A Qualitative Analysis of the State of the Art in Process Extraction from Text

Patrizio Bellan[a,b], Mauro Dragoni[a] and Chiara Ghidini[a]

[a]Fondazione Bruno Kessler, via Sommarive 18, 38050 Povo, Italy
[b]Free University of Bozen-Bolzano, Bolzano, Italy

Abstract

Within a company, processes are typically documented in form of unstructured textual information. To exploit all the techniques of Business Process Management and Process Mining, process models need to be represented in a formal (or semi-formal) representation, the process model diagram. However, manually obtaining an initial process model out of a process description document is a time consuming and cost intensive operation. Some initial solutions to address the challenge of process extraction from text have been proposed in the literature. But, the analysis of state of the art contributions reveals that this line of research has not reached its maturity yet and that process extraction from text can be considered an unresolved problem still in an early stage of development. Indeed, these contributions mainly adopt *ad-hoc* solutions based on rules, word-lists, and heuristics.

In this paper, we adopt the instrument of *qualitative analysis* on state-of-the-art approaches and tools to shed light on current limitations of the process extraction from text area. In addition to an analysis of the main reference papers we test reference tools on samples of text extracted from real documents describing Standard Operating Procedures that exhibit a greater complexity than the publicly available procedural descriptions so far used as reference text by the process extraction from text community. The analysis reveals the inability for those approaches to perform well in real scenarios.

The discussion of the results illustrates open points, fundamental challenges to solve, and gaps to fill. It also suggests new ideas on how to tackle some of the identified limitations which we intend to pursue in the future.

Keywords

Natural Language Processing, Information Extraction, Business Process Model

1. Introduction

Business Process Management (BPM) is a discipline that aims to discover, design, analyze, measure, improve, optimize, and manage business processes. A *business process* is a collection of ordered activities to model a specific business objective (typically represented in diagram) [1]. Process model diagrams are particularly important as most of the methodologies and techniques in the BPM field require them as a mean to analyze the real processes. Unfortunately, the initial elicitation of a process model diagram is a time consuming and cost intensive operation [2] that could require up to the 60% of the time spent in a project [3]. Therefore, there is an interest in discovering novel algorithmic procedures to alleviate the initial creation of process models.

AIxIA 2020 Discussion Papers Workshop
pbellan@fbk.eu (P. Bellan); dragoni@fbk.eu (M. Dragoni); ghidini@fbk.eu (C. Ghidini)
0000-0002-2971-1872 (P. Bellan); 0000-0003-0380-6571 (M. Dragoni); 0000-0003-1563-4965 (C. Ghidini)

Process Mining refers to the set of algorithmic methodologies that aim to automatically extract a process model as described in data. Typically these techniques exploit transactional data stored into so-called event-log. The fact that processes are often described as unstructured textual information has originated research efforts devoted to develop techniques able to automatize process extraction from text. Similarly to what has happened in other disciplines, such as for instance, ontology extraction from text [4], these techniques can be then embedded in modeling tools [5].

Process extraction from text can be regarded as the specific problem of finding an algorithmic function to generate a process model diagram from its procedural description. The ambiguous nature of natural language, the different writing styles, and the variability of domains to which the processes refer make this task extremely challenging. Among the contributions proposed in the literature, the work of Friedrich et al. [6], published in the 2011, is still regarded as one of the main state-of-the-art contributions, as emphasized in recent surveys [2, 7]. These surveys also highlight that, after almost ten years of research, this task is far from being resolved. This is due to two main factors: first, according to Riefer et al. [8], a number of contributions in this area date back to several years ago and thus they may be considered outdated, given the advances of Natural Language Processing (NLP) techniques in the last few years; second, according to Maqbool et al. [2], current approaches may be not able to scale up to real world scenarios.

Since a golden standard data set that can be used to compare different approaches is missing, performing an empirical evaluation to address the status of process extraction from text is particularly challenging. As a first step towards this direction we decided to perform a qualitative analysis of state of the art approaches and tools in process extraction from text to understand its limitations and challenges to be addressed. In particular, we focused on two state-of-the-art tools for the extraction of imperative [6] and declarative [9] process models and we used them on an heterogeneous selection of Standard Operating Procedures (SOPs) descriptions adopted in a company we collaborate with[1].

Our analysis shows the inability of current state-of-the-art tools to scale up to real scenarios. The discussion of the problems of current approaches highlights the limitations of the state-of-the-art contributions. It also illustrates open points, fundamental challenges to solve, and gaps to fill. Whereas we suggest new ideas on how to tackle some of them, which we intend to pursue in the future, other points are left for further discussions.

2. Problem Definition

The task of *process extraction from text* can be defined as the problem of generating a process model diagram from its procedural description by means of an algorithmic function *f*. This is a complex task since there is the need of taking care simultaneously of the multiple linguistics levels (syntactical, semantics and pragmatics) as well as of linguistics phenomena such as syntactic leeway and relevance. To reduce the overall complexity, the problem can be further broken down into two main stages: stage f_a, called *text-to-world model*, and stage f_b, called *world model-to-diagram*.

[1]These documents cannot be shared due to confidentiality agreements.

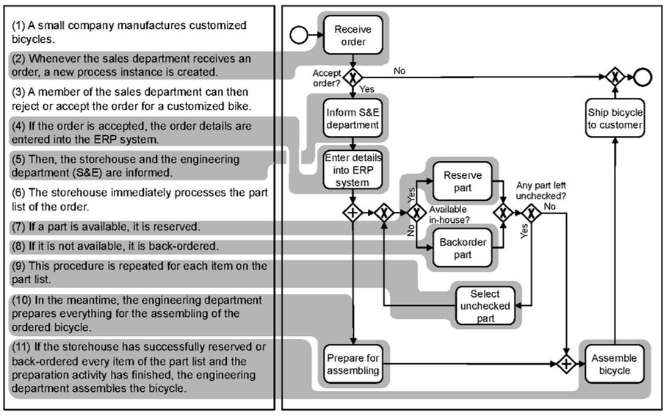

Figure 1: An example of text-to-model mapping taken from [7].

f_a performs the extraction of process elements from the text and memorize them in a structured representation, also called World Model. f_a can be further broken down into smaller tasks to better handle the problem complexity. For example, $f_{a.i}$ may take care of resolving anaphoric references. $f_{a.ii}$ may take care of filtering out textual fragments that are uninformative w.r.t. the process description. $f_{a.iii}$ may aim to extract process elements (i.e., activity, roles, events) and process structures (i.e., gateway's branches) from text and represent them in the diagram. And so on.

The second component f_b builds up the process model diagram starting from the world model. Also f_b can be further broken down into smaller tasks. As an example, $f_{b.i}$ can take care of adding process elements and process structures not explicitly described in the text, but necessary to correctly create the process model, such as the activity "ship bicycle to customer" in Figure 1. Another task, $f_{b.ii}$, may aim to connect process elements together following the same logic conveyed in the textual description. Here the challenge is to handle those cases in which the order provided in the process description is not the (logically, pragmatically, semantically) correct one in the process, (see for instance sentences 4 and 5 in Figure 1). Finally, a task $f_{b.iii}$ may generate textual labels for each process element and generate the process model diagram.

3. An Analysis of the Literature

In this section we provide a concise summary of our analysis of the state-of-the-art approaches for process extraction from text. The papers we considered are [10, 6, 11, 12, 13, 14, 15, 16, 17, 18, 19, 9, 20, 21, 22] and were chosen for the impact of their citation and relevance for this topic. In general, they mostly provide *ad-hoc* rules to perform the extraction of process elements and we compare them on the basis of: the input text accepted, the intermediate representation adopted, the output process generated, and the experimental evaluation made.

Input. The most common type of input representation is a completely unstructured natural language text tat corresponds to a process description [6, 20, 18, 19, 21, 12, 17, 22]. In [11, 16] the adoption of templates opened the possibility to analyse *every type of documents*. In [10] the focus is on users' interaction in which processes instances are captured by mean of *stories* told by the users them-self. Finally, the work in [14, 15] aims at detecting ambiguities and inconsistencies between a process description and its corresponding process model; therefore the input is restricted to text-model pairs only.

Intermediate representation. The most common type of *intermediate representation* is the CREWS [23] world model, adopted in [10, 6, 19]. A table-based representation, either in form of a structured table or in form of a spreed-sheet, is used in [13, 15, 16, 18, 9].

Output. The contributions can be divided in two groups. The first group, [11, 13, 16, 6, 20, 18, 19, 12, 17, 14, 15, 22], includes approaches that generate an *imperative* process model diagram using the BPMN 2.0 modelling language [24].The second group includes contributions that generate a *declarative* process model in which the behavior of a process is expressed using constraints on the relations between the process elements. van der Aa et al. [9] adopt DECLARE [25] as declarative language, whereas López et al. [21] represent the process model using DCR Graph.

Experimental Evaluation. The analysis of the experimental evaluations reveals a lack of uniformity that makes a comparison of the proposed contributions rather difficult. The works proposed in [11, 10, 13, 16, 19, 12, 15, 9] adopt well-known metrics (precision, recall, F1, and accuracy) to quantitatively evaluate the performance of the proposed systems on the quality of the elements extracted from the process textual description. In [6, 20] a graph-based measure quantitatively evaluate the quality of the process model created by the proposed systems from the textual description of a process. The work in [14] adopts the metric proposed in [26] that takes into account the semantics of the sentence and of the process element label pair being compared this because the focus of the research is on process model alignment. The work in [22] defines a new metric called *information gain* to properly measure the reduction of uncertainty among all the possible interpretation of a process description. Regarding the experimental evaluation data set adopted, the works presented in [6, 11, 14, 15, 16, 17, 22, 19, 9] all adopt the data set proposed in [6], or a subset of it.

3.1. Limitations of the literature

Contributions proposed in the literature attempted to solve process extraction from text task mainly with *ad-hoc* methods highly tailored to specific input data sets. The analysis of the literature above revealed three main limitations regarding the techniques adopted, the data being analyzed, and the metrics adopted to judge the quality of the proposed systems.

Limitation 1: Problems with the techniques adopted. All the work we have analysed are highly tailored to a specific form of input data. Different scenarios, and different process analysts, may exhibit different writing styles with different use of words (and their related meaning) to describe processes. Therefore, the proposed approaches may be not able to generalize well among different styles and/or word uses unless a new *ad-hoc* set of rules is added to the system.

Limitation 2: Problems with the data. The reference dataset proposed in [6] was not validated. Indeed, the vast majority of the process model diagrams in this data set were translated from other visual languages into BPMN. Also, some process descriptions were translated from German to English by the same authors without validation. Thus, this data set could be considered good for a preliminary development, but it is easy to see that it cannot represent an actual and solid benchmark. Also, this data set is not a representative sample of the variety of process description-process diagram pairs one may have in real scenarios (see Section 4.1).

Limitation 3: Problems with the metrics. The evaluation of the quality of a process model obtained from text is in all the works above either relying on information retrieval metrics, or on graph-based metrics. However, these metrics are not refined enough to make a distinction between the possible types of errors that can be generated, which could be more or less severe, and also the different correct ways to convey the same semantic in a process diagram.

4. Qualitative Analysis

In this Section, we present a *qualitative analysis* of the diagrams generated by state-of-the-art systems on real-world documents. We limit our evaluation to such an analysis due to the missing of a benchmark enabling their fair quantitative comparison. However, this kind of analysis has the advantage of providing an overall understanding of systems' behavior and it allows to highlights their current issues and limitations. In particular, we tested the *imperative* [6] and the *declarative* [9] systems on a sub-set of eleven representative and heterogeneous documents extracted from our SOP archives.

The work presented in [6] is considered the state-of-the-art technique about the generation of imperative process model out of a text. This contribution aims to extract a complete set of business elements through an extensive use of rules and heuristics then used to generate the corresponding process diagram. The pipeline proposed in [6] is composed of three modules: *Sentence Level Analysis*, *Text level Analysis*, *Process Model Generation*. The *Sentence Level Analysis* module performs common natural language processing (NLP) tasks: tokenization of sentences and words, and abbreviation resolution. Then, the text is parsed by the Stanford CoreNLP Library [27] to obtain a tree-based representation of sentences. This is done whether it is possible to determine (i) if the verb of the sentence is active or passive and (ii) if both Actor and Action are extracted. In this step sample-wise sentences are filtered out using a keywords-based list. The *Text level analysis* module performs the analysis of constituent relations within sentences and the resolution of co-references and relative references. Conditional markers are checked against a word-list of conditional indicator to determine the gateway's type (concurrent, parallel, inclusive or exclusive). Here, *WordNet* and *VerbNet* are used to increase the generalization capability of the tool. Information contained in the world model is enhanced and combined with the one extracted in this stage to tackle the problem of actions that span over multiple sentences. Finally, the *Process Model Generation* module transforms the world model into the equivalent BPMN representation and creates the corresponding model's labels for each diagram's objects.

The state-of-the-art technique related to the discovery of knowledge-intensive business process constraints from text is proposed in [9]. This work relies on a tailored NLP pipeline addresses several challenges:

- the use of synonyms for describing Activities;
- the unordered description of the process elements from the execution perspective;
- the identification of *noun-based actions*;
- the detection of *constraint restrictiveness*, i.e. to make a distinction among the binary constraints kind;
- the detection of *negation* in process description. This aspect may lead to a changing in the meaning of a process element or a branch;
- the identification of *multi-constraint descriptions* within a single sentence, a scenario usually observed through the presence of coordinating conjunctions.

The algorithm identifies the activities and their inter-relations after a deep analysis of the text semantics. The decomposition and the analysis of the input in order to fill slot-templates corresponding to the described declarative constraint is performed through three steps. First, the process starts with a linguistic pre-processing stage aims to extract semantic components from the given input in form of typed dependency relations. Second, the presence of temporal verbs is checked from the input sentence in order to detect verb-based and noun-based activities. Third, the data extracted in the previous steps are exploited to fill specific *slot*-templates of declarative constraints. A collection of 103 constraint descriptions extracted from the data set proposed in [6] are used to measure the quality of this approach. Well-known metrics of Precision, Recall and F1-score have been adopted. The method achieves an overall precision of 0.77 and a recall of 0.72, yielding an F1-score of 0.74.

4.1. Data set

The documents we used are significantly different with respect to the ones exploited by [6]. In particular, each document is made of almost ten pages each with many sections, typically composed by very long sentences with an extensive use of topic specific terms and abbreviations. These documents vary greatly in the writing style adopted because they were written by different authors along the years. As example, some procedure descriptions are written using long sentences and no bullet-lists while others contain procedural descriptions structured using bullet-lists and other formatting elements.

We report below two examples taken from a document of our SOPs' archive and a full text of a process description proposed in [6], respectively. The reader can easily notice the difference between the two texts. Differently from the text contained in a SOP, the sample proposed in [6] is easier to analyze due to the absence of uninformative text and to the use of a pattern-oriented technique for describing process elements.

Excerpt of a Standard Operative Procedure. Access for new Users. Every User has an unique user name, which is the same for all three environments. The password is different for every environments and every User is responsible to keep credentials as appropriate and to not share them with other Users or people. The password shall be changed periodically,

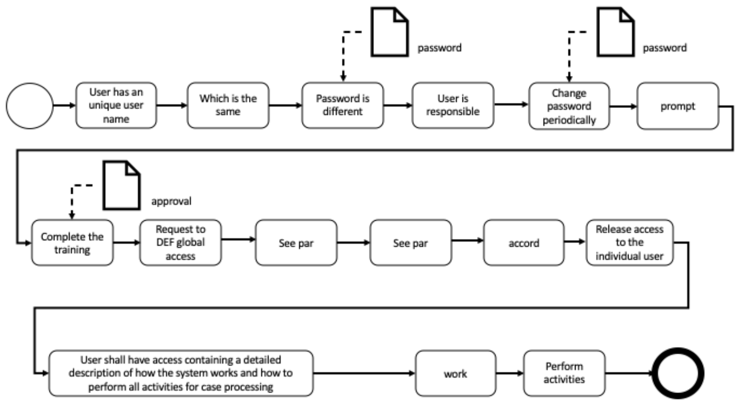

Figure 2: BPMN diagram of the excerpt reported in Sec.4.1 generated by [6].

upon prompt from the system, before the expiry date. Third Party Provider staff and other consultant(s) may gain access the database only after specific training performed by: - an XIS trainer, who will release a training certificate or - a ABC Company experienced Standard User who shall train the new User on the job: certificate is released by the ABC Company experienced Standard User after the User has successfully processed at least 10 case reports in the Sandbox. When the training is completed, upon Qualified Person for approval, a ABC Company Administrator User shall request to DEF company global access for the individual user (see par. X.xx), assigning specific roles according to the activities to be performed by the User. Access is released by DEF company to the individual User who receives username and password directly from the Help-desk. Users shall have access to the system - User Manuals available on the knowledge portal, containing a detailed description of how the system works and how to perform all activities for case processing.

A sample of the data set used in [6]. The MPON sents the dismissal to the MPOO. The MPOO reviews the dismissal. The MPOO opposes the dismissal of MPON or the MPOO confirms the dismissal of the MPON.

5. Results

In this Section, we report our observations on the diagrams produced by the tools in [6] and in [9] over our dataset.

The tool in [6] failed to produce a diagram for all the entire descriptions we tested it on, even when sentences were manually simplified (e.g., fixing missing punctuation). Therefore, we decided to test it with self-contained section's paragraphs or even a single sentence. To help the comprehension, Figure 5 depicts the diagram generated on the excerpt shown in Section 4.1). In most of the cases, the diagrams present wrong control flows consisting of either consecutive or parallel activities. Moreover, we observed the following errors:

Sentence Tokenization Errors The tool cannot detect the boundary of sentences correctly. In most of the cases, if an abbreviation like *e.g.* is present in the sentence, this is erroneously broken after the last dot (i.e., after the *g*).

Word Tokenization errors When two words are separated by a slash (like "user/manager"), the two words are not tokenized, but treated as a single unknown word.

Multi-Words-Expressions MWE are not taken into account. Therefore the internal parser fails to correctly handle these cases. This problem affects the generation of the dependency tree and, consequently, the generation of the process model diagram. For example, the word "*file system*" was erroneously parsed and treated as two atomic words: the verb *to file* and the noun *system.* This error propagated to the dependency tree of the sentence and caused the generation of the wrong activity *to file* in the diagram.

Missing Time Expressions Expressions representing BPMN time events (e.g., *Back up for 10 years*) are not extracted. These expressions are usually expressed with patterns that are complex to detect or by words that are more complex than the ones presents in word-lists and heuristics exploited by the tool.

Wrong Labels Almost all the labels of the process elements are wrong because of two types of reasons: a first group of labels is generated using (almost) the entire sentence under analysis (e.g., the third last activity in Figure 5); a second group of labels consists of a single word (e.g., *work* and *TO* in Figure 5) that does not describe the activity itself.

Wrong or Missing Events For instance, the *receiving* event described in the last sentences of the excerpt is missing in the diagram.

Erroneous Diagram Sometimes the semantics of the process in the text differs with the one represented in the diagram. For example, the tool represented the textual fragment "*...those instructions and procedures <u>are stored</u> in the standard procedure document number XXX*" as two different diagram's activities: "*<u>report</u> procedure in the standard procedure document number XXX*" and "*<u>report</u> instruction in the standard procedure document number XXX*". The change in meaning happens because the sentence means "to read instructions and procedures from the document" whereas the meaning conveyed in the diagram is "to write instructions and procedures in the document".

Wrong or Missing Roles, Activities, and Resources The diagram lacks pools and lanes representing the actors performed activities.

The tool in [9] analyses a single sentence at a time and it generates constraints related to the given sentence. Thus we provided as input each sentence separately. In general, the tool correctly recognized negative constraints. However, the following errors have been observed:

Wrong Labels Some activity labels are wrong because they either consist of the entire sentence, or they repeat sentence fragments in the label (e.g., *drug safety department drug safety department...*).

Missing Activities Some activities described in the text are not detected. This problems is related to the erroneous dependency relations generation.

Uninformative Textual Fragments not Discharged For example, the text contained in a parenthesis block, that often describes a concrete example, is never discharged. This problem may damage the dependency parser by making it more prone to errors and it may damage the readability of the results.

Wrong Constraint Type This problem is due to the limited set of declarative constraints considered. If the text refers to a different type of constraints, the tool anyway tries to represent it in one of the constraints considered.

6. Discussion

The qualitative analysis conducted on the current state-of-the-art tools reveals that the task of process extraction from text is far from being a solved problem. Our findings agree with [2] that rules, patterns, templates, and tailored approaches resulted to be effective only in the cases presented in the respective papers.

We tracked three source of errors: erroneous dependency tree generation, failed POS parsing, and errors coming from not filtering uninformative textual fragments out. The results suggest that these problems cannot be fixed by the adoption of word-lists because they are insufficient to handle the complexity of real world scenarios.

Along the errors found, label's phenomena affected the two state-of-the-art negatively on three aspects: degrades the readability of textual labels; makes these activities useless; makes the diagram rather difficult to understand. We argue that the task of *filtering uninformative textual fragments out* ($f_{a.ii}$) is an important pre-processing step to investigate because in real world procedural documents the process description is surrounded by useless information (w.r.t. a process description) that may behaves like noise in the later stages. The tasks of extracting process elements and process structures ($f_{a.iii}$) has to tackle the ambiguous nature of natural language and different linguistics styles. A promising solution to reduce errors relies on the adoption of statistical machine learning classifiers and word embeddings. Since not all the process elements and process structures (such as a control flow structure) can be modeled under the same operational definition, each process element and each process structure must be considered as a single category of problems. In particular, Roles and Artifacts (i.e., data store), can be defined as a binary classification problem. But, more complex process elements, such as Activities, Events and Gateways, need to be modeled as a multi-levels classification problem, likes a Temporal Relation Extraction problem, in which first the basic elements are identified (i.e., the condition state of a gateway or an activity), and then, the relation between process elements can be correctly classified. However, in the business process community, there is not an unified and clear definition to apply to guide extraction of process elements. A complementary strategy to alleviate these problems points out to leverage NLP solutions found for similar but different scenarios and research areas investigating transfer-learning techniques. For example, *How does it possible to adapt an existing (pre-trained) Event-Detection system to* *detect temporal events?* This kind of questions were never proposed in this topic, but they need to be investigated in order to make advances in this research line.

In addition, data-augmentation techniques can be used to expand the data availability, allowing training more sophisticated classifiers on a broader set of samples that should lead to better classification performance. In order to leverage statistical learning and increase the performance of data-augmentation, a source of annotate data (golden standard) is required. However, this kind of data is missing. In addition, no works present an annotation guidelines to guide data annotation of process elements in textual description. This may be regarded as the first future challenge to tackle, because it is impeding the development of this topic, nowadays.

We speculated about the first steps to perform with the goal of process extraction from text. Unfortunately, it is difficult to make clear hypothesis regarding the second stage of this task (f_b). In particular, $f_{b.i}$ that regards adding process elements and process structures not explicitly described in the text, requires a semantic reasoning on both text and model, simultaneously. Here, maybe an ontology could be employed. However, it is even more difficult to define what *reasoning* means in this case. A temporal reasoning may be required to correctly connect process elements together following the same logic conveyed in the textual description ($f_{b.ii}$). This is so because there are cases of mismatch between the textual description order and the logic conveyed in the semantic of the text. The last point, generates textual labels ($f_{b.iii}$), is the only one that have a clear definition in the literature. But, a technique able to extract only the useful information (words) out of a sentence depends on the success of the task $f_{a.iii}$ (for each process element). Finally, the last problem to solve regards how to solve the change in meaning problem, because it could have negative consequences.

About the Limitations found with the literature analysis, we proposed a partial answer, proposing possible research directions, to overcome problems related with the methodological approach (L1) and problems with the data (L2). About the limitation with the metrics (L3), some forms of reasoning over process models 'semantic should be required to judge as equivalent two different ways of modeling the same process. However, it is difficult to make hypothesis on how to solve this limitation.

7. Conclusions

Process extraction from textis a complex task far to be solved. In this work, we performed a qualitative analysis on state-of-the-artsystems to shed light on the issues and limitations of such approaches when applied to real-world natural language processes description. The discussion allowed us to speculate on some aspects, but revealed also some open points that leave rooms for future discussions.

References

[1] M. Dumas, M. L. Rosa, J. Mendling, H. A. Reijers, Fundamentals of Business Process Management, Springer, 2013.

[2] B. Maqbool, F. Azam, M. W. Anwar, W. H. Butt, J. Zeb, I. Zafar, A. K. Nazir, Z. Umair, A comprehensive investigation of BPMN models generation from textual requirements - tech-

niques, tools and trends, in: ICISA, volume 514 of *Lecture Notes in Electrical Engineering*, Springer, 2018, pp. 543–557.

[3] J. Herbst, An inductive approach to the acquisition and adaptation of workflow models, in: Proceedings of the IJCAI'99 Workshop on Intelligent Workflow and Process Management: The New Frontier for AI in Business, 1999, pp. 52–57.

[4] 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.

[5] C. Ghidini, M. Rospocher, L. Serafini, Serafini: Modeling in a wiki with moki: Reference architecture, International Journal On Advances in Life Sciences 4 (2012) 111–124.

[6] F. Friedrich, J. Mendling, F. Puhlmann, Process model generation from natural language text, in: Advanced Information Systems Engineering - 23rd International Conference, CAiSE 2011, London, UK, June 20-24, 2011. Proceedings, volume 6741 of *LNCS*, Springer, 2011, pp. 482–496.

[7] H. van der Aa, J. Carmona, H. Leopold, J. Mendling, L. Padró, Challenges and opportunities of applying natural language processing in business process management, in: Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018, Association for Computational Linguistics, 2018, pp. 2791–2801.

[8] M. Riefer, S. F. Ternis, T. Thaler, Mining process models from natural language text: A state-of-the-art analysis, Multikonferenz Wirtschaftsinformatik (MKWI-16), March (2016) 9–11.

[9] H. van der Aa, C. D. Ciccio, H. Leopold, H. A. Reijers, Extracting declarative process models from natural language, in: Advanced Information Systems Engineering - 31st International Conference, CAiSE 2019, Rome, Italy, June 3-7, 2019, Proceedings, volume 11483 of *LNCS*, Springer, 2019, pp. 365–382.

[10] J. C. de A. R. Gonçalves, F. M. Santoro, F. A. Baião, Let me tell you a story - on how to build process models, J. UCS 17 (2011) 276–295.

[11] L. Ackermann, B. Volz, model[nl]generation: natural language model extraction, in: Proceedings of the 2013 ACM workshop on Domain-specific modeling, DSM@SPLASH 2013, Indianapolis, Indiana, USA, October 27, 2013, ACM, 2013, pp. 45–50.

[12] K. P. Sawant, S. Roy, S. Sripathi, F. Plesse, A. S. M. Sajeev, Deriving requirements model from textual use cases, in: 36th International Conference on Software Engineering, ICSE '14, Companion Proceedings, Hyderabad, India, May 31 - June 07, 2014, ACM, 2014, pp. 235–244.

[13] E. V. Epure, P. Martín-Rodilla, C. Hug, R. Deneckère, C. Salinesi, Automatic process model discovery from textual methodologies, in: 9th IEEE International Conference on Research Challenges in Information Science, RCIS 2015, Athens, Greece, May 13-15, 2015, IEEE, 2015, pp. 19–30.

[14] H. van der Aa, H. Leopold, H. A. Reijers, Detecting inconsistencies between process models and textual descriptions, in: Business Process Management - 13th International Conference, BPM 2015, Innsbruck, Austria, August 31 - September 3, 2015, Proceedings, volume 9253 of *LNCS*, Springer, 2015, pp. 90–105.

[15] H. van der Aa, H. Leopold, H. A. Reijers, Comparing textual descriptions to process models

- the automatic detection of inconsistencies, Inf. Syst. 64 (2017) 447–460.
[16] R. C. B. Ferreira, L. H. Thom, M. Fantinato, A semi-automatic approach to identify business process elements in natural language texts, in: ICEIS 2017 - Proceedings of the 19th International Conference on Enterprise Information Systems, Volume 3, Porto, Portugal, April 26-29, 2017, SciTePress, 2017, pp. 250–261.
[17] T. S. Silva, L. H. Thom, A. Weber, J. Palazzo Moreira de Oliveira, M. Fantinato, Empirical analysis of sentence templates and ambiguity issues for business process descriptions, in: On the Move to Meaningful Internet Systems. OTM 2018 Conferences - Confederated International Conferences: CoopIS, C&TC, and ODBASE 2018, Valletta, Malta, October 22-26, 2018, Proceedings, Part I, volume 11229 of *LNCS*, Springer, 2018, pp. 279–297.
[18] K. Honkisz, K. Kluza, P. Wisniewski, A concept for generating business process models from natural language description, in: Knowledge Science, Engineering and Management - 11th International Conference, KSEM 2018, Changchun, China, August 17-19, 2018, Proceedings, Part I, volume 11061 of *LNCS*, Springer, 2018, pp. 91–103.
[19] H. Leopold, H. van der Aa, H. A. Reijers, Identifying candidate tasks for robotic process automation in textual process descriptions, in: Enterprise, Business-Process and Information Systems Modeling - 19th International Conference, BPMDS 2018, 23rd International Conference, EMMSAD 2018, Held at CAiSE 2018, Tallinn, Estonia, June 11-12, 2018, Proceedings, volume 318 of *LNBIP*, Springer, 2018, pp. 67–81.
[20] X. Han, L. Hu, Y. Dang, S. Agarwal, L. Mei, S. Li, X. Zhou, Automatic business process structure discovery using ordered neurons LSTM: A preliminary study, CoRR abs/2001.01243 (2020).
[21] H. A. López, S. Debois, T. T. Hildebrandt, M. Marquard, The process highlighter: From texts to declarative processes and back, in: Proceedings of the Dissertation Award, Demonstration, and Industrial Track at BPM 2018 co-located with 16th International Conference on Business Process Management (BPM 2018), Sydney, Australia, September 9-14, 2018, volume 2196 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2018, pp. 66–70.
[22] H. van der Aa, H. Leopold, H. A. Reijers, Checking process compliance against natural language specifications using behavioral spaces, Inf. Syst. 78 (2018) 83–95.
[23] C. B. Achour, Guiding scenario authoring, in: Information Modelling and Knowledge Bases X: 8th European-Japanese Conferences on Information Modelling and Knowledge Bases, EJC 1998, Vammala, Finland, May 26-29, 1998, volume 51 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 1998, pp. 152–171.
[24] OMG, Business Process Model and Notation (BPMN), Version 2.0, 2011.
[25] M. Pesic, H. Schonenberg, W. M. P. van der Aalst, DECLARE: full support for loosely-structured processes, in: 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007), 15-19 October 2007, Annapolis, Maryland, USA, IEEE Computer Society, 2007, pp. 287–300.
[26] R. Mihalcea, C. Corley, C. Strapparava, Corpus-based and knowledge-based measures of text semantic similarity, in: Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1, AAAI'06, AAAI Press, 2006, p. 775–780.
[27] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, D. McClosky, The Stanford CoreNLP natural language processing toolkit, in: Association for Computational Linguistics (ACL) System Demonstrations, 2014, pp. 55–60.