

# Process Extraction from Natural Language Text: the PET Dataset and Annotation Guidelines

Patrizio Bellan<sup>1,2</sup>, Chiara Ghidini<sup>1</sup>, Mauro Dragoni<sup>1</sup>, Simone Paolo Ponzetto<sup>3</sup> and Han van der Aa<sup>3</sup>

<sup>1</sup>Fondazione Bruno Kessler, via sommarive, 18, Povo (Tn), Italy

<sup>2</sup>Free University of Bozen-Bolzano, Bolzano (Bz), Italy

<sup>3</sup>University of Mannheim, Mannheim, Germany

## Abstract

Although there is a long tradition of work in NLP on extracting entities and relations from text, to date there exists very little work on the acquisition of business processes from unstructured data such as textual corpora of process descriptions. With this work, we aim to fill this gap and establish the first steps towards bridging data-driven information extraction methodologies from Natural Language Processing and the model-based formalization aimed at Business Process Management. For this, we develop the first corpus of business process descriptions annotated with activities, gateways, actors, and flow information. We present our new resource, including a detailed overview of the annotation schema and guidelines, as well as a variety of baselines to benchmark the difficulty and challenges of business process extraction from text.

## Keywords

Process Extraction from Text, Business Process Management, Information Extraction, Annotation Schema, Annotation Guidelines

## 1. Introduction

Information Extraction (IE), a key area of research focused on extracting structured representations from unstructured text, has a long-standing tradition in Natural Language Processing (NLP), from seminal contribution in the context of the Message Understanding Conference using finite-state techniques [2] all the way through current neural approaches to document-level relation extraction [3]. Despite this large volume of work, historically, most of the focus has concentrated on standard newswire text<sup>1</sup>. Moreover, most successful approaches are rather schema-weak, an approach epitomized by a very successful line of research such as Open Information Extraction [6, 7]. In this work, we propose the first steps towards shifting some of this focus in IE towards a new domain and task. Specifically, we focus on the problem of extracting a Business Process Model from textual content – which can, in turn, be viewed as the problem of extracting activities and workflow elements from process descriptions that

---

NL4AI 2022: Sixth Workshop on Natural Language for Artificial Intelligence, November 30, 2022, Udine, Italy [1]

✉ pbellan@fbk.eu (P. Bellan)

🆔 0000-0002-2971-1872 (P. Bellan); 0000-0003-1563-4965 (C. Ghidini); 0000-0003-0380-6571 (M. Dragoni); 0000-0001-7484-2049 (S. P. Ponzetto); 0000-0002-4200-4937 (H. v. d. Aa)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

<sup>1</sup>Notable exceptions are efforts devoted to IE for Ontology Construction [4, 5]

can be represented by adopting Business Process Management and Notation or compiled into Petri Nets [8]. But while there has been in recent years growing interest from the Business Process Management (BPM) community in the extraction of processes from text [9, 10, 11], current work has major limitations arguably due to the limited availability of domain-specific, human-annotated gold-standard data that could be used to train from scratch or fine-tune data-driven methods, and which are essential to enable task-specific comparison across competing approaches [12, 13]. Creating benchmarks from text, however, is at the heart of much work the NLP community – cf. the long-standing tradition of SENSEVAL and SemEval evaluation campaigns in computational semantics: despite the major limitations shown by current ‘leader-boardism’ [14], the availability of reference gold-standard dataset has the potential of fostering the application of NLP techniques to other fields, such as for instance BPM, and crucially make clear what the applicability and limitations of state-of-the-art approaches for the domain of interest are.

In [15] we have presented a new annotated dataset, called PET, of human-annotated processes in a corpus of process descriptions. In this paper we present this resource more in detail, by providing insights to the problem of Process Annotation from Text (Section 2), to the annotation guidelines (Section 3), the PET dataset itself (Section 4), and baseline results for information extraction tasks for process model extraction (Section 5).

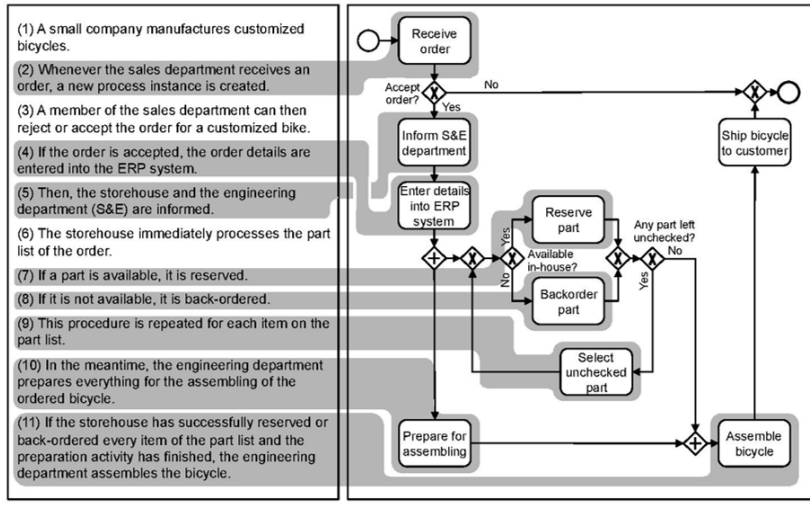
Our vision builds upon bringing together heterogeneous communities such as NLP and BPM practitioners by defining shared tasks and resources (cf. previous work from [16] at the intersection of NLP and political science). All resources described in this paper are freely available for the research community at [huggingface.co/datasets/patriziobellan/PET](https://huggingface.co/datasets/patriziobellan/PET).

## 2. Problem Background

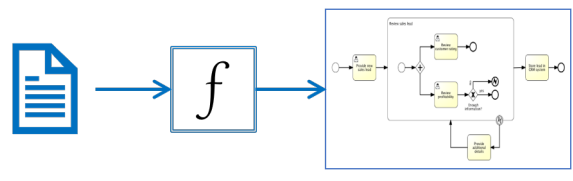
The extraction of a process model from documents is a complex task since the analysis of the natural language description of a process has to take care of the multiple linguistics levels (syntactical, semantics, and pragmatics) and mitigate linguistics phenomena such as syntactic leeway, simultaneously. Moreover, it has to handle the multiple possible interpretations (in forms of process model) that can be inferred from the same text, because the same semantic can be conveyed in multiple ways, and maybe not always equivalent. For example, there are different ways to represent, in a business process diagram, a repeated event or activity, but maybe the case that only one of these possible interpretations is the correct one to represent in the formal model.

Figure 1 presents an example of the process extraction task. The gray shadow boxes link each process description sentence on the left of the figure that describes a process element to its corresponding process element in the process diagram, represented in BPMN, on the right part of the figure. Here, the ninth sentence can be represented in different ways than the one reported in the diagram. It is possible to represent the same semantic, for example, adopting a sub-process element (in the case we need to re-use this part of the diagram somewhere else), or as a multi-instance activity (either parallel or sequential).

In general, the first task to perform when analyzing a process description regards filtering *uninformative sentences* of the process description out, because not all the sentences represent



**Figure 1:** The figure, taken from [17], shows an example of text-to-model mapping in which only the meaningful activities described in the process description (on the left) are mapped in the process model diagram (on the right). Two interesting aspects are worthy to note. First, not all the sentences correspond to a process model element. Second, the logical succession described in the text differs from the written sentences' order (as happens with sentences 4 and 5).



**Figure 2:** The figure shows an abstraction of the algorithmic function  $f$  that maps a natural language process description (represented with the blue document icon) to its formal representation, the process model diagram (displayed on the right part of the figure).

process elements. Then, Actions, Actors, Events, Gateways, Artifacts, and various types of process flows can be extracted. However, not only each sentence can describe multiple process elements, but also each word can have multiple meanings. Determining the correct intended meaning and mapping it into the corresponding process element implies considering these two aspects at once. Finally, the process elements discovered have to be logically organized following the semantics conveyed in the process description. So, defining the *logical succession of process model elements* is another challenge to tackle.

Figure 2 shows an abstract level of the process extraction from natural language text task that is conceptualized as an algorithmic function  $f$  that aims to “map” a natural language process description into its process model. In the figure, the process description is represented with the blue document icon on the left, and the process model generated from the function  $f$ , is represented on the right as a BPMN diagram.

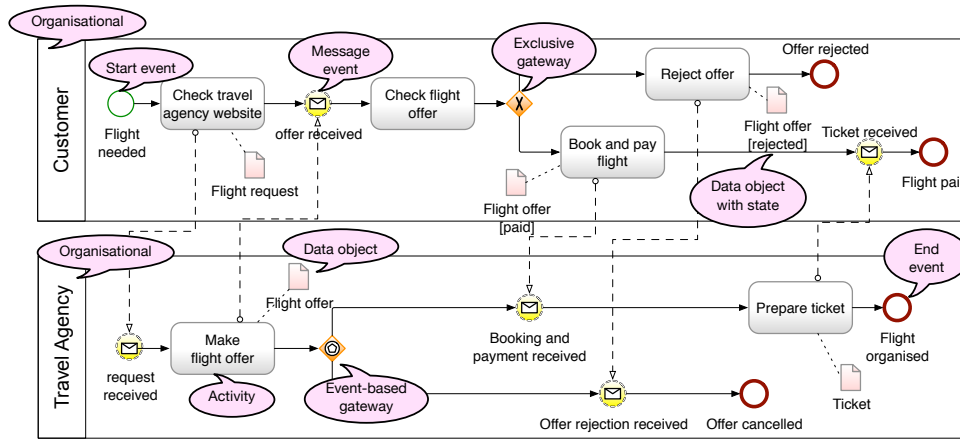


Figure 3: A Business Process Diagram in the BPMN language.

### 3. Annotation Guidelines

In this Section, we introduce <sup>2</sup> the annotation guidelines we defined to create the dataset. The annotation of a document describing a process is a difficult task. Being able to identify **process elements** requires having at least a rough understanding of the typical elements contained in process modeling languages. In terms of languages, we target a procedural language such as BPMN, although the guidelines may also apply to other standard procedural modeling languages such as UML Activity Diagram. To provide an overview of the graphical language and of the type of elements it typically contains, please refer to the diagram in Figures 3, taken from [18], which provides a model of a customer buying a flight ticket from a travel agency. Besides illustrating the scenario, the diagram is “annotated” with speech balloons indicating the type of entity denoted by the graphical constructs. Following the classification made in [18], we can group these constructs into three macro categories:

1. **Behavioral.** These elements are the ones that refer to the so-called *control flow* of the process, that is the flow determined by the set of activities that are performed in coordination. This category is the most articulated in a business process and contains at least 3 types of objects:
  - *activities* and *events*, that is the things that happen in time<sup>3</sup>. In our example the activity check the flight offer or the event payment received.
  - *flow objects*, that is constructs that enable the routing of the flow between the activities such as the sequence relation between activities, or the gateways that enable the routing of the flow. In our example the (precedence) relation between

<sup>2</sup>The reader may find the complete annotation guidelines document at [pdi.fbk.eu/pet/annotation-guidelines-for-process-description.pdf](http://pdi.fbk.eu/pet/annotation-guidelines-for-process-description.pdf)

<sup>3</sup>While these elements often have a different meaning in some modeling languages we do not distinguish between them here

make `flight offer` and check `flight offer` or the (mutually) exclusive gateway between `reject offer` and `book and pay flight`, and finally

- *states*, that is conditions of the world that affect the flow in the process such as the pre-post conditions for the occurrence of an activity or a guard on a gateway. In our example, the (un)satisfied status of the customer w.r.t. a `flight offer`.
2. **Data object.** These elements usually describe, at a high level of abstraction, the objects upon which an activity acts. Examples in the scenario above are the `flight request` and the `flight ticket`. Note that sometimes these data objects complement the activity itself (as in the case of the data object `flight request`, which is produced by the activity `check travel agency website` while in other cases they are implicitly described in the activity itself, as in the case of `flight ticket` with the activity `prepare ticket`. In this latter case, the data object is often left implicit.
  3. **Organizational.** These elements are usually related to the *who* question, and often describe, at a high level of abstraction, the roles / organizational structures involved in the activities of the process.

It is important to highlight that *Data* and *Organizational* objects do not exist per-se in a business process diagram but they usually refer to the activities. More formally, they are participants of the activity as they participate in the activity itself.

We aim at proposing a general annotation schema able to deal with unknown scenarios. Note that, while inspired by [18], the conceptual layers described in that paper slightly differ from the Annotation schema proposed in this document. This was done to increase the flexibility of the annotation schema to capture the different ways in which a process element can be described.

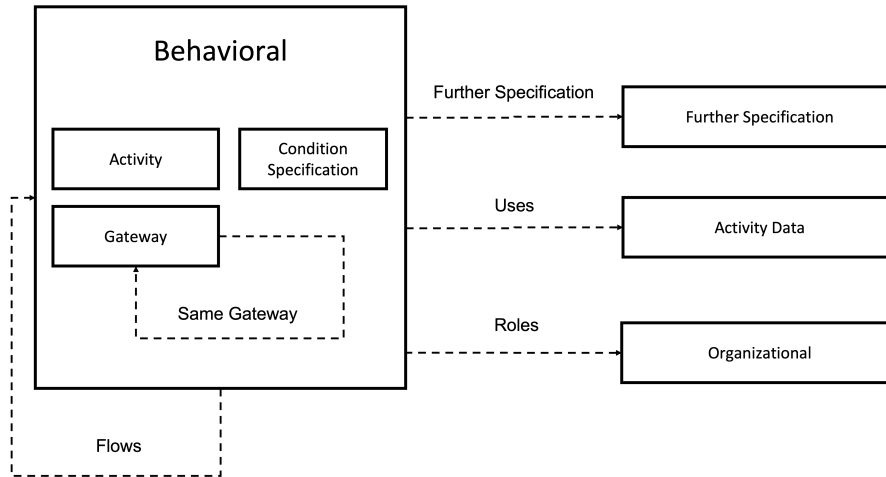
As a crucial example, we decided to break down activity to differentiate among the activity elements it is composed of. In particular, we capture the activity “action” expression and the object the activity acts on in two different annotation layers. This choice allows to easier the annotation workload and it also reduces the possibility of making errors (for example, connecting with a *Sequence Flow relation* the activity data to the actor responsible for the execution of the activity). For instance, we differentiate the expression describing an *Activity* to the object the activity uses. The overall goal is to annotate process model elements and their relations in documents.

We implemented the annotation schema described in this document in the Inception Annotation tool ([inception-project.github.io](http://inception-project.github.io)). The schema can be downloaded from [pdi.fbk.eu/pet/inception-schema.json](http://pdi.fbk.eu/pet/inception-schema.json).

### 3.1. Layers Overview

Here, we explore the process elements we considered in the proposed dataset and their relations. Figure 4 provides an overview of the different layers and shows their relations.

**Behavioral Layer** The *Behavioral layer* captures information about the behavioral elements described and their relations. Figure 4 shows the relations between the *Behavioral layer* and the other ones. The *Behavioral layer* is the core layer since it captures activities, gateways, branch



**Figure 4:** Annotation schema.

conditions, and flow relations. An *activity* element represents a single task performed within a process model. A *gateway* element represents a decision point, and the *condition specification* represents the condition that a process execution instance must satisfy to be allowed to enter a specific branch of a gateway. A *Flow* is a relation that defines the process model logic by connecting all the elements that belong to this layer together.

The Behavioral layer is composed of six features: *Element Type*, *Uses*, *Flow*, *Roles*, *Further Specification*, *Same Gateway*. Since this layer captures both *Activity* and *Gateways*, not all the features are always required, but they depend on the *Element Type* and the situation described in the text. For example, if a text does not describe any *Actor Performer* the feature *Roles* is left empty.

The feature **Element Type** defines the type of the process model elements marked as *Activity*, or *AND Gateway*, or *XOR Gateway*, *Condition Specification*. This layer is connected to the layer *Activity Data* by the *Uses* relation. This feature links activity to the *Activity Data* annotated in the layer *Activity Data*. Hence, this relation allows connecting an activity expression (either verbal or nominal) with the object the activity acts on. Process participants (actors involved in an activity) that are captured in the *Organizational* layer, are bound to activity through the feature **Roles**. Here we differentiate between *Actor Performer* relation that links the actor who is responsible for an activity execution to the activity, to the *Actor Recipient* relation that links the actor who receives the results of the execution of an activity. The **Further Specification** feature allows connecting activity to its important details (captured in the *Further Specification* layer). The *Further Specification* layer captures the important information about an activity that is not captured by the other layers, such as the mean, the manner of execution, or how an activity is executed. The **Same Gateway** feature allows connecting all the parts describing the same gateway together, since its description may span over multiple sentences. This means that only gateway elements can be connected by this relation. The *Behavioral* layer makes a connection to itself through the relation **Flow**. This feature allows for defining the process model logic by connecting behavioral elements in sequential order.

**Activity Data Layer** The *Activity Data* layer captures the object of an activity expression acts on.

**Further Specification** The *Further Specification* layer captures important details of an Activity, such as the mean or the manner of its execution.

**Organizational Layer** The *Organizational* layer is meant to annotate at a high level of abstraction the process participants that are responsible for activities. They typically represent the *Actors* involved in a process.

### 3.2. Examples

We conclude this section by showing some annotations in a text. We start with the annotation of activities.

The sentence

“The office sends the forms to the customer by email”

is annotated as follows: the activity sends Uses the activity data the forms; the actor The office is the *Actor Performer* of sends while the actor the customer is its *Actor Recipient*; the further specification details by email is link to the sends via *Further Specification* relation.

A different situation concerns the annotation of gateways. Besides the well-defined theoretical definitions of Gateways, the annotation of gateways is challenging for two main reasons. First, the description of a gateway typically spans on multiple text pieces and/or on multiple sentences. To deal with this challenge, we use the *Same Gateway* relation to reconstruct the gateway. Consider the following example:

“If an error is detected another arbitrary repair activity is executed, otherwise the repair is finished.”

Here, the gateway description span over two sentence pieces: If and otherwise. We reconstruct the gateway object by connecting If to otherwise using the *Same Gateway* relation. The second reason concerns the lack of an explicit description of merging points in texts. To deal with this challenge, we decided to capture the annotation of a merging point using *Flows* relation. We connect the ending point of each branch of a gateway to the next common behavioral element. As in:

“The ongoing repair consists of two activities. The first activity is to check the hardware, whereas the second activity checks the software. Then, the CRS test the system functionality.”

Here, the word whereas describes an AND Gateway with two branches: (i) check the hardware and (ii) check the software. The next common element (where the process flow goes through) is test the system functionality. To create a merging point at the end of the gateway, we connect check the hardware to test the system functionality, and check the software to test the system functionality.

**Table 1**  
Documents Statistics

Statistic	Value
Total Documents	45
Total Sentences	417
Average sentences per document	9.27
Average words per document	168.2
Average words per sentence	18.15

## 4. The Dataset

The guidelines described in Section 3 were used for the creation of the PET dataset described in this Section and available at [huggingface.co/datasets/patriziobellan/PET](https://huggingface.co/datasets/patriziobellan/PET). The creation of the first version of the PET dataset started from the Friedrich dataset: a set of 47 textual documents preliminary exploited within the BPMN community to start the investigation of the process extraction from natural language text research topic. The reader may find the introduction to the raw version of the textual documents used for building the PET in [19].

The reasons for which we started from this set of documents are two-fold. First, these documents are well-known within the community. This aspect allows to give continuity to the investigation in this research area as well as to start from a base set of documents that are in line with the type of process narratives considered relevant by the community. Second, the documents contained in the dataset are not explicitly annotated with the elements described in Section 3. Indeed, the dataset described in [19] contains only the raw text and a possible corresponding BPMN diagram. However, many of these diagrams were translated by the same authors from other process modeling languages into BPMN without any validation performed by experts. Therefore, the diagrams should not be taken as gold-standard reference. As a consequence, they can not be used to mark process elements in the process descriptions. Hence, the whole work of text processing and elements annotation has to be provided.

The dataset construction process has been split into five main phases:

1. **Text pre-processing.** As the first operation, we check the content of each document and tokenized it. The initial check activity was necessary since some of the original texts were automatically translated into English by the authors of the dataset. The translations were never validated, indeed, several errors have been found and fixed.
2. **Text Annotation.** Each text has been annotated by using the guidelines introduced in 3. The team was composed of five annotators with high expertise in BPMN. Each document has been assigned to three experts that were in charge of identifying all the elements and flows with each document. Within this phase, each annotator has been supported by the Inception tool integrated with the annotation schema available on the dataset web page.
3. **Automatic annotation fixing.** At the end of the second phase, we run an automatic procedure relying on a rules-based script to automatically fix annotations that were not compliant with the guidelines. For example, if a modal verb was erroneously included in



the annotation of an *Activity*, the procedure removed it from the annotation. Another example is the missing article within an annotation related to an *Actor*. In this case, the script included it in the annotation. This phase allowed us to remove possible annotation errors and to obtain annotations compliant with the guidelines.

4. **Agreement Computation.** Here, we computed, on the annotation provided by the experts, the agreement scores for each process element and each relation between process elements pair adopting the methodology proposed in [20].<sup>4</sup> By following such a methodology, an annotation was considered in agreement among the experts if and only if they capture the same span of words and they assign the same process element tag to the annotation. In the same way, a relation was considered in agreement if and only if the experts strictly annotated the same span of words representing (i) the process element related to the source element; (ii) the process element related to the target element; and, (iii) the relation tag between source and target. The only exception regards the *same gateway relation* in which source and target are interchangeable since in this type of relation the relation arrow does not matter. The final agreement scores were obtained by averaging the individual scores obtained by the comparison of annotators pairs. Tables 2 and 3 show the annotation agreement computed for each process element and each process relation, respectively. We can observe how, in general, experts agreed concerning the main elements and flows contained within a process description. On the contrary, the annotation of information classified as *Further Specification* led to several disagreement situations. Such situations were analyzed and mitigated within the next phase.
5. **Reconciliation.** The last phase consisted of the mitigation of the disagreements within the annotations provided by the experts. This phase aims to obtain a shared and agreed set of gold annotations on each text for both entities and relations. Such entities enable, as well, the generation of the related full-connected process model flow that can be rendered by using, but not limited to, a BPMN diagram. During this last phase, among the 47 documents originally included in the dataset, 2 of them were discarded. Such texts were not fully annotated by the annotators since they were not able to completely understand which process elements were included in some specific parts of the text. For this reason, the final size of the dataset is 45 textual descriptions of the corresponding process models together with their annotations. We report in Table 1 the statistics related to the current version of the document, in Tables 4 the detailed statistic about process elements, and in Table 5 the detailed statistic about process elements relations.

We loaded the dataset to the Huggingface repository. We created two task cards for our dataset: (i) *token classification* that aims to predict process elements described in texts, and (ii) *relation extraction* that aims at classifying the relation between two process elements.

---

<sup>4</sup>We measured the agreement in terms of the F1 measure because, besides being straightforward to calculate, it is directly interpretable. Note that chance-corrected measures like  $\kappa$  approach the F1-measure as the number of cases that raters agree are negative grows [20].

**Table 2**  
Annotation Agreement on Process Elements

	Annotators		
	Precision	Recall	F1
Activity	0.960	0.869	0.912
Activity Data	0.934	0.734	0.822
Actor	0.958	0.837	0.893
Further Specification	0.430	0.329	0.373
XOR Gateway	0.881	0.860	0.870
AND Gateway	0.889	0.727	0.800
Condition Specification	0.856	0.761	0.806
Overall	0.915	0.787	0.846

**Table 3**  
Annotation Agreement on Relations

	Annotators		
	Precision	Recall	F1
Sequence Flow	1.000	0.671	0.803
Uses	1.000	0.715	0.834
Actor Performer	0.996	0.743	0.851
Actor Recipient	0.997	0.772	0.871
Further Specification	0.644	0.320	0.427
Same Gateway	0.876	0.720	0.791
Overall	0.982	0.687	0.809

**Table 4**  
Entities Statistics

	Activity	Activity Data	Actor	Further Specification	XOR Gateway	AND Gateway	Condition Specification
Absolute count	501	451	439	64	117	8	80
Relative count	30.16%	27.21%	26.43%	3.86%	7.04%	0.48%	4.82%
Per document	11.13	10.04	9.76	1.42	2.6	0.18	1.78
Per sentence	1.2	1.08	1.05	0.15	0.28	0.02	0.19
Average length	1.1	3.49	2.32	5.19	1.26	2.12	6.04
Standard deviation	0.48	2.47	1.11	3.4	0.77	1.54	3.04

## 5. Baselines Results

We present in this Section three baselines we developed to provide preliminary results obtained on the dataset and also to show how the dataset can be used for testing different extraction approaches. Indeed, as described in Section 4, there are different types of elements that can be extracted (e.g., activities, actors, relations) and different assumptions that can be made (e.g., the exploitation of gold information or the process of the raw text).

From this perspective, we tested our baselines under three different settings and by using two different families of approaches: Conditional Random Fields (CRF) and Rule-Based (RB):

- Baseline 1 (B1): by starting from the raw text (i.e., no information related to process elements or relations has been used), a CRF-based approach has been used for building a

model to support the extraction of single entities (e.g., activities, actors).

- Baseline 2 (B2): by starting from the existing gold information concerning the annotation of process elements, an RB strategy has been used for detecting relations between entities.
- Baseline 3 (B3): this baseline relies on the output of B1 concerning the annotations of process elements. Then, the RB strategy has been used for detecting relations between entities.

Concerning the CRF approach, we adopted the CRF model described in [21] by encoding data following the IOB2 schema.

Results have been obtained by performing 5-folds cross-validation and by averaging observed performance.

While, concerning the RB approach, we defined a set of rules taking into account the text position of process elements. The rules defined are the following:

1. Rule 1 (R1): (*sequence flows*) are annotated by connecting two consecutive behavioral process elements.
2. Rule 2 (R2): (*same gateway*) relations are annotated by connecting two gateway of the same type if they are detected in the same sentence or if they are detected in two consecutive sentences.
3. Rule 3 (R3): (*sequence flows*) relations are annotated between each gateway that is not part of any *same gateway* relation and the next activity detected.
4. Rule 4 (R4): for each activity defined in a sentence, (*actor performer/recipient*) relations are annotated by linking the left-side closest actor as actor performer and the right-side closest actor as actor recipient.
5. Rule 5 (R5): (*further specification*) annotations are defined by connecting each further specification element to the closest activity in the text.
6. Rule 6 (R6): (*uses*) annotations are defined by connecting activity data elements to the closest left-side activity of the same sentence. If no activities are defined on the left side, the right side is considered.

**Table 5**  
Relations Statistics

	Flows	Uses	Actor Performer	Actor Recipient	Further Specification	Same Gateway
Absolute count	674	468	312	164	64	42
Relative count	39.10%	27.15%	18.10%	9.51%	3.71%	2.44%
Count per document	15.31	10.6	6.96	3.64	1.42	0.96
Count per sentence	1.65	1.14	0.75	0.39	0.15	0.1

**Table 6**

Results obtained by Baseline 1 concerning the extraction of Process Elements.

	Baseline 1		
	Precision	Recall	F1
Activity	0.913	0.733	0.813
Activity Data	0.870	0.580	0.696
Actor	0.896	0.665	0.763
Further Specification	0.381	0.125	0.188
XOR Gateway	0.872	0.701	0.777
AND Gateway	0.000	0.000	0.000
Condition Specification	0.800	0.500	0.615
Overall	0.880	0.633	0.736

**Table 7**

Results obtained by Baseline 2 and Baseline 3 concerning the extraction of Process Relations.

	Baseline 2			Baseline 3		
	Precision	Recall	F1	Precision	Recall	F1
Sequence Flow	1.000	0.787	0.881	1.000	0.370	0.540
Uses	1.000	0.891	0.942	1.000	0.488	0.656
Actor Performer	0.992	0.808	0.891	0.994	0.534	0.694
Actor Recipient	0.993	0.817	0.896	1.000	0.476	0.645
Further Specification	1.000	0.828	0.906	0.875	0.109	0.194
Same Gateway	0.973	0.837	0.900	0.897	0.605	0.722
Overall	0.997	0.825	0.903	0.994	0.438	0.608

Table 6 and 7 provide the results obtained by the three baseline approaches described above.

An observation of baselines' performance highlights the general capabilities of the adopted approaches in detecting both process elements and relations with high precision. Exceptions are the *further specification* and *AND Gateway* elements where the baseline obtained very poor performance. While on the one hand, the observed precision is high, on the other hand, the recall is the metric for which lower performance was obtained. In turn, this affected the value of the F1 as well. Hence, an interesting challenge worth of being investigated in this domain seems to be the detection of all elements rather than detecting them correctly.

## 6. Conclusion

In this paper, we presented the PET dataset. The dataset contains 45 documents containing narrative descriptions of business processes and their annotations. Together with the dataset, we provided the set of guidelines we defined and adopted for annotating all documents. The dataset-building procedure has been described and, for completeness, we provided three baselines implementing straightforward approaches to give a starting point for designing the next generation of process extraction from natural language text approaches.

## References

- [1] D. Nozza, L. Passaro, M. Polignano, Preface to the Sixth Workshop on Natural Language for Artificial Intelligence (NL4AI), in: D. Nozza, L. C. Passaro, M. Polignano (Eds.), Proceedings of the Sixth Workshop on Natural Language for Artificial Intelligence (NL4AI 2022) co-located with 21th International Conference of the Italian Association for Artificial Intelligence (AI\*IA 2022), November 30, 2022, CEUR-WS.org, 2022.
- [2] J. R. Hobbs, M. E. Stickel, D. E. Appelt, P. Martin, Interpretation as abduction, *Artificial Intelligence* 63 (1993) 69–142.
- [3] Y. Yao, D. Ye, P. Li, X. Han, Y. Lin, Z. Liu, Z. Liu, L. Huang, J. Zhou, M. Sun, DocRED: A large-scale document-level relation extraction dataset, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 764–777.
- [4] P. Cimiano, *Ontology learning and population from text - algorithms, evaluation and applications*, Springer, 2006. URL: <https://doi.org/10.1007/978-0-387-39252-3>. doi:10.1007/978-0-387-39252-3.
- [5] G. Petrucci, M. Rospocher, C. Ghidini, Expressive ontology learning as neural machine translation, *J. Web Semant.* 52-53 (2018) 66–82. URL: <https://doi.org/10.1016/j.websem.2018.10.002>. doi:10.1016/j.websem.2018.10.002.
- [6] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, O. Etzioni, Open information extraction from the web, in: Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI'07, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007, p. 2670–2676.
- [7] L. Cui, F. Wei, M. Zhou, Neural open information extraction, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 407–413.
- [8] W. M. Aalst, Business process management as the "killer app" for petri nets, *Softw. Syst. Model.* 14 (2015) 685–691.
- [9] F. Friedrich, J. Mendling, F. Puhmann, Process model generation from natural language text, in: H. Mouratidis, C. Rolland (Eds.), *Advanced Information Systems Engineering - 23rd International Conference, CAiSE 2011, London, UK, June 20-24, 2011. Proceedings*, volume 6741 of *Lecture Notes in Computer Science*, Springer, 2011, pp. 482–496.
- [10] H. van der Aa, C. D. Ciccio, H. Leopold, H. A. Reijers, Extracting declarative process models from natural language, in: P. Giorgini, B. Weber (Eds.), *Advanced Information Systems Engineering - 31st International Conference, CAiSE 2019, Rome, Italy, June 3-7, 2019. Proceedings*, volume 11483 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 365–382.
- [11] C. Qian, L. Wen, A. Kumar, L. Lin, L. Lin, Z. Zong, S. Li, J. Wang, An approach for process model extraction by multi-grained text classification, in: S. Dustdar, E. Yu, C. Salinesi, D. Rieu, V. Pant (Eds.), *Advanced Information Systems Engineering - 32nd International Conference, CAiSE 2020, Grenoble, France, June 8-12, 2020. Proceedings*, volume 12127 of *Lecture Notes in Computer Science*, Springer, 2020, pp. 268–282.
- [12] P. Bellan, M. Dragoni, C. Ghidini, A qualitative analysis of the state of the art in process extraction from text, in: G. Vizzari, M. Palmonari, A. Orlandini (Eds.), *Proceedings of*

the AIXIA 2020 Discussion Papers Workshop co-located with the the 19th International Conference of the Italian Association for Artificial Intelligence (AIXIA2020), Anywhere, November 27th, 2020, volume 2776 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 19–30.

- [13] P. Bellan, Process extraction from natural language text, in: W. M. P. van der Aalst, J. vom Brocke, M. Comuzzi, C. D. Ciccio, F. García, A. Kumar, J. Mendling, B. T. Pentland, L. Pufahl, M. Reichert, M. Weske (Eds.), *Proceedings of the Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Track at BPM 2020 co-located with the 18th International Conference on Business Process Management (BPM 2020)*, Sevilla, Spain, September 13-18, 2020, volume 2673 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 53–60.
- [14] K. Ethayarajh, D. Jurafsky, Utility is in the eye of the user: A critique of NLP leaderboards, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, Online, November 16-20, 2020, Association for Computational Linguistics, 2020, pp. 4846–4853.
- [15] P. Bellan, H. van der Aa, M. Dragoni, C. Ghidini, S. P. Ponzetto, PET: An annotated dataset for process extraction from natural language text tasks, in: *Proceedings of the 1st Workshop on Natural Language Processing for Business Process Management (NLP4BPM)*, 2022. To appear. Also available at <https://arxiv.org/abs/2203.04860>.
- [16] F. Nanni, G. Glavaš, S. P. Ponzetto, S. Tonelli, N. Conti, A. Aker, A. P. Apro시오, A. Bleier, B. Carlotti, T. Gessler, T. Henrichsen, D. Hovy, C. Kahmann, M. Karan, A. Matsuo, S. Menini, D. Nguyen, A. Niekler, L. Posch, F. Vegetti, Z. Waseem, T. Whyte, N. Yordanova, Findings from the hackathon on understanding euroscepticism through the lens of textual data, in: D. Fišer, M. Eskevich, F. de Jong (Eds.), *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, European Language Resources Association (ELRA), Paris, France, 2018.
- [17] H. van der Aa, H. Leopold, H. A. Reijers, Detecting inconsistencies between process models and textual descriptions, in: H. R. Motahari-Nezhad, J. Recker, M. Weidlich (Eds.), *Business Process Management - 13th International Conference, BPM 2015*, Innsbruck, Austria, August 31 - September 3, 2015, *Proceedings*, volume 9253 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 90–105. URL: [https://doi.org/10.1007/978-3-319-23063-4\\_6](https://doi.org/10.1007/978-3-319-23063-4_6). doi:10.1007/978-3-319-23063-4\_6.
- [18] G. Adamo, S. Borgo, C. Di Francescomarino, C. Ghidini, N. Guarino, E. M. Sanfilippo, Business processes and their participants: An ontological perspective, in: *Proceedings of the 16th International Conference of the Italian Association for Artificial Intelligence (AI\*IA 2017)*, volume 10640 of *Lecture Notes in Computer Science*, Springer International Publishing, 2017, pp. 215–228.
- [19] F. Friedrich, Automated generation of business process models from natural language input, M. Sc., School of Business and Economics. Humboldt-Universität zu Berli (2010).
- [20] G. Hripcsak, A. S. Rothschild, Technical brief: Agreement, the f-measure, and reliability in information retrieval, *J. Am. Medical Informatics Assoc.* 12 (2005) 296–298.
- [21] N. Okazaki, Crfsuite: a fast implementation of conditional random fields (crfs), 2007.