Term Extraction from Medical Documents Using Word Embeddings

Matthias Bay
MINDS-Medical GmbH
Frankfurt, Germany
bay@minds-medical.de

Daniel Bruneß
KITE - Kompetenzzentrum für Informationstechnologie
Technische Hochschule Mittelhessen
Friedberg, Germany
daniel.bruness@kite.thm.de

Miriam Herold
Department of Business Informatics
Goethe University
Frankfurt, Germany
herold@cs.uni-frankfurt.de

Christian Schulze
Department MND
Technische Hochschule Mittelhessen
Friedberg, Germany
christian.schulze@mnd.thm.de

Michael Guckert
Department MND
Technische Hochschule Mittelhessen
Friedberg, Germany
michael.guckert@mnd.thm.de

Mirjam Minor
Department of Business Informatics
Goethe University
Frankfurt, Germany
minor@cs.uni-frankfurt.de

***Abstract*—In this paper we present a new method for the extraction of discipline-specific terms from medical documents. Due to the small text corpora and the specific nature of medical documents, there are limitations for approaches that are solely based on term frequencies. A combination of such methods with procedures that are sensitive to semantic aspects is therefore promising. We use word embeddings in a neighborhood context based method which we call Snowball because of its layerwise way of working. Snowball is integrated together with established methods into an end to end pipeline with which we can process documents to extract relevant terms. Proof of concept is given on a gold standard created recently together with experts in medical coding. The preliminary results highlight the feasibility of our approach and its potential for automated, machine learning based text processing in the medical context.**

***Index Terms*—Machine learning, natural language processing, text mining, term extraction, machine learning applications**

I. Introduction and Motivation

Medical specialist language has been subject of several research efforts. Medical ontologies such as SNOMED CT [1] or MeSH [2] are well-known, valuable knowledge sources to support the search for medical publications. Medical terms from the ontology can be used as search keys in repositories of medical publications, such as PubMed [3]. The nature of patient-related documents, such as medical reports and referral letters differs from medical publications in terms of scope, medical language, and linguistic expression. It has been observed that each hospital, insurance company, and further health organization has an own body of terms for medical documentation which is task- and discipline-specific to a very high degree. Moreover, the terminology from medical ontologies has a low rate of occurrence in patient-related documents. In addition to medical terms, paraphrasing terms from everyday language are prevalent. Both types of terms, general and specialist terms, are relevant to retrieve, summarize, or classify such documents. For instance in an accident report of an emergency hospital (cmp. Fig. 3), `17-year-old`, `boy` and `during inline skating` might be as important as the medical term `clavicle`. In the medical history document for an insurance, the term `collarbone` might be used instead of `clavicle`. There is a research gap to create a disciplinary-specific body of terms that comprises the relevant terms from both, medical and everyday language to characterize patient-related documents.

The paper proposes an automated procedure for term extraction which is capable to learn from text corpora of limited size and generality. In brief, the procedure comprises two steps. First, word embeddings are learned following the well-established, hierarchical softmax learning algorithm resulting in a continuous-bag-of-words (CBOW) model [4]. Second, only those terms from the CBOW model are retained in the body of terms that are in close neighborhood to a seeding set of medical terms. A novel *Snowball* method is introduced for this step where the cosine similarity for the word embeddings determines the neighborhood of terms. The degree of neighborhood induces the layers of a snowball of relevant terms with the medical terms in the core. The approach has been developed and evaluated in the TLDia project[1] in the application domain of *medical coding* in German language, i.e. the annotation of patient-related documents with diseases and health problems. Today, medical coding is still a tedious task which is mainly performed by medical documentation staff. Term extraction is a first step towards the long-term goal of automated medical coding. Coding can be regarded a classification problem for documents based on terms. Relevant terms from a document corpus serve as the input space for the classification algorithm. Organizing the terms in an ontology in future will allow to augment the classification method with semantics. Further, it will provide a means for transfer

This work was funded by the Federal State of Hesse under the LOEWE-HA project no. 701/19-21.

[1] https://tinygu.de/TLDia

learning, i.e. to reuse and adapt the learned body of terms in related contexts [5].

The paper is organized as follows: related work is discussed in Section II. The extraction method is presented in Section III and evaluated by an experiment in Section IV. Section V provides a conclusion and a brief discussion of future work.

II. Related Work

Term extraction, also called automatic term recognition (ATR), has been studied since the nineties [6] and is still far from being solved [7]. In contrast to our work, many ATR approaches consider *termhood* as a major criterion for term relevance. Termhood is defined as *the degree that a linguistic unit is related to domain-specific concepts* [6]. Termhood includes that a term appears relatively more frequent in that domain than in others [8]. Our notion of term relevance goes beyond termhood since everyday language plays an important role in addition to medical terms. We do not distinguish between general and specialist language and denote this by discipline-specific language.

In the application field of information extraction in patient-related documents, Apache cTAKES [9] is a popular tool applying rule-based ontology mapping of POS tagged noun phrases. Using dictionary lookups, UMLS [10] concepts are extracted and mapped onto ontologies such as ICD or SNOMED-CT. Generally, this results in relatively high recall values at the cost of a low precision, since all potential concepts are considered relevant [11]. Authors of [12] employed cTAKES to evaluate concept extraction of translated clinical notes using a German OpenNLP model and the German UMLS database. They were confronted with comparably worse recall values due to the less extensive size of the German UMLS database. These approaches depend on the content of databases, whilst our approach is more flexible in changing the seeding set. cTAKES' is utilized by [13] for concept extraction in an automated clinical coding task. Extracted concepts are mapped to either SNOMED-CT or ICD-9 ontology to later serve as features to augment text data in a binary classification task. Common supervised machine learning approaches used in [11], [13], [14] are not applicable to our case due to the lack of labeled training data and the small corpus size. Another dictionary-based approach is used by [15] who take several dictionaries for term lookup and a regular expression based rule engine to extract biological terms such as protein names. A *approximate dictionary lookup* technique was implemented by [16], calculating the importance of a component in a multi-word expression to handle the shortcomings of a direct dictionary lookup. In our work, we try to mitigate the effect of linguistic variants of concepts by applying specialized pre-processing for the German language and including a terms neighborhood using a *word2vec* model.

III. Term extraction method

The goal of the term extraction method is to learn a body of relevant terms from a text corpus called source corpus and from a dictionary of technical terms to reflect the discipline-specific language. Since medical notes tend to be noisy the text corpus is prepared in a pre-processing procedure first before the actual term extraction is performed.

A. Pre-processing

Medical documents are full of short sentences, acronyms, spelling mistakes and measures [17]. Additionally, compound words consisting of two or more words are common in the German language and increase vocabulary size [18]. Multiple pre-processing steps are required to address these obstacles. Fig. 1 depicts our pipeline. A given text is tokenized using a slightly modified version of the well-established *NLTK* [19] tokenizer (*language* parameter set to 'german') that preserves sentence integrity on abbreviations. Stop words (based on the *NLTK* German list), special characters, and numerical values are removed. We then use the *RNNTagger* [20] for Part-of-Speech tagging and token lemmatization. Abbreviation detection and disambiguation are based on a dictionary lookup combined with context similarity measures of resolved acronyms using the German *fastText* [21] word embedding model. Abbreviation disambiguation makes use of the document context, for which an initial run of TF/IDF is necessary. Compound splitting also requires the *fastText* model and is the final step of the pre-processing pipeline. Note that the order of steps in the pipeline is obligatory: Tokenization, cleaning and POS tagging need to be handled first, since the subsequent token-modifying steps require clean singly tagged tokens. The latter three steps must begin with lemmatization, because abbreviation resolution and compound splitting perform better on lemmatized tokens. Abbreviation resolution must precede compound splitting, as the resolution of an abbreviation can be a compound word, i.e. the token *HWK* is left unchanged by the lemmatizer and is then resolved to the compound word *Halswirbelkörper* (cervical vertebral body) during abbreviation resolution and is then split into $Hals|Wirbel|K\ddot{o}rper$.

This pre-processing pipeline is applied to each text used in term extraction, i.e. training and test documents and the dictionary of technical terms.

B. Term Extraction

We now describe our semantic aware *Snowball* method for single term extraction as an alternative for pure frequency-oriented methods.

The name *Snowball* reflects the basic idea of the method: layers of terms are added iteratively to the body of terms according to their similarity to terms already in the snowball

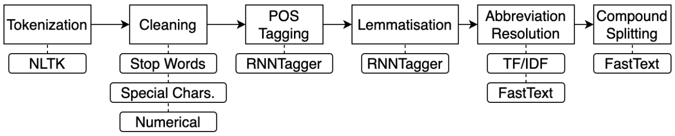

Fig. 1. Pre-processing pipeline.

Algorithm 1: Snowball

```
input: $voc$ = word2vec model vocabulary set, $rv$ =
  dictionary of technical terms ;
$cc = f(rv \cap voc)$;
$layer_0 = cc$;
i=0;
repeat
    $cc = cc \cup layer_i$;
    $\theta_{i+1} = \theta(i+1)$;
    $layer_{i+1} = f(\bigcup_{v \in layer_i} N_{\theta_{i+1}}(t) \setminus cc)$;
    $i = i + 1$;
until $layer_i = \emptyset$;
output: $cc$ set of concept candidates;
```

(see Fig. 2). The core seeding set of concept candidates is the intersection of a source corpus voc and the pre-processed technical term dictionary vocabulary rv. We use *word2vec* embeddings [22] for both the source corpus and the technical term vocabulary. Starting with this core set, more domain relevant terms are collected layer-wise by adding neighbors of each of the members in voc to the candidate set. Each term for which a concept candidate with a cosine similarity above a dynamically adapted threshold θ_i exists is added to the next layer.

Let $sim_{cos}(x, y)$ be the cosine similarity of two terms x, y relative to some trained *word2vec* vector representation of a vocabulary V with $x, y \in V$. For $v \in V$ and $0 \leq \theta \leq 1$ we define the *θ-similarity neighborhood* of v as $N_\theta(v) = \{u \in V : sim_{cos}(v, u) \geq \theta\}$. $f : 2^V \to 2^V$ is a linguistic filter that only lets pass nouns, proper names and attributive adjectives. The value of θ_i is computed with the following sigmoidal function $\theta\left(i\right) = \frac{0.2}{1+e^{-(i-m)}} + 0.7$ so that the threshold grows asymptotically towards 0.9 and requires higher similarity the further we move away from the core set. Parameter settings for m are discussed below.

Our snowball method (see alg. 1) runs until no further terms are extracted. Alternatively, a maximum number of iterations i_{max} could be specified as stop criterion.

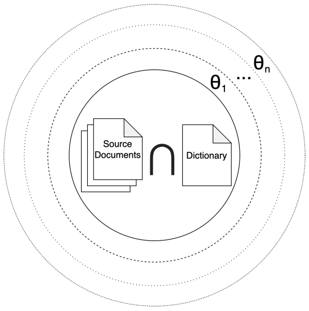

Fig. 2. Snowball uses the intersection of source corpus and dictionary of technical terms vocabulary as seeding set. Layers are added successively using word similarities with different similarity thresholds.

In the next section we apply *Snowball* and TF/IDF on exemplary test cases to compare the results achieved with both methods. We use TF/IDF as described in [23] with a dynamically shortened set of extracted terms to reduce the number of false positives which are caused by terms of lower relevance extracted from documents too short. Therefore, we restrict term extraction to the top n TF/IDF values, with n set to 67.4% of the document length for full coverage and 44.1% for part coverage. Both values for n were determined with grid search optimizing the F_1 score. Beyond comparing the two methods, we also investigate potential synergies by combining the results of both methods.

As measure for the quality of the extracted set of terms in the following we observe the coverage of the final set of candidate terms on the annotated test documents in different experimental setups.

IV. Evaluation

In our evaluation we first compare results achieved by our *Snowball* method with results of TF/IDF on a discipline-specific corpus. We chose TF/IDF, since it is a common baseline of unsupervised statistics-based key phrase extraction methods [24]. The second aspect of the evaluation is measuring the impact of single steps (lemmatization, abbreviation resolution, compound splitting) of our pre-precessing pipeline on the quality of the term extraction. Additionally, we illustrate the impact of the parameter m on the θ function for the similarity threshold in the *Snowball* method.

A. Experimental Setup

We provide a repository[2] for our evaluation standard of the *Jena Synthetic Clinical Corpus* (JSynCC) [25] and a parameter file for word2vec, TF/IDF and *Snowball*. JSynCC[3] and the *gensim word2vec*[4] implementation are available on the internet.

Since our later use case is based on the German language, we decided to use the ICD-10-GM[5] vocabulary as dictionary of technical terms and JSynCC as data basis for our test. Otherwise, it is difficult to find German data in the clinical domain for NLP applications, such as *Annotated Corpora for Term Extraction Research* (ACTER) [26] which is a data set for term extraction in text for heart failure or the *Medical Information Mart for Intensive Care* (MIMIC-III). It holds a extensive collection of patient data [27], but both data sets are not available in German.

Our built of the JSynCC consists of 1,058 medical case descriptions, henceforth referred to as *documents*, originating from 10 books of diverse topics. While each of the books has been considered for equal topic distribution, we randomly selected 5% of the documents as test cases for term extraction. We obtained 1,006 documents for the source corpus and 42 documents for the test corpus, whereas 11 duplicates

[2] https://github.com/TLDia2019/4TH-IEEE-MNLP

[3] https://github.com/JULIELab/jsyncc

[4] https://radimrehurek.com/gensim/

[5] ICD-10 German Modification

were removed from overall corpus. For validation purposes, the test corpus documents were annotated manually. Three expert annotators collected single and multi-word expressions, such as `Röntgenaufnahme` (X-Ray) and `schmerzhafte Schwellung` (painful swelling), respectively, for each of the test documents and agreed in a discussion on a consensus list of 2,583 expressions in total. For this reason we also considered the evaluation on multi-word expressions using partial coverage of terms.

The JSynCC *word2vec* model is trained using CBOW and hierarchical softmax. Because of the comparatively small corpus size we chose an embedding dimension of 16 (compare rule of thumb: $dim = \sqrt[4]{|vocab|}$ [28]) with a relatively high window size of 14 with the aim to consider a rather broad context. Minimum word count is set to 3 and the model is trained in 4 iterations. The intersection is determined on the text representations of the corpus and the ICD-10-GM vocabulary.

Performance measures of TF/IDF and *Snowball* are compared as well as the impact of the pre-processing steps. We measure the coverage on terms as full and partial coverage so that we also consider tagging of multi-word expressions in the annotated list. Full coverage means that a term completely matches an annotated expression. For example `Röntgenaufnahme` is found as a single term and is also element of the annotated expression list. A term is considered as partly covered if it is a part of the annotated expression. For instance, the existing overlap of the extracted term `Schwellung` and the annotated expression `schmerzhafte Schwellung` is counted as a partial coverage.

The pre-processing pipeline can be configured to schedule pre-processing steps in the pipeline for ablation studies.

Table I depicts the setup of pre-processing configurations under examination. While configuration C_* includes the pre-processing steps lemmatization, abbreviation resolving and compound splitting, $C_\perp$ does not make use of any of them. Iterations C_L and C_A include lemmatization and abbreviation resolving, respectively. Compound splitting is enabled in C_C.

For each of the five configurations, source and test documents have been treated identically. After processing the source data, test data was processed and the extracted terms were tagged in the documents.

Recall, precision and F_1 score have been measured for the five configurations for each method. Those are TF/IDF, *Snowball* and the combination of both ($combined = \{$TF/IDF $\cup$ *Snowball*$\}$).

Regarding our novel method, we make the hypothesis $H1$, which assumes that the *Snowball* method performs well under the conditions of a small corpus in a specific language. For the ablation study, we formulate two more hypotheses. Hypothesis $H2$ assumes that $C_\perp$ performs worse than C_*, since in C_* all pre-processing steps are activated and therefore are supposed to yield better extraction performance due to text normalization. We assume that we reduce data sparsity applying lemmatization and compound splitting and refine semantic relations inside the word embeddings by resolving abbreviations. The third hypothesis $H3$ assumes that the combination of all pre-processing steps C_* provides better term extraction results than the standalone configurations C_L, C_A, and C_C.

TABLE I
PIPELINE CONFIGURATIONS WITH VARYING PRE-PROCESSING STEPS INCLUDED.

Method	C_*	$C_\perp$	C_L	C_A	C_C
Lemmatization	x		x		
Abbreviation Resolving	x			x	
Compound Splitting	x				x

B. Experimental Results

Fig. 3 illustrates our processing pipelines output given an input text[6]. The output text is tokenised into sentences (line numbers) and tokens. Lemmata and resolved abbreviations are marked with square brackets and curly brackets, respectively, while angle brackets denote split compounds. Extracted terms (TF/IDF & *Snowball*) are marked with asterisk and annotated terms with circumflex.

Table II depicts the precision (P), recall (R), and F_1 score (F_1) values for the five configurations each used together with TF/IDF, *Snowball* and the combination of both. This shows that for every configuration the precision of *Snowball* is higher than of TF/IDF. TF/IDF principally achieved higher recall values compared to *Snowball* except for partial coverage in C_C. Also, the F_1 score of *Snowball* is higher in comparison to TF/IDF in all configurations. The combined approach shows higher recall values than *Snowball* and TF/IDF separated and mixed results for precision and F_1. The high recall of

```
[INPUT]
Ein 17-jähriger Junge ist beim Inlineskaten auf die
rechte Schulter gefallen. Bei der klinischen
Untersuchung finden Sie eine schmerzhafte Schwellung
im Bereich der Schaftmitte der rechten Klavikula.
Sie veranlassen eine Röntgenaufnahme der Klavikula
a. p. und tangential.

[OUTPUT]
1   Ein[ein] 17-jähriger[17-jährig]<17|jährig>
    Junge* ist beim[bei] Inlineskaten* auf die[der]
    rechte^ Schulter^* gefallen[fallen].
2   Bei der klinischen[klinisch]^ Untersuchung^*
    finden Sie eine[ein] schmerzhafte[schmerzhaft]^
    Schwellung^* im[in] Bereich* der Schaftmitte
    <Schaft|Mitte>* der rechten[rechte]^
    Klavikula^*.
3   Sie veranlassen eine[ein] Röntgenaufnahme
    <Röntgen|Aufnahme>^* der Klavikula* a. p.
    {anterior-posterior}^ und tangential^*.
```

Fig. 3. Exemplary input and output of the pre-processing and term extraction. Square brackets denote lemmata, curly brackets indicate a resolved abbreviation and angle brackets split compounds. ($*$) denotes extracted terms, (ˆ) annotated expressions. Text taken from [29]

[6] Translatable as: "A 17-year-old boy has fallen on his right shoulder while inline skating. During the clinical examination you discover a painful swelling in the area of the shaft center of the right clavicle. You order an X-ray of the clavicle a. p. and tangential."

TABLE II
TERM EXTRACTION RESULTS PER CONFIGURATION AND METHOD.

Conf.	Cov.	TF/IDF P	TF/IDF R	TF/IDF F_1	Snowball P	Snowball R	Snowball F_1	Combined P	Combined R	Combined F_1
C_*	Partial	45.72	**77.35**	57.47	**58.69**	**75.53**	**66.05**	44.92	97.21	61.45
	Full	23.33	**57.22**	**33.15**	**35.29**	**45.41**	**39.72**	23.28	**58.54**	33.31
$C_\perp$	Partial	**49.85**	69.73	58.14	55.85	66.55	60.73	46.25	96.36	62.50
	Full	22.97	48.63	31.20	32.23	38.40	35.05	24.08	58.42	34.10
C_L	Partial	49.73	72.13	**58.87**	57.90	66.94	62.09	**47.24**	95.93	**63.31**
	Full	**23.41**	51.18	32.13	34.06	39.37	36.52	**24.29**	58.34	**34.30**
C_A	Partial	49.63	69.53	57.92	55.73	67.36	61.00	46.08	96.40	62.35
	Full	22.89	48.43	31.09	31.90	38.56	34.92	24.11	58.42	34.13
C_C	Partial	45.33	73.48	56.07	56.66	74.10	64.22	43.87	**97.44**	60.50
	Full	22.57	54.78	31.97	33.68	44.06	38.18	22.94	**58.54**	32.96

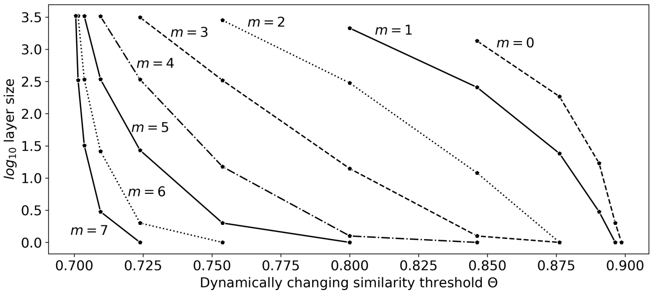

Fig. 4. Increase of vocabulary per layer depending on θ which is influenced by parameter m of the sigmoidal function.

combined indicate the benefits of using both methods to cover almost all relevant elements. This is emphasized by the small intersection in C_*, where $|\text{TF/IDF} \cup \textit{Snowball}| = 3,779$ and $|\text{TF/IDF} \cap \textit{Snowball}| = 1,610$ terms.

Direct comparison of the different configurations shows that C_* has achieved the highest recall for TF/IDF and *Snowball*. This also holds for the precision and F_1 of *Snowball* and the F_1 for full coverage of TF/IDF. Furthermore, applying no pre-processing $C_\perp$ gives the best precision result for partial coverage on TF/IDF. C_L leads to the best precision results for TF/IDF (full coverage) and both coverage types in *combined*. Additionally, it provides the best results for F_1 in *combined* and F_1 (partial coverage) of TF/IDF. C_C gives the highest recall for *combined*.

Fig. 4 illustrates the impact of choosing different values for the parameter m of the threshold function θ. Having set the upper and lower boundaries to 0.7 and 0.9, respectively, results in a first threshold of 0.85 for $m = 0$. The algorithm converges after 4 layers with 1,547 candidate terms, which is the lowest number for this experiment. With increasing m, the final number of candidate terms increases, too, while the number of layers fluctuates between 3 and 4. Choosing $m = 7$ gives an initial threshold of 0.7, and a final similarity threshold of 0.72 after 4 layers and 3,664 candidate terms in total. The higher m is set, the steeper is the decline of the layer size with increasing θ. To stay in the snowball metaphor: the higher the value of m the more snow is collected in the beginning, eventually resulting in the biggest snowball.
During a grid search, we found the optimal m regarding F_1 score is 5.

C. Implications

We consider H1 confirmed since *Snowball* has a higher F_1 score than TF/IDF due to a higher precision and a comparable recall (partial) in C_*. Additionally, we can see that the intersection of TF/IDF and *Snowball* is small. This makes *Snowball* a good complement for TF/IDF. The intersection of the ICD-10-GM corpus and JSynCC contains between 9,000 and 11,000 expressions, depending on the pre-processing configuration. *Snowball* makes use of around 40% of the terms, i.e. the *Snowball* method supplies a large selection of terms.

Hypothesis H2 has been confirmed for *Snowball*. TF/IDF precision and F_1 score of $C_\perp$ is higher for partial coverage metrics compared to C_*, where recall is still higher in C_*. For TF/IDF, the effect of pre-processing is varying and requires further analysis regarding the impact of each step.

With regard to H2, we have seen that all pre-processing steps together (C_*) lead to generally good results. The *Snowball* results show that all separated processing steps have lower values than C_*. Compound splitting C_C has the greatest impact on recall and F_1 score, while lemmatization C_L improves precision. On the other hand, compound splitting has a negative effect on TF/IDF precision. The best precision for TF/IDF is obtained in $C_\perp$ (partial coverage) or C_L (full coverage). Abbreviation resolution generally shows no improvement to the methods. Compound splitting increases the terms that are found in general, this is mainly reflected in a higher true positive and false positive rate. Lemmatization reduces the overall terms that are found but mostly false negatives. Abbreviation resolving seems to have a marginal impact on the term extraction at all. The reason for this may be the small size of the corpus and the rate of abbreviations per document. The above observations lead to the interpretation that hypothesis H3 is confirmed for the *Snowball* method but rejected for TF/IDF, since C_L has higher precision (full coverage) and F_1 score (partial coverage) than C_*.

V. CONCLUSION AND FUTURE WORK

The proposed approach of machine learning using word embeddings with shallow linguistic information has achieved promising experimental results. It is liable that further application fields for term extraction will benefit in addition to the chosen application field of coding German medical documents.

In our experiments the combination of the well established statistical extraction method TF/IDF with our novel *Snowball* method has a higher term coverage than standalone application. The layer-wise addition of candidate terms using a *word2vec* model makes use of semantic features in a corpus based on a discipline-specific seeding set of vocabulary. The pre-processing ablation study revealed the positive effects of lemmatization for both methods and compound splitting for *Snowball*, while the results of abbreviation resolution remained below our expectations.

The long-term goal of our term extraction is the enrichment of an ontology in the medical domain (MeSH). In next steps of our research we will therefore address the subsequent steps in the ontology enrichment process. We focus on the extraction of relations between terms and between terms and concepts in the base ontology. Further, we will work on improving the pre-processing pipeline by adding negation handling and spell checking which is promising because analysis of our corpus showed numerous negations and spelling mistakes. Also, further evaluation of the meaningfulness of components such as abbreviation resolution, which is currently an error-prone method. It increases data sparsity if it creates two different resolutions for one acronym in the same document. We expect better results using *BERT* for word disambiguation. The source corpus for the *word2vec* model can be increased in size with the use of additional corpora, for example Wikipedia articles belonging to the *Medicine* category. Pre-trained models, e.g. *fastText*, will also be tested. Additionally, further examination of the growth of the *Snowball* layers and the assessment of the layer quality will provide deeper insight into the behavior of that method. The feasibility of extending our approach to a multi-term expression extraction method by applying a n-gram-based *word2vec* model will also be investigated, to fully match our annotated expression. Additional potential candidate methods are *C-Value/NC-Value*, n-gram-based TF/IDF and conditional random fields (CRFs).

Acknowledgments

The authors acknowledge the assistance received from the medical coding experts of MINDS medical and the student researchers Atilla Yetkin, Tobias Schmidt, and Oxana Malyshkina.

References

[1] Snomed International, “March 2020 INTERIM release of SNOMED Clinical Terms,” 2020. [Online]. Available: www.snomed.org,lastaccess: May4,2020

[2] National Library of Medicine. MeSH - Medical Subject Headings. (2020, May 5). [Online]. Available: https://www.nlm.nih.gov/mesh/

[3] ——. PubMed. (2020, May 5). [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/

[4] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proc.*, 2013.

[5] D. Kudenko, “Special Issue on Transfer Learning,” *KI*, vol. 28, no. 1, pp. 5–6, 2014.

[6] K. Kageura and B. Umino, “Methods of automatic term recognition: A review,” *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, vol. 3, no. 2, pp. 259–289, 1996.

[7] N. Astrakhantsev, “ATR4S: toolkit with state-of-the-art automatic terms recognition methods in Scala,” *Language Resources and Evaluation*, vol. 52, no. 3, pp. 853–872, 2018.

[8] W. Wong, W. Liu, and M. Bennamoun, “Determination of Unithood and Termhood for Term Recognition,” in *Handbook of Research on Text and Web Mining Technologies*. IGI Global, 2009.

[9] G. K. Savova, J. J. Masanz, P. V. Ogren, J. Zheng, S. Sohn, K. K. Schuler, and C. G. Chute, “Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications,” *JAMIA*, vol. 17, no. 5, pp. 507–513, 2010.

[10] O. Bodenreider, “The unified medical language system (UMLS): integrating biomedical terminology,” *Nucleic Acids Research*, vol. 32, no. Database-Issue, pp. 267–270, 2004.

[11] W. Boag, E. Sergeeva, S. Kulshreshtha, P. Szolovits, A. Rumshisky, and T. Naumann, “Cliner 2.0: Accessible and accurate clinical concept extraction,” *CoRR*, 2018.

[12] M. Becker and B. Böckmann, “Extraction of UMLS® Concepts Using Apache cTAKES™ for German Language,” in *Health Informatics Meets eHealth - Predictive Modeling in Healthcare - From Prediction to Prevention - Proc. of the 10th eHealth2016 Conference*, ser. Studies in Health Technology and Informatics, vol. 223. IOS Press, 2016, pp. 71–76.

[13] S. Wiegreffe, E. Choi, S. Yan, J. Sun, and J. Eisenstein, “Clinical concept extraction for document-level coding,” in *Proc. of the 18th BioNLP Workshop and Shared Task*. Association for Computational Linguistics, 2019, pp. 261–272.

[14] A. Hazem, M. Bouhandi, F. Boudin, and B. Daille, “Termeval 2020: Taln-ls2n system for automatic term extraction,” in *Proc. of the 6th International Workshop on Computational Terminology*, 2020, pp. 95–100.

[15] L. V. Subramaniam, S. Mukherjea, P. Kankar, B. Srivastava, V. S. Batra, P. V. Kamesam, and R. Kothari, “Information extraction from biomedical literature: methodology, evaluation and an application,” in *Proc. of the 2003 ACM CIKM International Conference on Information and Knowledge Management, New Orleans, Louisiana, USA, November 2-8, 2003*. ACM, 2003, pp. 410–417.

[16] X. Zhou, X. Zhang, and X. Hu, “Maxmatcher: Biological concept extraction using approximate dictionary lookup,” in *PRICAI 2006: Trends in Artificial Intelligence, 9th Pacific Rim International Conference on Artificial Intelligence*, ser. Lecture Notes in Computer Science, vol. 4099. Springer, 2006, pp. 1145–1149.

[17] H. Nguyen and J. Patrick, “Text mining in clinical domain: Dealing with noise,” in *Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. ACM, 2016, pp. 549–558.

[18] M. Weller-Di Marco, “Simple compound splitting for german,” in *Proc. of the 13th Workshop on Multiword Expressions (MWE 2017)*. Association for Computational Linguistics, 2017, pp. 161–166.

[19] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. O’Reilly Media, Inc., 2009.

[20] H. Schmid, “Deep learning-based morphological taggers and lemmatizers for annotating historical texts,” in *Proc. of the 3rd International Conference on Digital Access to Textual Cultural Heritage, DATeCH 2019, Brussels, Belgium, May 08-10, 2019*. ACM, 2019, pp. 133–137.

[21] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, “Learning word vectors for 157 languages,” in *Proc. of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA), 2018.

[22] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” *CoRR*, vol. abs/1310.4546, 2013. [Online]. Available: http://arxiv.org/abs/1310.4546

[23] K. S. Jones, “A statistical interpretation of term specificity and its application in retrieval,” *Journal of Documentation*, vol. 28, pp. 11–21, 1972.

[24] E. Papagiannopoulou and G. Tsoumakas, “A review of keyphrase extraction,” *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 10, no. 2, 2020. [Online]. Available: https://doi.org/10.1002/widm.1339

[25] C. Lohr, S. Buechel, and U. Hahn, “Sharing Copies of Synthetic Clinical Corpora without Physical Distribution — A Case Study to Get Around IPRs and Privacy Constraints Featuring the German JSYNCC Corpus,” in *Proc. of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA), 2018.

[26] T. A. Rigouts, V. Hoste, and E. Lefever, “In no uncertain terms: a dataset for monolingual and multilingual automatic term extraction from comparable corpora,” *Language Resources and Evaluation*, vol. 54, no. 2, pp. 385–418, 2020.

[27] A. E. Johnson, T. J. Pollard, L. S. Lu, L. wei H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, C. L. Anthony, and R. G. Mark, “MIMIC-III, a freely accessible critical care database,” *Scientific Data*, vol. 3, no. 1, 2016, paper no. 160035.

[28] Google Inc. Introducing TensorFlow Feature Columns. (2020, October 1). [Online]. Available: https://developers.googleblog.com/2017/11/introducing-tensorflow-feature-columns.html

[29] S. Eisoldt, *Fallbuch Chirurgie - 140 Fälle aktiv bearbeiten*. Stuttgart: Georg Thieme Verlag, 2017.