5462
occupation	5149

Figure 10 shows the distribution of the extracted relations by language of the Wiki, by topic, and by hub. It can be observed that there are some deviations from the overall distribution of Wikis. For example, a number of relations are extracted from Japanese and Finnish Wikis, although those are not in the top 10 of languages according to the distribution of Wikis. In general, one would expect that the productivity of our approach for different language would more or less resemble the overall distribution of languages across Wikis. Whether these deviations stem from the nature of the Wikis or the nature of the languages at hand is an open research question.

The distribution by hub resembles the original distribution of Wikis per hub. The same holds for the distribution of topics, however, for the latter, the number of relations extracted from sports Wikis is higher than the overall fraction of Wikis of that topic.

In general, comparing those results to those achieved on DBpedia, it can be observed that the models learned for DBkWik are much less productive: on DBpedia, one model learned for English produces around 1000 statements, for DBkWik, it is less than 200. The reason is that for DBpedia, each model is applied on the entire Wikipedia, while for DBkWik, models are so far only applied to each Wiki in isolation.



(a) Top 10 languages **(b)** Hubs **(c)** Top 10 topics

Figure 10. Distribution of learned models and extracted relations by language, hub, and topic.

6. Conclusions and Outlook

Adding new relations to existing knowledge graphs is an important task in adding value to those knowledge graphs. In this paper, we have introduced an approach that adds relations to knowledge graphs which are linked Wikis by performing relation extraction on the corresponding Wiki page's abstracts. Unlike other works in that area, the approach presented in this paper uses background knowledge from DBpedia, but does not rely on any language-specific techniques, such as POS tagging, stemming, or dependency parsing. Thus, it can be applied to Wikis in any language. Furthermore, we have shown that the approach cannot only be applied to Wikipedia, but also presented experiments on a larger and more diverse Wikifarm, i.e., *Fandom powered by Wikia*.

We have conducted two series of experiments, one with DBpedia and Wikipedia, the other with DBkWik. The experimental results for DBpedia show that the approach can add up to one million additional statements to the knowledge graph. By extending the set of abstracts from English to the most common languages, the coverage both of relations for which high quality models can be learned, as well as of instantiation of those relations, significantly increases. Furthermore, we can observe that for certain relations, the models learned are rather universal across languages.

In a second set of experiments, we have shown that the approach is not just applicable to Wikipedia, but also to other Wikis. Given that there are hundreds of thousands of Wikis on the Internet, and—as shown for DBkWik—they can be utilized for knowledge graph extraction, this shows that the approach is valuable not only for extracting information from Wikipedia. Here, we are able to extend DBkWik by more than 300,000 relation assertions.

There are quite a few interesting directions for future work. Following the observation in [29] that multi-lingual training can improve the performance for each single language, it might be interesting to apply models also on languages on which they had not been learned. Assuming that certain patterns exist in many languages (e.g., the first place being mentioned in an article about a person being the person's birth place), this may increase the amount of data extracted.

The finding that for some relations, the models are rather universal across different languages, gives way to interesting refinements of the approach. Since the learning algorithms sometimes fail to learn a high quality model for one language, it would be interesting to reuse a model which performed good on other languages in those cases, i.e., to decouple the learning and the application of the models. With such a decoupled approach, even more instantiations of relations can be potentially extracted, although the effect on precision would need to be examined with great care. Furthermore, designing ensemble solutions that collect evidence from different languages could be designed, which would allow for lowering the strict precision threshold of 95% applied in this paper (i.e., several lower precision models agreeing on the same extracted relations could be a similarly or even more valuable evidence as one high-precision model).

Likewise, for the DBkWik experiments, as discussed above, entities and and relations are not matched. This means that each model is trained and applied individually for each Wiki. As for the future, we plan to match the Wikis upon each other, we expect much more statements as a result after that preprocessing step, since (a) each model can be *trained* with more training data (i.e., it therefore has a higher likelihood to cross the 0.95 precision threshold), and (b) each model is *applied* on more Wikis, hence will be more productive.

In our experiments, we have only concentrated on relations between entities so far. However, a significant fraction of statements in DBpedia, DBkWik, and other knowledge graphs also have literals as objects. That said, it should be possible to extend the framework to such statements as well. Although numbers, years, and dates are usually not linked to other entities, they are quite easy to detect and parse using, e.g., regular expressions or specific taggers and parsers such as *HeidelTime* [60] or the CRF-based parser introduced in [61]. With such a detection step in place, it would also be possible to learn rules for datatype properties, such as: *the first date in an abstract about a person is that person's birthdate*, etc.

Furthermore, our focus so far has been on adding missing relations. A different, yet related problem is the detection of wrong relations [6,62,63]. Here, we could use our approach to gather *evidence* for relations in different language editions of Wikipedia. Relations for which there is little evidence could then be discarded (similar to DeFacto [64]). While for adding knowledge, we have tuned our models towards *precision*, such an approach, however, would rather require a tuning towards *recall*. In addition, since there are also quite a few errors in numerical literals in DBpedia [65,66], an extension such as the one described above could also help detecting such issues.

When it comes to the extension to multiple Wikis, there are also possible applications to knowledge fusion, which has already been examined for relations extracted from Wikipedia infoboxes in different languages [67]. Here, when utilizing thousands of Wikis, it is possible to gain multiple sources of support for an extracted statement, and hence develop more sophisticated fusion methodologies.

So far, we have worked on one genre of text, i.e., abstracts of encyclopedic articles. However, we are confident that this approach can also be applied to other genres of articles as well, as long as those follow typical structures. Examples include, but are not limited to: extracting relations from movie, music, and book reviews, from short biographies, or from product descriptions. All those are texts that are not strictly structured, but expose certain patterns. While for the Wikipedia abstracts covered in this paper, links to the DBpedia knowledge graph are implicitly given, other text corpora would require entity linking using tools such as DBpedia Spotlight [68].

In summary, we have shown that abstracts in Wikis are a valuable source of knowledge for extending knowledge graphs such as DBpedia and DBkWik. Those abstracts expose patterns which can be captured by language-independent features, thus allowing for the design of language-agnostic systems for relation extraction from such abstracts.

Author Contributions: Nicolas Heist and Heiko Paulheim conceived the approach. Nicolas Heist conducted the experiments on DBpedia, Heiko Paulheim conducted the language-specific analysis, and Sven Hertling conducted the experiments on DBkWik.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P.N.; Hellmann, S.; Morsey, M.; van Kleef, P.; Auer, S.; et al. DBpedia—A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semant. Web J.* **2013**, *6*, 167–195.
2. Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; Taylor, J. Freebase: A Collaboratively Created Graph Database For Structuring Human Knowledge. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, Vancouver, BC, Canada, 9–12 June 2008; ACM: New York, NY, USA, 2008; pp. 1247–1250.
3. Vrandečić, D.; Krötzsch, M. Wikidata: A Free Collaborative Knowledge Base. *Commun. ACM* **2014**, *57*, 78–85.
4. Mahdisoltani, F.; Biega, J.; Suchanek, F.M. YAGO3: A Knowledge Base from Multilingual Wikipedias. In Proceedings of the Conference on Innovative Data Systems Research, Asilomar, CA, USA, 4–7 January 2015.

5. Ringler, D.; Paulheim, H. One knowledge graph to rule them all? Analyzing the differences between DBpedia, YAGO, Wikidata & Co. In Proceedings of the German Conference on Artificial Intelligence (Künstliche Intelligenz), Dortmund, Germany, 25–29 September 2017; Springer: New York, NY, USA, 2017; pp. 366–372.
6. Paulheim, H. Knowledge Graph Refinement: A Survey of Approaches and Evaluation Methods. *Semant. Web* **2016**, *8*, 489–508
7. Hofmann, A.; Perchani, S.; Portisch, J.; Hertling, S.; Paulheim, H. DBkWik: Towards knowledge graph creation from thousands of wikis. In Proceedings of the International Semantic Web Conference (Posters and Demos), Vienna, Austria, 21–25 October 2017.
8. Fundel, K.; Küffner, R.; Zimmer, R. RelEx—Relation extraction using dependency parse trees. *Bioinformatics* **2007**, *23*, 365–371.
9. Schutz, A.; Buitelaar, P. Relext: A tool for relation extraction from text in ontology extension. In Proceedings of the Semantic Web—ISWC 2005, Galway, Ireland, 6–10 November 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 593–606.
10. Zelenko, D.; Aone, C.; Richardella, A. Kernel methods for relation extraction. *J. Mach. Learn. Res.* **2003**, *3*, 1083–1106.
11. Bender, E.M. Linguistically naïve! = language independent: Why NLP needs linguistic typology. In Proceedings of the EACL 2009 Workshop on the Interaction between Linguistics and Computational Linguistics: Virtuous, Vicious or Vacuous? Athens, Greece, 30 March 2009; pp. 26–32.
12. Leuf, B.; Cunningham, W. *The Wiki Way: Quick Collaboration on the Web*; Addison-Wesley: Boston, MA, USA, 2001.
13. MediaWiki. Available online: https://www.mediawiki.org/. (accessed on 29 March 2018).
14. Wiki Usage. Available online: https://trends.builtwith.com/cms/wiki (accessed on 29 March 2018).
15. Weaver, G.; Strickland, B.; Crane, G. Quantifying the accuracy of relational statements in wikipedia: A methodology. In Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '06), Chapel Hill, NC, USA, 11–15 June 2006; Volume 6, p. 358.
16. Hachey, B.; Radford, W.; Nothman, J.; Honnibal, M.; Curran, J.R. Evaluating entity linking with Wikipedia. *Artif. Intell.* **2013**, *194*, 130–150.
17. Moro, A.; Navigli, R. SemEval-2015 Task 13: Multilingual All-Words Sense Disambiguation and Entity Linking. In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Denver, CO, USA, 4–5 June 2015; pp. 288–297.
18. Heist, N.; Paulheim, H. Language-agnostic relation extraction from wikipedia abstracts. In Proceedings of the International Semantic Web Conference, Vienna, Austria, 21–25 October 2007; Springer: Cham, Switzerland, 2017, pp. 383–399.
19. Mintz, M.; Bills, S.; Snow, R.; Jurafsky, D. Distant supervision for relation extraction without labeled data. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, Singapore, 2–7 August 2009; pp. 1003–1011.
20. Aprosio, A.P.; Giuliano, C.; Lavelli, A. Extending the Coverage of DBpedia Properties using Distant Supervision over Wikipedia. In Proceedings of the NLP&DBpedia, Sydney, Australia, 22 October 2013; Volume 1064, CEUR Workshop Proceedings.
21. Gerber, D.; Ngomo, A.C.N. Bootstrapping the Linked Data web. In Proceedings of the Workshop on Web Scale Knowledge Extraction, Bonn, Germany, 23–27 October 2011.
22. Nguyen, D.P.; Matsuo, Y.; Ishizuka, M. Relation extraction from wikipedia using subtree mining. In Proceedings of the National Conference on Artificial Intelligence, Vancouver, BC, Canada, 22–26 July 2007; Volume 22, p. 1414.
23. Yan, Y.; Okazaki, N.; Matsuo, Y.; Yang, Z.; Ishizuka, M. Unsupervised Relation Extraction by Mining Wikipedia Texts Using Information from the Web. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, Singapore, 2–7 August 2009; pp. 1021–1029.
24. Lange, D.; Böhm, C.; Naumann, F. Extracting structured information from Wikipedia articles to populate infoboxes. In Proceedings of the 19th ACM Conference on Information and Knowledge Management (CIKM), Toronto, ON, Canada, 26–30 October 2010; ACM: New York, NY, USA, 2010; pp. 1661–1664.

25. Wu, F.; Hoffmann, R.; Weld, D.S. Information extraction from Wikipedia: Moving down the long tail. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008; ACM: New York, NY, USA, 2008; pp. 731–739.
26. Wang, G.; Yu, Y.; Zhu, H. PORE: Positive-only Relation Extraction from Wikipedia Text. In Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference, ISWC'07/ASWC'07, Busan, Korea, 11–15 November 2007; Springer: New York, NY, USA, 2007; pp. 580–594.
27. Faruqui, M.; Kumar, S. Multilingual open relation extraction using cross-lingual projection. *arXiv* **2015**, arXiv:1503.06450.
28. Nguyen, T.H.; Grishman, R. Relation extraction: Perspective from convolutional neural networks. In Proceedings of the NAACL-HLT, Denver, CO, USA, 5 June 2015; pp. 39–48.
29. Verga, P.; Belanger, D.; Strubell, E.; Roth, B.; McCallum, A. Multilingual relation extraction using compositional universal schema. *arXiv* **2015**, arXiv:1511.06396.
30. Zeng, D.; Liu, K.; Chen, Y.; Zhao, J. Distant supervision for relation extraction via piecewise convolutional neural networks. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), Lisbon, Portugal, 17–21 September 2015; pp. 17–21.
31. Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; Zhao, J. Relation Classification via Convolutional Deep Neural Network. In Proceedings of the COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland, 23–29 August 2014; pp. 2335–2344.
32. DBpedia(2014). Available online: http://oldwiki.dbpedia.org/Downloads2014 (accessed on 29 March 2018).
33. Web Ontology Language (OWL). Available online: https://www.w3.org/OWL/ (accessed on 29 March 2018).
34. Galárraga, L.A.; Teflioudi, C.; Hose, K.; Suchanek, F. AMIE: Association rule mining under incomplete evidence in ontological knowledge bases. In Proceedings of the 22nd International Conference on World Wide Web, Rio de Janeiro, Brazil, 13–17 May 2013; pp. 413–422.
35. Dong, X.L.; Murphy, K.; Gabrilovich, E.; Heitz, G.; Horn, W.; Lao, N.; Strohmann, T.; Sun, S.; Zhang, W. Knowledge Vault: A Web-scale approach to probabilistic knowledge fusion. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 601–610.
36. Cohen, W.W. Fast Effective Rule Induction. In Proceedings of the Machine Learning, Twelfth International Conference on Machine Learning, Tahoe City, CA, USA, 9–12 July 1995; pp. 115–123.
37. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32.
38. Kubat, M. Neural networks: A comprehensive foundation by Simon Haykin, Macmillan, 1994, ISBN 0-02-352781-7. *Knowl. Eng. Rev.* **1999**, *13*, 409–412.
39. Cristianini, N.; Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*; Cambridge University Press: Cambridge, UK, 2010.
40. Rapidminer. Available online: http://www.rapidminer.com/ (accessed on 29 March 2018).
41. Code and data. Available online: http://dws.informatik.uni-mannheim.de/en/research/language-agnostic-relation-extraction-from-wikipedia-abstracts (accessed on 29 March 2018).
42. Wikistats. Available online: http://wikistats.wmflabs.org/display.php?t=wp (accessed on 29 March 2018).
43. DBpedia ontology. Available online: http://dbpedia.org/services-resources/ontology (accessed on 29 March 2018).
44. Paulheim, H. Data-driven joint debugging of the DBpedia mappings and ontology. In Proceedings of the European Semantic Web Conference, Portoroz, Slovenia, 28 May–1 June 2017; Springer: Cham, Switzerland, 2017; pp. 404–418.
45. Skinner(2017). Available online: https://tobyskinner.net/2017/06/11/the-worlds-most-prolific-writer/ (accessed on 29 March 2018).
46. Oxford Internet Institute(2013). Available online: http://geography.oii.ox.ac.uk/?page=geographic-intersections-of-languages-in-wikipedia (accessed on 29 March 2018).
47. Ristoski, P.; Bizer, C.; Paulheim, H. Mining the web of linked data with rapidminer. *Web Semant. Sci. Serv. Agents World Wide Web* **2015**, *35*, 142–151.
48. DBpedia Extraction Framework(2018). Available online: https://github.com/dbpedia/extraction-framework (accessed on 29 March 2018).
49. WikiApiary(2018). Available online: https://wikiapiary.com/wiki/Statistics (accessed on 29 March 2018).
50. Fandom(2018). Available online: http://fandom.wikia.com/ (accessed on 29 March 2018).

51. Alexa. Available online: http://www.alexa.com/topsites/category/Computers/Software/Groupware/Wiki/Wiki_Farms (accessed on 29 March 2018).
52. DBkWik(2018a). Available online: http://dbkwik.webdatacommons.org (accessed on 29 March 2018).
53. DBkWik(2018b). Available online: http://data.dws.informatik.uni-mannheim.de/dbkwik/dbkwik-v1.0.tar.gz (accessed on 29 March 2018).
54. Rahm, E.; Bernstein, P.A. A survey of approaches to automatic schema matching. *VLDB J.* **2001**, *10*, 334–350.
55. Shvaiko, P.; Euzenat, J. A survey of schema-based matching approaches. In *Journal on Data Semantics IV*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 146–171.
56. Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I.H. The WEKA data mining software: An update. *ACM SIGKDD Explor. Newslett.* **2009**, *11*, 10–18.
57. Völker, J.; Niepert, M. Statistical schema induction. In Proceedings of the Extended Semantic Web Conference, Heraklion, Greece, 29 May–2 June 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 124–138.
58. Töpper, G.; Knuth, M.; Sack, H. DBpedia ontology enrichment for inconsistency detection. In Proceedings of the 8th International Conference on Semantic Systems, Graz, Austria, 5–7 September 2012; ACM: New York, NY, USA, 2012; pp. 33–40.
59. Paulheim, H.; Bizer, C. Type inference on noisy rdf data. In Proceedings of the International Semantic Web Conference, Sydney, Australia, 21–25 October 2013; Springer: Berlin/Heidelberg, Germany, 2013; pp. 510–525.
60. Strötgen, J.; Gertz, M. HeidelTime: High quality rule-based extraction and normalization of temporal expressions. In Proceedings of the 5th International Workshop on Semantic Evaluation, Los Angeles, CA, USA, 15–16 July 2010; pp. 321–324.
61. Paulheim, H. A robust number parser based on conditional random fields. In Proceedings of the Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz), Dortmund, Germany, 25–29 September2017; Springer: Cham, Switzerland, 2017; pp. 337–343.
62. Paulheim, H.; Bizer, C. Improving the Quality of Linked Data Using Statistical Distributions. *Int. J. Semant. Web Inf. Syst.* **2014**, *10*, 63–86.
63. Paulheim, H.; Gangemi, A. Serving DBpedia with DOLCE—More than Just Adding a Cherry on Top. In Proceedings of the International Semantic Web Conference, Bethlehem, PA, USA, 11–15 October 2015; Springer: Cham, Switzerland, 2015; Volume 9366, LNCS.
64. Gerber, D.; Esteves, D.; Lehmann, J.; Bühmann, L.; Usbeck, R.; Ngomo, A.C.N.; Speck, R. DeFacto—Temporal and multilingual Deep Fact Validation. *Web Semant. Sci. Serv. Agents World Wide Web* **2015**, *35*, 85–101.
65. Fleischhacker, D.; Paulheim, H.; Bryl, V.; Völker, J.; Bizer, C. Detecting Errors in Numerical Linked Data Using Cross-Checked Outlier Detection. In Proceedings of the Semantic Web—ISWC, Riva del Garda, Italy, 19–23 October 2014; Springer: Cham, Switzerland, 2014; Volume 8796, LNCS, pp. 357–372.
66. Wienand, D.; Paulheim, H. Detecting Incorrect Numerical Data in DBpedia. In Proceedings of the Semantic Web: Trends and Challenges, Anissaras, Greece, 25–29 May 2014; Springer: Cham, Switzerland, 2014; Volume 8465, LNCS, pp. 504–518.
67. Bryl, V.; Bizer, C. Learning conflict resolution strategies for cross-language wikipedia data fusion. In Proceedings of the 23rd International Conference on World Wide Web, Seoul, Korea, 7–11 April 2014; ACM: New York, NY, USA, 2014; pp. 1129–1134.
68. Mendes, P.N.; Jakob, M.; García-Silva, A.; Bizer, C. DBpedia spotlight: Shedding light on the web of documents. In Proceedings of the 7th International Conference on Semantic Systems, Graz, Austria, 7–9 September 2011.