Extracting process graphs from medical text data

An approach towards a systematic framework to extract and mine medical sequential processes descriptions from large text sources.

Andreas Niekler, Christian Kahmann

University of Leipzig, Department of computer sciences, Natural Language Group,
Augustusplatz 10, 04109 Leipzig,
(aniekler|kahmann)@informatik.uni-leipzig.de

Abstract. In this paper a natural language processing workflow to extract sequential activities from large collections of medical text documents is developed. A graph-based data structure is introduced to merge extracted sequences which contain similar activities in order to build a global graph on procedures which are described in documents on similar topics or tasks. The method describes an information extraction process which will, in the future, enrich or create knowledge bases for process models or activity sequences for the medical domain.

Keywords: relation extraction, natural language processing, graph processing, process models

1 Introduction

Medical publications, surgical procedure reports or medical records typically contain procedural descriptions. For example, all activities included in a medical study must be documented for reproducibility purposes, in surgical reports a stepwise description of included procedures is documented and in medical records a history of medical treatment is listed. Additionally, related studies or reports describe alike activities with some alterations or rely on preceding activities that may be described in other documents. This kind of knowledge can be contained in large document collections like the PubMed Dataset.[1] For example, the preparation steps before DNA could be sequenced are often the same but need to be documented for each study. Such redundant activity descriptions can be found amongst many documents describing research within the same domain or field of research. Nevertheless, differences amongst the activities in related documents also exist. A complete overview of activities from a defined document collection provides an easy insight to workflows and paradigms within a domain or field of study. For example, consider the following text snippets extracted from three different documents within the PubMed Dataset.

[1] http://www.ncbi.nlm.nih.gov/pubmed/

- aCL and B2GP-I autoantibodies were evaluated at baseline and at 3 and 6 months after the beginning of infliximab treatment. Statistical analysis was performed using Statistica 7.0 PL software. Differences between groups were analyzed using Mann-Whitney U test. A p value less than 0.05 was considered to be statistically significant. We observed 4 aCL IgM-positive (12.5%) patients before the beginning of infliximab treatment.
- The statistical analysis was performed using the SPSS 11$\dot{0}$ package program. Differences between groups were analyzed using the Mann-Whitney U test. Correlation analyzes were performed using Pearson's correlation test.
- In the event of discordant scores, which differed by a maximum of 1 point, the mean of the two scores was used. Because the semiquantitative data are nonparametric, these data are presented as median (range). Differences between groups were analyzed using the nonparametric Mann–Whitney U-test. For markers scored as present or absent, the χ^2 test was used.

As one can see from the examples, all documents contain the application of the *Mann–Whitney U-test*, even though it is expressed slightly different in the texts. This repeatedly used activity is concurrently used with other activities within the documents. Thus, finding this link between the documents and aligning the activities w.r.t. redundant activities helps to structure and analyze procedural knowledge from topical- or domain-related medical texts. For example, early stages or parts of a larger process might be documented separately to other parts or later stages. In order to extract complete and connected descriptions of such procedural knowledge it is a promising approach to utilize links between different documents and connect the extracted knowledge accordingly.

In this paper a general natural language processing workflow to extract sequential activities from large collections of medical text documents is developed. A graph-based data structure is introduced to merge extracted sequences which contain similar activities in order to build a global graph on procedures which are described in documents on similar topics or tasks. After the review of related work in section 2 the paper introduces the approach in section 3. To demonstrate the potential of such a general workflow we introduce a working example and possible applications on the basis of a subset of the PubMed dataset in section 3.

2 Related work

The extraction of procedural knowledge from text documents has been investigated for different domains. For example, [3], [10] and [1] describe the process and activity extraction from text as natural language processing (NLP) pipeline. They apply a static rule set on the available features from the results of the NLP pipeline in order to construct the procedural models. In general, the described NLP pipelines use sentence separation, tokenization, POS-tagging and a sentence parser. Other techniques for named-entity-or multi-word-unit-detection are also mandatory for this task. [10] and [1] apply their methodology to the cooking recipe domain and extract procedural models from single recipe descriptions whereas [3] applies the techniques to different domains with promising results. Additionally, anaphora resolution is also applied in order to match a result of an activity, e.g. the combination of different components like ingredients to prepare a "sauce", to later occurrences of that result in the text which might be references with a different token. Other works try to model processes and activities from tutorial instructions given in natural language or utilize use-case descriptions from requirement specifications [4, 11]. In those cases the approach concentrates on a domain and the process description is limited to a fixed set of possible activities.

The creation of probabilitic graphical models using multiple medical records has been investigated in [5]. In this work the authors extract medical problems, tests and treatments from Electronic Medical Records. The extracted information is encoded within a graph structure where the associations between the different types of entities are modeled with co-occurrence statistics. The result of this process is transferred into a probabilistic graphical model which can be used to infer most likely treatments and tests for a medical problem. This work is highly related to the methods described in the paper presented here. However, there are differences in the addressed requirements and properties of the data. The work of [5] builds on the fact that different diseases and their according treatment and testing strategies are contained redundantly in the records. This allows to extract co-occurrence statistics among the mutually used medical concepts in different medical records to determine the strength of their association. The approach is focused and tailored to the domain of medical records and addresses the properties of this text source. The workflow described in our paper yields a general approach to the problem of procedural knowledge extraction for different domains. Thus, the co-occurence information of mutual used activities can be very sparse and the chaining and linking of the extracted entities and concepts are addressed in a different way.

With the exception of [5] all examples create process models from single documents. The combination of knowledge and process descriptions from multiple documents is rarely studied. The proposed method in our paper concentrates on the integration of multiple dependent activity sequences found within a domain or text collection. Thereby, we do not fit our methodology to the properties of a text source or domain and mainly use uninformed approaches. The objective of our methodology is the general extraction of global activity sequences from text sources relating to a domain, work field or task of choice.

3 Text Mining methodology for process extraction

In this section we describe a methodology which extracts and links activities from medical text documents. The described system follows a sequence of procedures in order to create an activity graph as a result. First, the text sources have to be processed in order to access the entity items in the text. Different entities in a sentence are related and form an expressed activity. Therefore, the extraction of valid relations that form activities is introduced to the text processing step. The expression of the activities could vary throughout different text collections. To adopt to such properties we describe an active learning process with a Support Vector Machine classification. This approach supports a semi-automated and fast creation of training examples for the classification task of relations in our proposed methodology. The detected activities within single documents can be represented as vertices in a directed graph. This representation is based on the fact that the data structure must reflect temporal relations among the activities such as sequences. Thus, the second step in our proposed methodology is the creation of a directed graph structure which can be further used for the representation of the activities contained within a text collection. In the following section the text processing and the graph creation are discussed in detail.

3.1 Text processing and classification for activity extraction

The text sources must be separated into sentences and tokens first by using state of the art tools.[2] Additionally, POS-Tagging was applied to the text sources. To extract the procedural knowledge from the texts, named entity recognition (NER) is required as a pre-processing step. Many NER-algorithms for different purposes have been studied. The state of the art ranges from conditional random field classifiers to ensemble learners which combine multiple entity detection algorithms [2, 7]. It would be possible to use 3rd party named-entity detection tools in order to annotate entities automatically but the quality depends on the text source in combination with the algorithm. Since this paper describes a mechanism for using annotated entities to extract activities from text documents it isn't the main focus to vote for a single NER-solution. For simplicity and understandability the experiments in this paper were implemented using a standard pattern-based entity detection to put explanations about the decisions for a specific NER-solution aside. A typical pattern for the detection of entities in the medical domain is (adjective* noun+) which identifies all nouns as entities and, in addition, identifies multi-word-units which consist of a sequence of adjectives followed by a sequence of nouns.[3]

In the separated and preprocessed sentences multiple entities may form an activity. Consider the sentence "Real-time_JJ PCR_NNP was_VBD done_VBN using_VBG the_DT fluorescent-labelled_JJ oligonucleotide_NN probes_NNS". Following the pattern for entity extraction given in the above section the entities "Real-time PCR" and "fluorescent-labelled oligonucleotide probes" are extracted from the sentence. The two entities form the activity "done" which can be part of a chain of activities document throughout multiple documents.

The characteristics of activities or relations between entities change within different domains or described procedures. Thus, the process for identifying and

[2] OpenNLP was used to process the text sources for this paper. `http://opennlp.apache.org/`

[3] The "*" implies a minimum occurrence of 0 and an unbounded maximum occurrency. The "+" implies a minimum occurrence of 1 and an unbounded maximum occurrence.

connecting entities to activities within the sentences should not be fixed or static. To answer this fact the identification of relations or activities is defined as classification task using a Support Vector Machine (SVM) along with word- and POS-Tag-level features [6]. If a sentence contains an entity E_1 and E_2 the two words before E_1, the two words after E_2 and all words between E_1 and E_2 are extracted as features. Furthermore the POS-tags of the extracted words are used as features for the SVM. To name the features the extracted words are prefixed with a feature name. For example, if one word between E_1 and E_2 is "using_VBG" the word gets the prefix "BETWEEN_" and will become the feature "BETWEEN_using". The same procedure is applied with the POS-Tag of this word to form the feature "BETWEEN_VBG". The feature set for each relation in the training data is joined into an example-feature-matrix in order to train the SVM.

Before the training process is applied the user must define the type and the form of the desired relation. On the basis of this definition training examples are collected from the data. For this purpose an active learning procedure is introduced where the user iteratively collects training data with the support of an automatic classification. An initial search for sentences that include a minimum of entities and verbs that indicate an activity is conducted.[4] The search is implemented using a customizable pattern which may be altered w.r.t. different domains and relation types. The set of matching sentences which contain this custom pattern is presented to the user. Correct entities are selected from the proposed sentences along with the definition whether there is a relation between them or not. The features are extracted automatically and the set of positive and negative examples is used to train an initial SVM model. The trained model is used to identify additional examples in the data. The user judges on those examples and with every batch of new examples the classifier can be refined.[5] If the training quality of the SVM does not change with new examples a final model is trained and applied to all documents. The result is a set of sentences from a document collection where each sentence contains an activity or valid relation between entities.

3.2 Process graphs for activity representation and processing

In the next processing step a data structure is constructed on the basis of the set of activities that were identified by the classification process. For each activity the two entities E_1 and E_2, the Verb V (past participle between them, a document

[4] A basic pattern for this purpose is given by the pseudo-pattern (adjective* noun+) ... (using_VBG) ... (adjective* noun+) where the "..." indicate optional words that may be contained between E_1 and E_2

[5] To implement an active learning process one must simply present positive and negative classification results to the user. After the judgment, the training set can be refined and extended. A new model can be trained for the next iteration of the active learning procedure and new examples can be classified and presented to a user. The presentation and feedback mechanism can be implemented using a graphical user interface or simple command line interactions.

identifier and a sentence identifier are stored[6]. A graph structure A, a directed graph, is introduced where all identified activities are represented as vertices. All vertices that build a sequence of activities within a document are connected with directed edges, e.g. consecutive activities will be connected as a chain of activities within the graph structure. For example, consider the following sentences.

– Pathological_JJ diagnosis_NN of_IN patients_NNS with_IN atherosclerosis-RNA_NNP extraction_NN from_IN biopsies_NNS was_VBD done_VBN by_IN the_DT Qiagen_NNP Kit_NNP protocol_NN ._.
– RNA_NNP was_VBD cleaned_VBN from_IN DNA_NNP contamination_NN using_VBG DNAse_NNP Qiagen_NNP ._.
– Reverse_VB transcription_NN was_VBD done_VBN using_VBG Promega's_NNP reverse_VB transcriptase_NN M-MLV_NNP protocol_NN ._.
– Real-time_JJ PCR_NNP was_VBD done_VBN using_VBG the_DT fluorescent-labelled_JJ oligonucleotide_NN probes_NNS ._.
– Reaction_NN was_VBD done_VBN using_VBG the_DT chemical_NN supplies_NNS manufactured_VBN by_IN the_DT company_NN Eurogene_NNP ._.

Those examples can be seen as a sequence of activity descriptions from one document and will be connected as a sequence using directed edges between subsequent relations, e.g. vertices in the graph. This procedure creates a chain of connected vertices for every document in A. The main target for the further processing of A is the linking of different activity chains from multiple documents. This will produce a graph structure which represents networks of activities that supplement each other. In A the connected components can be understood as a summary of activities which come from, or lead to, similar activities. For example, multiple surgical reports contain many redundant descriptions for a certain type of surgical procedure. In some cases there might have been complications and the surgeon had to react on those. Those complications are included in a report between two relations R_a and R_b which might be subsequent in other documents describing the same procedure without complications. A graph which merges different sequential activities from different documents should introduce a direct edge and a cycle of activities between R_a and R_b describing the additional complications. In a later review of the graph this cycle represents single differences from the analyzed standard procedure.

To detect similar relations throughout different documents a similarity operation $sim(R_{D_1}, R_{D_2})$ is defined. This similarity operation can be constructed on the basis of word level similarity or semantic similarity. With a preprocessing of the corpus like word2vec or a co-occurrence analysis each of the relation components can be augmented by semantic vectors representing the associated vocabulary, e.g. the semantic embedding [**?**,**?**]. This allows to compare entities semantically and conceptual similarities between entities can be used to find

[6] The verb V could also be seen as the modifier or the name of the activity and could be replaced in other tasks. The usage of V (past participle) works for the examples in this paper and can be different in other domains

	a	b	c	d	e
a	0	1	0	0	0
b	0	0	1	0	0
c	0	0	0	0	0
d	0	0	0	0	1
e	0	0	0	0	0

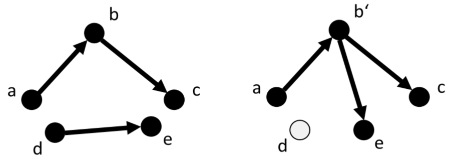
a:

	a	b	c	d	e
a	0	0	0	0	0
b	0	0	0	1	0
c	0	0	0	0	0
d	0	1	0	0	0
e	0	0	0	0	0

b:

	a	b	c	e
a	0	1	0	0
b’	0	0	1	1
c	0	0	0	0
e	0	0	0	0

c:



d:

Fig. 1: Collapsing and merging vertices based on similarity information in S with a:) an Example of the adjeceny matrix of A for document based activity sequences, b:) adjacency matrix S between similar relations and c:) the resulting matrix A' after the collapsing of A. In d:) the merging and edge transfer between two vertices is displayed.

alike relations. Note, that the similarity function is another exchangable component of the information extraction approach described in this paper. It can be altered for differnt sources or domains in order to achieve an optimal quality. For simplicity and to concentrate on the graph processing methodology a Jaccard similarity based on character 3-grams is used as similarity function for the examples in this paper. The similarity between all activities is calculated for E_1, E_2 and Vseparately which results in three different similarity matrices. In the data, similar entities can consist of multiple words and some additional abbreviations in parenthesis which introduces some slight differences amongst them. The usage of character 3-grams is robust for such little variations. The resulting three similarity matrices are transformed to adjacency matrices by applying a threshold to the values. All similarity values that exceed the threshold will be set to 1, the indicator for an edge between two relations. Values beneath the threshold will be set to 0 to indicate no similarity between two relations. It is also imaginable to set three different thresholds for each single similarity matrix or to weight the matrices for further processing. All resulting adjacency matrices are multiplicated element-wise in order to create a single adjacency matrix S of similar activities, e.g. two activities where E_1, E_2 and V are similar between the two activities are represented by the value of 1 in the final matrix.

In the following step the activities considered to be similar are collapsed using the adjacency matrix S resulting in a graph A'. This process is sketched in figure 1. Starting from graph A all edges from similar vertices are taken over to a single vertex and all vertices where the edges where taken from are deleted. That means similar vertices are collapsed to a single vertice and the associated ingoing and outgoing edges of those relations are merged. The resulting graph connects the sequences of different documents where similar activities build single vertices with more than one incoming or outgoing edge ($d^+_G(v) > 1$ or $d^-_G(v) > 1$). Activities with this property are identified more frequently than other activities in the data and thus are of some importance for the overall activity summarization. In summary, it can be said, A' is an unconnected graph where a set N of connected components can be identified. This set represents different graphs where the interaction and coherence of related processes, described in different documents, is encoded.

4 Applications and Examples

The resulting graph can be exploited for different applications. In the following examples three possible applications of exploratory data analysis are discussed. All examples are created on the basis of data from the PubMed dataset which was additionally reduced to a subcorpus consisting of 2.813 documents. The documents all contain the keyword phrase "Ankylosing spondylitis"[7], which represents an autoimmune disease of the axial skeleton.

4.1 Summarization of activities as process graphs

In the first example the method is used to extract sequences of activities from studies in a specific domain. The corpus was separated into sentences and word-tokens. Additionally, POS-tagging was applied with the PENN Tagset.[8] The processing is started by looking for sentences containing the pattern (adjective* noun+) ... (was,were,has,been,had)_VBD) ... (using_VBG) ... (adjective* noun+). This pattern can be seen as a user defined constraint which could be altered for other text sources or domains. In this case the pattern reflects stereotype sentences from the corpus which describe an activity that has been carried out by the authors of the underlying medical paper. The user decides which of the matching sentences suit the defined or required description of an activity. All validated examples are passed to an initial training set for the SVM classifier described in section 3.1. After this step an active learning process is applied and the training set is extended semi-automatically. The overall process, including all generated training examples, identifies 14.087 relations from the corpus which will be further processed. The next step links similar relations from the classification result as described in section 3.2. For this example the threshold for the Jaccard similarity of R_{D_1} and R_{D_2} is set to 0.5. Afterwards, the graph adjacency matrix of A is created and the edges for the inner-document connections of the relations are inserted. The similarity matrices for all E_1, E_2 and V are multiplied element-wise to find similar relations and A is collapsed by the similarity information to produce the matrix A'. The resulting adjacency matrix is converted

[7] http://en.wikipedia.org/wiki/Ankylosing_spondylitis

[8] http://www.comp.leeds.ac.uk/ccalas/tagsets/upenn.html

to a graph. The graph contains a set of connected components. Each connected component can be seen as a single graph which represents a separated summary of activities. The single connected components can be visualized and explored separately by an analyst. To filter for activity summaries containing prominent relations the components are only kept if they contain at least one vertice which is based on a relation that was found more than 3 times in the data. Furthermore all components which do not contain relations from documents with a given set of keywords included are filtered out. This additional procedure allows to drill down the analysis to a user defined focus. For this example the keywords "gene" and "tissue" where used to filter out graph components drawn from documents not containing those words. The initial graph A of the given example consists of 14.987 vertices and 23.902 edges. The graph contains 1385 connected components with a minimum of 1 edge. The median diameter among all connected components is 6. The final graph A' is reduced to 10.453 vertices and 9.234 edges. This processed version of the graph contains 1.063 connected components. The median diameter among all connected components in A' is 10. As one can see, the diameter of the connected components rises and the procedural knowledge among different documents is linked within the final graph.

The resulting graphs can be visualized and analyzed. In this experiment Gephi is used for visualization purposes [8]. Within Gephi a graph could be further processed and explored. For example, a user can filter the graph for vertices that have a certain degree on incoming and outgoing edges. The final visualizations are very useful to summarize and understand the activities which are normally hidden within large document collections. In figure 2 an example of a visualized graph structure is given. It can be seen that different activities can produce data which is undergoing a statistical analysis.

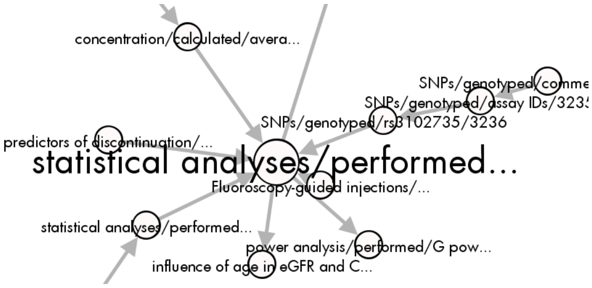

concentration/calculated/avera...
SNPs/genotyped/comme
SNPs/genotyped/assay IDs/3235
SNPs/genotyped/rs3102735/3236
predictors of discontinuation/...
statistical analyses/performed...
Fluoroscopy-guided injections/...
statistical analyses/performed...
power analysis/performed/G pow...
influence of age in eGFR and C...

Fig. 2: A graph visualization centered on the central activity "statistical analyses/performed/software".

4.2 Summarization of activities as lists

On the foundation of the graph A' a summarization of the activities as sorted lists can be extracted. The basic problem in producing a global summary of the activities found in the documents is that fact, that their global position in the whole process is unknown. The only known fact is the relative position w.r.t. direct neighbors in the graph. Those neighbors are normally the preceding or following activities from one document. The graph structure can be used to correct or set a global positioning index for each activity in the following way.

1. First, a sequence of all shortest paths (SP) within a single connected component is built. This process is repeated for all connected components in A'.
2. For each set of SP's an iteration from the longest to the shortest SP is conducted.
3. For one SP the process follows the direction of the edges, starting from the global positioning index of the first vertex, which might be the sentence number from the source document of the underlying relation. All subsequent vertices are forced to have a larger position index than their preceding vertex in the current SP.

Some SP's are overlayed and contain identical vertices. Thus, a vertex can also be included in other SP's. The redundant correction of vertices which are contained in different SP's would lead to a violation of the ascending positioning within a path. Therefore, all possible corrected positions for such a redundant vertex are stored. Remember, that those redundant vertices come from activities which were found several times in the data. The activities finally can be sorted by their global position. The position for activities with multiple position values is averaged. In table 1 a possible result is sketched. Such a view allows to review different phases of activities in complex processes which were reported within a document collection.

4.3 Information Retrieval within process graphs

The graph structure is also useful for querying information. To query the connected graph components all vertices containing a given keyword are accessed and preceding and following vertices are extracted. The query for "gene" results in a set of vertices containing the information given in table 2. Of course it is imaginable to select preceding or following vertices which are more than one vertex away from the matching activities. This application allows for the detailed review of activities linked to a user defined concept. The given concept could be a certain technology or method. In turn, the graph could be mined for an activity like "tissue cells" and the prequisite methods and technologies and products of the activity are observable.

4.4 Future work

This paper describes an idea of an information extraction process which creates global activity descriptions from many text documents. A relation-extraction and relation-connecting workflow based on text mining methods is presented

Name of activity	*avg. position*
significant differences/determined/analysis of variance	464,5
MNCs (nuclei/counted/light microscopy	464,5
Statistical analyses/performed/software R	467,7
results/tested/Wilcoxon test	470
P values/presented/Altman and Bland	471,3
Inter-group differences/evaluated/Mann-Whitney U test	474,5
genesets/identified/test	475
Laser Capture Microdissection/carried/ Zeiss/PALM Microbeam Instrument	477
Figures/plotted/PoseView	479,5
protein concentration/analyzed/Bradford assay reagent	479,5
statistical analysis/performed/Prism	480
Statistical analyses/performed/SPSS V	480
analyses/performed/Stata	480,2
analyses/performed/Statistical Package for Social S...	480,3
Statistic analyses/conducted/SPSS	481
Analyses/done/SAS software	482,7
analyses/performed/STATA	484
Statistical analyses/performed/SAS	485,5

Table 1: A graph corrected sequence of activities. Activities found several times in the data are printed in bold and their positions are averaged.

Incoming
data/genotyped/different platforms
concentration of genomic DNA/measured/ng/
Supernatants/analyzed/eBioscience
quality controlGenomic DNA/extracted/Puregene DNA Isolation Kit (Gentra Systems , Minneapolis , MN , USA)
PBMCs/counted/CASY cell counter (Roche)
hit
major histocompatibility complex region/genotyped/Illumina Infinium 15K array
samples/genotyped/ImmunoChi
healthy donors/genotyped/Applied Biosystems TaqMan SNP
individual/genotyped/Affymetrix Genome-Wide Human SNP Array
controls/genotyped/Illumina HumanCNV370-duo chip
outgoing
Genotype calls/made/BRNNP algorithm
Power calculations/carried/Genetic Power Calculator
analysis of intensity clusters and genotype calls/performed/Illumina Genome Studio software
fine mapping linkage study , allele frequencies/estimated/MENDEL software
RNA levels/quantified/Illumina HT-12 V3.0 platform

Table 2: Example result for activities including the word "gene". Additionally the incoming and outgoing activities are shown.

and the experiments show promising results for practical applications which need to be optimized and evaluated in quality and accuracy. The potential of user refined learning classifiers for relation classification is highlighted in order to be domain and text source independent. For the merging of activities a very simple similarity function is used for this paper and the accuracy is not optimal. Nevertheless, it is possible to show the potential for useful applications based on the described information extraction process for relations.

In order to quantitatively judge on the quality of the extraction process an evaluation dataset and evaluation strategy needs to be developed as prequisite for future work. More research on suitable similarity functions for relations which can also handle semantic similarities will optimize the quality of the graph merging process. Future work will also include the adoption of domain knowledge from knowledge bases. Those has been described as very helpful resources in order to adopt to a domain in [9]. The links and dependencies between entities and their possible representations in the data can be encoded in those data structures by domain experts. This will add supervision and control to the graph creation process and thus allows for a higher precision of the graph. Additionally, anaphora resolution can be modeled with knowledge bases to connect graph structures where the relations represent processes which produce other entities as results. Such edges can't be established with character or semantic comparison of the relations. In the moment a connection can only be established if the producing process is encoded within a single document. Furthermore, the introduction of manual corrections steps to refine the graph and the improvement of the quality and the transferability of the relation extraction classification may also be promising elements to optimize the quality.

Acknowledgement

ExB Labs GmbH kindly helped to compile and preprocess the corpus for this paper.

Funding: The project is funded by European Regional Development Fund (ERDF/EFRE) and the European Social Fund (ESF).

Europäische Union
Europäischer Fonds für regionale Entwicklung
Europäischer Sozialfonds

Europa fördert Sachsen.
EFRE
ESF

References

1. Valmi Dufour-Lussier, Florence Le Ber, Jean Lieber, and Emmanuel Nauer. Automatic case acquisition from texts for process-oriented case-based reasoning. *Information Systems*, 40:153–167, 2014.

2. Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics, 2005.
3. Fabian Friedrich, Jan Mendling, and Frank Puhlmann. Process Model Generation from Natural Language Text. In *Advanced Information Systems Engineering*, volume 6741, pages 482–496. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
4. Yolanda Gil, Varun Ratnakar, and Christian Frtiz. TellMe: learning procedures from tutorial instruction. page 227. ACM Press, 2011.
5. Travis Goodwin and Sanda Harabagiu. Clinical data-driven probabilistic graph processing. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may 2014. European Language Resources Association (ELRA).
6. Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 427–434, 2005.
7. C. Hänig, S. Bordag, and S. Thomas. Modular classifier ensemble architecture for named entity recognition on low resource systems. In *Proceedings of the KONVENS GermEval Shared Task on Named Entity Recognition*, pages 113–116, Hildesheim, Germany, 2014.
8. Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An Open Source Software for Exploring and Manipulating Networks. 2009.
9. Kirk Roberts and Sanda M Harabagiu. A flexible framework for deriving assertions from electronic medical records. *Journal of the American Medical Informatics Association*, 18(5):568–573, 2011.
10. Pol Schumacher, Mirjam Minor, Kirstin Walter, and Ralph Bergmann. Extraction of procedural knowledge from the web: a comparison of two workflow extraction approaches. page 739. ACM Press, 2012.
11. Tao Yue, Lionel C. Briand, and Yvan Labiche. An Automated Approach to Transform Use Cases into Activity Diagrams. In *Modelling Foundations and Applications*, volume 6138, pages 337–353. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.