Mining Process Models from Natural Language Text: A State-of-the-Art Analysis

Maximilian Riefer[1], Simon Felix Ternis[2] and Tom Thaler[2]

[1] Saarland University, Institute for Information Systems (IWi), s9marief@stud.uni-saarland.de
[2] Institute for Information Systems at the German Research Center for Artificial Intelligence (DFKI GmbH) and Saarland University, {firstname.lastname}@iwi.dfki.de

Abstract

Workflow projects are time-consuming processes. They include the knowledge extraction and the creation of process models. The necessary information is often available as textual resources. Therefore, process model mining from natural language text has been a research area of growing interest. This paper gives an overview of the current state-of-the-art in text-to-model mining. For this purpose, different approaches focusing on business process models are presented, analyzed and compared against each other on a theoretical and technical level. The resulting overview covers both advantages and disadvantages of current techniques. This should establish a sturdy basis on which further research can be conducted.

1 Introduction

Organizations are constantly trying to analyze and improve their business processes. This is only possible if the knowledge about the processes is available in a structured form like a business process model. The extraction of this knowledge and the generation of process models is a time-consuming and costly process. At the same time, 85% of the knowledge and information are estimated to be available in an unstructured form, mostly as text documents (Blumberg and Atre 2003). Modern natural language processing techniques render it possible to mine those text documents for process models. The automatic generation of process models from natural language text could speed up the whole workflow project of an organization. It gives people with no knowledge about process modeling the possibility to create process models, which is an important goal in view of the fact that structured data becomes more important. This field of research has been attaining more and more attention in recent years. Text Mining approaches have been developed for UML class diagrams (Bajwa and Choudhary 2011), entity relationship models (ERM) (Omar et al. 2008) or business process models (Friedrich et al. 2011). Current approaches do not aim at replacing an analyst but at helping him to create better models in less time.

The developed text mining approaches mainly focus on several specific model types and, thus, on specific contexts. A general overview in terms of a state-of-the-art analysis is missing. This paper

aims at giving an overview of existing approaches, the different natural language methods they use and a comparative analysis of the existing techniques. The focus of this paper is laid on the comparative analysis of the identified text mining approaches. Three main aspects are identified and analyzed: the textual input, the used NLP techniques and the model generation. Analyzing these aspects allows for a categorization of the existing approaches.

The used research methodology and the identification of the relevant literature are presented in section 2, while section 3 introduces the most important methods and terms for processing natural language texts. Section 4 gives an overview of current approaches. The comparative analysis, which consists of a detailed theoretical analysis and a proposed practical analysis, is conducted in section 5. The results are discussed in Section 6, followed by a conclusion in section 7.

2 Methodology

To define the current state of research and to identify different approaches for mining process models from natural language text, a systematical literature review was conducted. The three literature databases Google Scholar, SpringerLink and Scopus were used for the research. The following search keywords were derived from the title and thematic of this paper: *natural language processing*, *process model*, *process modeling* or p*rocess model generation*, *process model discovery*, *text mining*, *process mining* and *workflow*. These were used in various combinations. As the used keywords cover broad research areas, they lead to a high number of search results. That complicated the identification of the relevant literature. The search results were checked through a title and abstract screening to identify the relevant work. Hence, only publications which explicitly mention text-to-model transformations were considered as relevant. There turned out to be a problem with the author's way of describing their work: instead of referring to text mining or natural language processing, they often used the text type, such as use cases or group stories, to outline their work. The search in a database was aborted when a significant amount of repetitions or loss of precision was noticed. It turned out, that the keyword search provided a low degree of relevant papers. Hence the keyword search was skipped and changed to a cross reference search. Table 1 shows the literature search results.

	number of results		
search results / database	Google Scholar	SpringerLink	Scopus
„process mining“ „natural language processing“	394	441	713
„process modeling“ „ natural language“	5990	>10000	882
„process model“ „natural language“	>10000	>10000	2663
„process model discovery“ „natural language processing“	7	13	59
„process model generation“ „natural language processing“	50	21	169
„process mining“ „text mining“	525	575	1629
„workflow discovery“ „natural language processing“	18	3	7

Table 1: Results of literature research

Afterwards, further works were detected through a backwards search. The work of (Leopold 2013) provided a proficient starting point. The focus was set on approaches which generate a business process model. Five appropriate approaches could be identified:

1. BPMN model from text artefacts (Ghose et al. 2007)
2. BPMN model from group stories (Goncalves et al. 2009)
3. BPMN model from use cases (Sinha and Paradkar 2010)

4. BPMN model from text (Friedrich et al. 2011)
5. Model from text methodologies (Viorica Epure et al. 2015)

There seems to be a lack of publications from 2011 to 2015 and current works still refer to the aforementioned approaches (van der Aa et al. 2015). Because of the mentioned difficulties, it cannot be guaranteed, that the presented selection is exhaustive. The last approach does not build a BPMN model but is the most recent work deriving models from natural language texts which could be found. In the overview at hand, the different approaches will be presented and compared.

3 Introduction to Natural Language Processing

The processing of natural language requires different analyses of the text. This section introduces the most important methods and terms which are used for the identified approaches, whereby both syntactic and semantic analyses are performed. The syntactic analysis operates on a word at sentence level and annotates the words and phrases of a sentence, while the semantic analysis is used to provide an understanding of the semantic coherence of words and phrases in the text.

3.1 Syntactic Analysis

The syntactic analysis usually comprises three main parts: the tokenization, the part-of-speech (POS) tagging and the parsing. *Tokenization* describes the process of splitting a text into several parts, called tokens. A token is a string between two separators like spaces, tabs or periods and, hence, usually consists of a single word. A *POS tagger* tries to annotate the different words with their word form (e.g. verb, noun). Many different methods for POS tagging are available. On average, they achieve an accuracy of about 97% of correctly annotated tokens (Manning 2011) in the context of natural language texts. Then, a *parser* analyzes a given sequence of tokens and builds a syntax tree, following a given grammar. It can be differentiated between a shallow parser and a deep parser. A shallow parser often just searches for nominal phrases and verbal phrases as smallest constituents, while a deep parser tries to determine every word and phrase of a sentence.

3.2 Semantic Analysis

A semantic analysis is used to determine contextual meaning as e.g. homonyms or synonyms. For the semantic analysis, different semantic dictionaries and knowledge bases are used. Two well-known examples are WordNet (Miller 1995) and FrameNet (Baker and Sato 2003).

WordNet. "WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. […] The main relation among words in WordNet is synonymy, as between the words shut and close."[1]

FrameNet. "The FrameNet project is building a lexical database of English that is both human- and machine-readable, based on annotating examples of how words are used in actual texts. [...]. FrameNet is based on a theory of meaning called Frame Semantic [...]. The basic idea is straightforward: that the meanings of most words can best be understood on the basis of a semantic frame: a description of a type of event, relation, or entity and the participants in it."[2]

[1] Princeton University "About WordNet." WordNet. Princeton University. 2010. <http://wordnet.princeton.edu>
[2] https://framenet.icsi.berkeley.edu/fndrupal/about

Another part of semantic analysis is the anaphora resolution. An anaphora "is a word that refers to a word used earlier in a sentence and replaces it, for example the word 'it' in the sentence 'Joe dropped a glass and it broke'"[3]. Anaphora resolutions links noun phrases to the related pronoun. This is also important for text-to-model mining, because activities can be split up into different sentences and anaphora analysis helps to connect and recognize these sentences as one activity.

4 Overview on Existing Approaches

This section introduces the different approaches. Every approach is categorized in three aspects. Textual input refers to the required source document and its constraints. Text analysis introduces the used NLP technologies. Model generation describes the resulting model characteristics.

4.1 Process Discovery from Model and Text Artefacts (Ghose et al. 2007)

Textual input. As there are no explicit restrictions mentioned for text files, every textual input should be usable. Additionally, this approach is able to use existing BPMN models to check for consistency of the created models.

Text analysis. A text is analyzed in two ways. One is to look for text patterns which indicate process structures. Such a pattern could be "If < condition/event >, [then] < action >". A second analysis is performed with the Natural Language Toolkit (NLTK) (Bird et al. 2009). First, the text is annotated with POS tags and, then, a syntax tree is parsed by a shallow parser. In this tree, combinations of verb phrases (vp) and noun phrases (np) indicate activities (a) and actions. The sequence <vp,np> would be considered as an activity where (np) functioned as an object in difference to an action sequence <np, a> where (np) functions as an actor. Sequence flows are discovered through predefined signal words.

Model generation. No sound BPMN model is created. Discovered model parts are modeled without connection if no coherence is found.

4.2 Business process mining from group stories (Goncalves et al. 2009)

Textual input. The authors use group stories, a collective narrative technique, to extract knowledge from people who are involved in a process (Santoro et al. 2008). A story which is written by a team is likely to contain more useful information than a collection of individual interviews. A team consists of story tellers, moderators and analysts. A moderator is responsible for a correct and usable text format. The required form of input texts is not specified any further.

Text analysis. First, the text is tokenized, then, it is annotated with POS tags by a trigram tagger (Brants 2000) and parsed by a shallow parser. Activities, actors, actions and parameters are identified through templates based on patterns of verb- and noun-phrases. Flow elements are identified through keywords. No semantic analysis is performed. Due to the similarity of process models to scenarios, they use a variation of the CREWS scenario model to store the extracted information. This is based on a CREWS use case, which showed similarities to process descriptions. Both actions and actors are core concepts of a model. An action is performed by an actor and can have one or more start and end events (Maiden 1998).

[3] http://dictionary.cambridge.org/dictionary/english/anaphor

Model generation. The extracted model elements are modeled into BPMN models. These models do not have to be complete, e.g. they don't necessarily have start or end events. With the help of analysts, the story telling team generates the final model.

4.3 Use Cases to Process Specifications in Business Process Modeling Notation (Sinha and Paradkar 2010)

Textual input. Use cases are used as source documents. The authors did not provide a required form for use cases, the process descriptions have to be extracted manually. They define a use case as a sequence of statements, statements as a sequence of actions. A statement can be a conditional statement, which is an exception statement associated to the nearest preceding statements.

Text analysis. The text is tokenized and lemmatized. Afterwards, the POS tagging is conducted by a tagger based on a linear classification technique called robust risk minimization (Zhang et al. 2002). The text is parsed by a shallow parser based on "Finite State Transducer" (Boguraev and Neff 2008). For the semantic analysis with an anaphora resolution, domain specific databases are created manually. Flow elements are identified by keywords. The gained information is stored in a use case description meta model containing a model of the domain with its actors and business items and a use case model, in which the actions and initiating actors of every sentence are stored.

Model generation. For every actor, a swim lane is created. Afterwards, the process elements are added and the process model is built sentence by sentence. The approach only supports a basic selection of BPMN symbols. Furthermore, a method is presented, which allows for an automated combination of single process models resulting in a complete business process.

4.4 Process Model Generation from Natural Language Text (Friedrich et al. 2011)

Textual input. Every text can be used as long as it fulfills some requirements. The text has to describe a model, must not contain questions and has to be described sequentially. Furthermore, it should describe the view of an involved actor and non-sequential jumps have to be made explicit.

Text analysis. For the syntactical analysis, tools from the Stanford NLP Group are used to parse a syntax tree with Stanford dependencies. The semantic analysis is based on dictionaries and databases as WordNet and FrameNet and also includes a self-developed anaphora resolution. The authors utilize stop word lists to identify flow elements. As in earlier approaches, they use a variation of the CREWS scenario model as a world model to store the intermediate information.

Model generation. After the creation of swim lanes for the actors, nodes for every activity and event are created. These nodes are connected with their related flow elements. This approach is able to use more BPMN elements thanks to a deeper and more complex text analysis.

4.5 Automatic Process Model Discovery from Textual Methodologies (Viorica Epure et al. 2015)

Textual input. This approach uses archaeological methodologies as textual inputs. Such methodologies describe archaeological processes with natural language texts. The authors only chose methodologies which describe a single process instance to simplify the given task.

Text analysis. In a first step, the text is cleaned and unwanted punctuation and phrases are removed. The Stanford Parser and a combination of Stanford and NLTK tagger are used to generate a syntax tree. Afterwards, transitive verbs are identified as they are the elements most

likely to represent an activity. For this task, the databases WordNet and VerbNet (Kipper et al. 2004) are used. To identify the relationships between activities, domain specific rules, based on patterns of found activities and keywords, are utilized. The text is analyzed sentence by sentence.

Model generation. The only identified elements are activities and relationships, no explicit model is generated. A textual representation is returned (e.g. Start → (Act1 || act2) → Act4 → Stop).

5 Comparative Analysis

The comparative analysis has to be performed on a theoretical basis because there are no implementations available. The theoretical analysis bases on the previous presentation of the approaches. The different aspects *textual input*, *text analysis* and *target document* are compared and, if possible and useful, they will be rated. For lack of implementations, the involved research groups mostly use internal prototypes or refer to software and applications which were no longer accessible or operable. Hence, a practical analysis was not feasible. The insights obtained by the overview of existing approaches will be used to propose some measures and automated techniques to analyze and evaluate future approaches with the corresponding implementations.

5.1 Theoretical Analysis

There are three main aspects which can be analyzed and compared. These are the textual input, the chosen NLP technique and the procedure of deriving a process model. In the following analysis, the different approaches, in order of the given overview, are referenced as Artefacts, GroupStories, UseCases, Alltext and Methodologies.

5.1.1 Textual Input

Artefacts is the only approach which does not introduce any constraints for input texts. This is possible because it is not the goal to extract a complete model but just model parts in no particular order to be combined by an analyst. GroupStories utilizes a moderator during the extraction of knowledge to guarantee a reliable process description. In UseCases, it is not explicated which form a use case is required to have. They state that the relevant parts are extracted manually with all numeral extensions but not whether the order of the sentences is altered. It is to assume that it, like all other approaches, depends on a sequential order of the event descriptions and an explicit indication of jumps. Overall, the text requirements introduced for Alltext generally apply to every approach which ties to generate a correct model out of a model description. These requirements are, besides the sequential order, the non-presence of questions in the process description and the existence of a process level.

5.1.2 NLP techniques

The core of every text-to-model approach is the text analysis with NLP. The amount and quality of extracted information decides on the complexity of the generated models. Besides the ability to extract the information of a given problem, an important factor of NLP techniques is their flexibility. To prove useful as a universal text-to-model approach, the capability to adapt to different domains or even languages is necessary.

As a first step, every approach conducts a syntactic analysis. They use POS taggers to annotate the text. Alltext and Methodologies use the Stanford Tagger, Artefacts the NLTK, GroupStories a

trigram tagger and UseCases a tagger on basis of Robust Risk Minimization (RRM). POS tagging is a well-researched problem. With exception of RRM (92%), all taggers work with a precision of approx. 97%. Methodologies found that a combination of tagging methods could yield to a better result. Therefore, the chosen technique is of minor importance. The POS taggers are available for many languages or can easily be trained for different languages with respective text corpora.

In a second step, the annotated texts are parsed to build a syntax tree. Artefakt, GroupStories and UseCases use shallow parsers. Alltext and Methodologies use, with the Stanford Parser, a deep parsing method. With more complex trees, better results can be achieved. Except for GroupStories, all approaches execute a semantic analysis. Alltext and Methodologies use available databases and dictionaries as WordNet, FrameNet or VerbNet. Artefacts and UseCases rely on self-created domain databases. By creating own databases, they achieve a high flexibility and can easily be adapted to different domains. UseCases and Alltext implemented concepts of anaphora resolution as part of the semantic analysis. This allows the detection of activities which are described in more than one sentence.

With all information available and all elements identified, the produced syntax trees were searched for patterns of verb and noun phrases to identify activities, events and actors. The activities are the central elements of a process model. To identify these activities, verbs and verb phrase and their related noun phrases are used. For the identification of events and sequence flows, word lists which indicate sequential (e.g. then, after), conditional (e.g. if, whereas) or parallel (e.g. while, in the meantime) flows were established. Methodologies uses a domain specific database with all combinations of verbs and signal words to identify model elements. Moreover, GroupStories, UseCases and AllText use a modified meta model based on the CREWS scenario model to store the extracted information.

5.1.3 Model Generation

Methodologies only analyses the text for activities and does not create a BPMN model but a descriptive process representation out of activities (e.g. Start → (Act1 ‖ act2) → Act4 → Stop). Artefacts only creates model parts, which have to be combined by an analyst. For the remaining approaches, the extracted data, stored in a modified CREWS meta-model, is used to build the process models. First, an actor will be represented as swim lane in a BPMN model. Then, events and activities will be included. Alltext creates all nodes first and connects them with sequence flows afterwards. UseCases creates the activities, events and sequence flows in the order of their occurrence in a use case. Because of its advanced text analysis, Alltext is able to use a higher number of BPMN symbols and therefore can create a more complex process model.

AllText is the only approach which refers to current labeling conventions and applies them to label the model elements appropriately. Methodologies mentions the labeling problem but does not stick to the conventions.

5.1.4 Summary

The approaches are different in most of their aspects. Thus, they provide a good overview of current methods and possibilities of text-to-model mining. Table 2 summarizes the previously presented sections. The resulting advantages and disadvantages of the presented approaches are listed in Table 3. The most advanced among the presented approaches is (Friedrich et al. 2011). It introduces many features and ideas, which should serve as standard for every approach.

			Approach				
			Artifacts (Ghose et al. 2007)	Group-Stories (Goncalves et al. 2009)	UseCases (Sinha and Paradkar 2010)	Alltext (Friedrich et al. 2011)	Methodology (Viorica Epure et al. 2015)
Textual Input	Flexi-bility	Speech		●	●		
		Domain			●	●	
	Require-ments	Sequence			●	●	●
		Explicit Jumps				●	
		Keywords	●	●	●	●	
NLP-Techniques	Syntactic Analysis	Stanford-Tagger				●	●
		NLTK-Tagger	●				●
		Trigram-Tagger		●			
		Linear Classific.			●		
		Stanford-Parser				●	●
		NLTK-Parser					
		Shallow- Parser	●	●	●		
		Templates	●				
	Semantic Analysis	WordNet				●	●
		FrameNet				●	
		VerbNet					●
		Individ. Ontologies	●		●		
		Anaphora Res.			●	●	
Model Generation		Structured Desc.			●	●	
		Activities	●	●	●	●	●
		Connectors /	●	●	●	●	●
		Events			●	●	
		Actors	●	●	●	●	

Table 2: Comparative analysis summary

Although some ideas can also be found in earlier approaches, Friedrich combined them to create the most complete approach. As for texts, it introduces only minor constraints to support the text mining. For the text analysis, state-of-the-art applications as the Stanford Parser and databases as WordNet were used. Additionally, an own technique for anaphora resolution was presented, which, overall, achieved a better score than existing anaphora resolution applications as BART (Versley et al. 2008) in this special environment of process models. Due to the thoroughly executed analysis, a more complex model generation is rendered possible. On average, they were able to correctly reconstruct 76% of the models in a test set based on their textual descriptions. Meanwhile, the used NLP techniques are outdated. Usually, an approach should advance with the possibilities of its NLP techniques. Hence, a new evaluation with current techniques is necessary.

	Advantages	**Disadvantages**
Artefacts	use of reference models	creates only model parts
GroupStories	flexible about speech	no semantic analysis no complete BPMN models
UseCases	very flexible about speech and domain allows relative fast computation	no clarity about input text format high effort and domain knowledge necessary
AllText	creates complete models thorough semantic analysis	-
Methodologies	no word lists necessary	creates no models

Table 3: Advantages and disadvantages of the available approaches

5.2 Proposed Practical Empirical Analysis Approaches

The different approaches presented try to build a model from natural language text, hence, the quality of the generated model can be used to rate the performance of an approach. However, it cannot be stated how well an approach performs on a given text. To compare the general performance, a test set has to be used. Such a test set contains text-model pairs. These pairs are validated by experts and analysts. The automatically generated model and the reference model can be compared to measure the performance. A well-designed approach should produce a high quality process model. With further improvements of the developed approaches and after reaching a certain standard for the generated models, the run time of an approach should be as low as possible. Therefore, the time needed to create a model out of a textual resource should be part of every practical analysis. There is a lack of available implementations for the presented approaches and a practical analysis is not feasible. Such an analysis should be the content of future research. For this reason and encouraged by the evaluation of Friedrich et al. (2011), a research group at the DFKI is currently working on an advanced version of this approach. As a part of the RefMod-Miner[4], a text-to-BPMN and a newly adapted text-to-EPC approach are realized. Currently, the applications are in a testing stage. They show some promising results when the textual resources fit certain textual requirements. Overall, there is a low tolerance regarding different text types. In the future, a practical analysis based on the proposed techniques will be performed.

5.2.1 Model Quality

Recent research states the importance of model quality. There are two rather easy to check quality criteria: The syntactic quality, which states whether the model is correctly modeled according to the syntax of the modeling language and the semantic quality, which states how well a model describes the modeled domain. Since the models are generated automatically, the syntactic quality should be guaranteed. The semantic model quality of an approach can be measured with the help of a test set with texts and their corresponding manually generated models. In combination with the model similarity, a metric can be computed serving as a measure for the semantic quality. A third, rather subjective, criterion is the practical quality, which defines whether a model can easily be understood. (Houy et al. 2012) did some research on how to measure the understandability of conceptual models like process models. They found that understanding a model contains different dimensions. The aspects of understanding a model can be subjective or objective. Furthermore, the understanding can depend on the effectiveness or the efficiency of the model. Four more or less measurable understanding criteria were identified: *correctly answering questions about the model, problem solving based on the model content, perceived ease of understanding a model* and *time needed to understand a model*. These criteria cannot be computed automatically, which makes it hard to rate an approach based on how understandable the created model is. But it has been shown that the understanding of process models can be improved by a systematic labeling style. Therefore, an approach should at last follow naming conventions. (Leopold et al. 2013) introduced an approach to automatically detect naming convention violations. Moreover, the model quality has to be seen in reference to the participants of a process and the creators of a process description. Thus, terminology, grammatical structures or abstraction levels can differ. For example, the description and perception of a process by the head of the sales department may significantly differ from the technical specification used by a system administrator.

[4] http://refmod-miner.dfki.de

5.2.2 Model Similarity

If reference models are available, the automatically and manually generated models can be compared to determine the performance of an approach. A reasonable metric for this task is the model similarity, which can be applied for different aspects of a process model (Dijkman et al. 2011), as e.g. based on structural properties, on semantic similarities of the labels or on both aspects. In the area of business process models, there exist various similarity measures which can be used for the practical empirical evaluation. The selection of the similarity measure has to be carried out depending on the content of the automatically generated model. If the model contains, for example, organizational units, the similarity measure should consider these for the similarity calculation. Otherwise, it is not possible to correctly verify the quality of the derived model. The reference model can be seen as an optimal description in form of a process model of the input text. Thus, the calculation of the similarity value between the manually generated reference model and the automatically generated process model provides a numerical value, which can be used to determine the overall quality of the generated model. A higher similarity value means a higher conformance and, thus, a higher quality of the generated model.

6 Discussion

With the presentation and the comparative analysis of current approaches, a starting point for further research was given. The previous analysis was conducted on a theoretical basis as there are not enough implementations available. The research groups are using internal prototypes. The analysis could be much more informative if a proper practical analysis could be executed. Furthermore, with the focus on business process models, only a small area of the topic is covered. Still, it shows the difficulties of further research. Differences in possible textual resources and the different forms of output models require many different approaches. An important factor in future research is the importance of the individual research parts, which have to be combined for text-to-model transformations. Mainly, those are NLP, process models and model mining. Further research and development will benefit from the work at hand and shall guarantee advancement.

A simple analysis was performed by the authors, with an independent text-to-EPC implementation, based on Friedrich et al. (2011), which is considered as the current state-of-the-art. It was not validated with a full test set but small tests already showed some existing problems. The low tolerance towards flawed textual input files is a problem for current and future research. Small differences in the required text format results in a significantly lower model quality, not taking spelling and punctuation errors into account. That could be a reason for which the research on text-to-model transformation was declining. The requirements for textual process descriptions are too high. Further research is required to handle more variations and errors of textual inputs.

7 Conclusion

The goal of this paper was to determine the state-of-the-art of the current research in text-to-model transformation. The major focus was set on approaches which generate business process models. Since 2011, no thorough approach has been published and only one more current but incomplete approach from 2015 could be found. A brief presentation of existing approaches was given and combined with a theoretical comparative analysis. Due to a lack of available implementations, no practical analysis could be performed. Although the lack of a practical

analysis, the approach developed by Friedrich was identified as the current state-of-the-art. Thus, the current research project at the DFKI aims at an advanced implementation of (Friedrich et al. 2011). As part of the research prototype RefMod-Miner, BPMN models and EPCs can be generated. The application shows some of the problems which have to be overcome in the next years. This paper presents a first step towards further research in the field of model mining out of natural language text. In the future, the focus should be set on an increased text tolerance in order to enable the analysis of more complex textual resources and to gain a deeper understanding of them. Another goal should be to obtain more implementations and focus on a more practical analysis of existing approaches. In this way, strengths and weaknesses of approaches could be identified more reliably and the gained information used to develop more advanced methods.

Literature

Bajwa IS, Choudhary MA (2011) From Natural Language Software Specifications to UML Class Models. Paper presented at the 13th International Conference on Enterprise Information Systems

Baker CF, Sato H (2003) The FrameNet data and software. Paper presented at the Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 2, Sapporo, Japan

Bird S, Klein E, Loper E (2009) Natural Language Processing with Python. O'Reilly Media, Inc.,

Blumberg R, Atre S (2003) The problem with unstructured data. RM Review (13):42-49

Boguraev B, Neff M (2008) Navigating through Dense Annotation Spaces. Paper presented at the Sixth International Conference on Language Resources and Evaluation (LREC'08), Marrakech, Morocco

Brants T (2000) TnT: a statistical part-of-speech tagger. Paper presented at the Proceedings of the sixth conference on Applied natural language processing, Seattle, Washington,

Dijkman R, Dumas M, van Dongen B, Käärik R, Mendling J (2011) Similarity of business process models: Metrics and evaluation. Information Systems 36 (2):498-516. doi:Doi 10.1016/J.Is.2010.09.006

Friedrich F, Mendling J, Puhlmann F (2011) Process model generation from natural language text. Paper presented at the Proceedings of the 23rd international conference on Advanced information systems engineering, London, UK

Ghose A, Koliadis G, Chueng A Process Discovery from Model and Text Artefacts. In: Services, 2007 IEEE Congress on, 9-13 July 2007 2007. pp 167-174. doi:10.1109/services.2007.52

Goncalves JC, Santoro FM, Baiao FA (2009) Business process mining from group stories. Paper presented at the Proceedings of the 2009 13th International Conference on Computer Supported Cooperative Work in Design, CSCWD, Santiago

Houy C, Fettke P, Loos P (2012) Understanding Understandability of Conceptual Models – What Are We Actually Talking about? In: Atzeni P, Cheung D, Ram S (eds) Conceptual Modeling, 31st International Conference ER 2012 (LNCS 7532). Lecture Notes on Computer Science, vol 7532. Springer, Berlin, Heidelberg, pp 64-77

Kipper K, Snyder B, Palmer M (2004) Extending a Verb-lexicon Using a Semantically Annotated Corpus. Paper presented at the LREC

Leopold H (2013) Natural Language in Business Process Models. Lecture Notes in Business Information Processing (LNBIP), vol 168. Springerr, Berlin

Leopold H, Eid-Sabbaghb R-H, Mendling J, Guerreiro Azevedod L, Araujo Baiãod F (2013) Detection of naming convention violations in process models for different languages. Decision Support Systems 56 (1):310-325

Maiden NAM (1998) CREWS-SAVRE: Scenarios for Acquiring and Validating Requirements. In: Sutcliffe A, Benyon D (eds) Domain Modelling for Interactive Systems Design. Springer US, pp 39-66. doi:10.1007/978-1-4615-5613-8_3

Manning CD (2011) Part-of-speech tagging from 97% to 100%: is it time for some linguistics? Paper presented at the Proceedings of the 12th international conference on Computational linguistics and intelligent text processing - Volume Part I, Tokyo, Japan

Miller GA (1995) WordNet: a lexical database for English. Communications of the ACM 38 (11):39-41

Omar N, Hassan R, Arshad H, Sahran S (2008) Automation of database design through semantic analysis. Paper presented at the Proceedings of the 7th WSEAS international conference on Computational intelligence, man-machine systems and cybernetics, Cairo, Egypt,

Santoro FM, Borges M, Pino JA Tell us your process: A group storytelling approach to cooperative process modeling. In: Computer Supported Cooperative Work in Design, 2008. CSCWD 2008. 12th International Conference on, 16-18 April 2008 2008. pp 29-34. doi:10.1109/cscwd.2008.4536950

Sinha A, Paradkar A (2010) Use Cases to Process Specifications in Business Process Modeling Notation. Paper presented at the 2010 IEEE International Conference on Web Services, Miami, Florida

van der Aa H, Leopold H, Mannhardt F, Reijers H (2015) On the Fragmentation of Process Information: Challenges, Solutions, and Outlook. In: Gaaloul K, Schmidt R, Nurcan S, Guerreiro S, Ma Q (eds) Enterprise, Business-Process and Information Systems Modeling, vol 214. Lecture Notes in Business Information Processing. Springer International Publishing, pp 3-18. doi:10.1007/978-3-319-19237-6_1

Versley Y, Ponzetto SP, Poesio M, Eidelman V, Jern A, Smith J, Yang X, Moschitti A (2008) BART: a modular toolkit for coreference resolution. Paper presented at the Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Demo Session, Columbus, Ohio

Viorica Epure E, Martin-Rodilla P, Hug C, Deneckere R, Salinesi C Automatic process model discovery from textual methodologies. In: Research Challenges in Information Science (RCIS), 2015 IEEE 9th International Conference on, 13-15 May 2015 2015. pp 19-30. doi:10.1109/rcis.2015.7128860

Zhang T, Damerau F, Johnson D (2002) Text chunking based on a generalization of winnow. J Mach Learn Res 2:615-637