German Medical Data Sciences 2023 — Science. Close to People.
R. Röhrig et al. (Eds.)

doi:10.3233/SHTI230712

DE-Lemma: A Maximum-Entropy Based Lemmatizer for German Medical Text

Martin WIESNER[a,1]
[a] *Dept. of Medical Informatics, Heilbronn University, Heilbronn, Germany*

Abstract. When processing written German language, it is helpful, to use the base form (or: lemma) of possibly inflected words, such as verbs, nouns or named entities. However, for German text from the (bio)medical domain, e.g., discharge letters, or entries stored in electronic medical or health records (EMR, EHR), difficulties exist in finding the correct lemma, as, for instance, the medical language has roots in Latin or Greek. In such cases, stemming techniques might provide inaccurate results for text written in German. This study demonstrates a Machine Learning approach for training *Apache OpenNLP*-based lemmatizer models from publicly available German treebanks. The resulting four "DE-Lemma" models were evaluated against a sample of (bio)medical nouns, randomly selected from real-world discharge letters. The most promising DE-Lemma model achieved an accuracy of 88.0% (F_1 = .936).

Keywords. Natural Language Processing, Machine Learning, Text Mining

1. Introduction

When processing written German language, it is helpful, if not necessary, to use the base form of possibly inflected words [1], since this is the form to which all other forms refer. For example, all potential forms of a verb or noun can be automatically recognized if the base form, or *lemma*, is known. As a result, automatic text processing can be performed more effectively. This facilitates *Natural Language Processing* (NLP) tasks such as *Information Extraction* (IE) [2] and *Information Retrieval* (IR) and ensures that all related words are correctly recognized and processed.

1.1. Background

Several difficulties can arise when reducing German words to their respective lemma:

(i) Irregular verbs: Many verbs have irregular conjugations, which means that they do not always behave according to the regular rules for forming root forms.
(ii) Word formation processes: Many words are formed by word-formation processes, including derivation, composition, and conversion.
(iii) Foreign words: Many German words have roots in other languages, especially Latin, Greek, French, and English. Identifying the correct lemma can thus be more complex.
(iv) Colloquial language: Colloquialisms and dialects may differ from the standard language and use different grammatical rules and vocabulary.

[1] Corresponding Author, Max-Planck-Str. 39, 74081 Heilbronn, DE; E-mail: wiesner@hs-heilbronn.de.

Different approaches have been proposed to find the base form of nouns or named entities in written (biomedical) text:

(a) Rule-based: via rules and patterns created based on language expertise and linguistics.
(b) Dictionary-based: via a dictionary or lexicon to identify the different forms of a word and find the corresponding base form. This method has been demonstrated in many NLP systems [3, 4] but requires a lot of effort to create and maintain the underlying dictionary (e.g., names, places, or diseases).
(c) Stemming: a procedure in which the "root stem" of a word is determined by truncation (e.g., "Patienten" becomes the base form "Patient"). For the (bio)medical domain, such a root stem is often not a lexicographically correct word as it would not be found *as is* in dictionaries, since many German words have irregular plural forms (e.g., the plural "Bakterien" with lemma: "Bakterium"; likewise, "Operationssäle": "Operationssaal").
(d) Parts-of-Speech Tagging (POS): this method determines the part of speech of each token (e.g., noun, adjective, verb) and can help to find the basic forms of nouns.
(e) Machine Learning (ML) based approaches: Here, models are trained to automatically recognize base forms of words. This usually requires large amounts of annotated data, that is corpora or treebanks, to train the model.
(f) Hybrid approaches: combine multiple approaches to achieve improved results. An example is the use of rule-based and dictionary-based methods.

All approaches have been implemented in Python, Java, or other programming languages. However, to the best of the author's knowledge, no ML-based model has been published or made available for identifying German lemmas efficiently in the (bio)medical domain.

1.2. Objectives

This study demonstrates the technical feasibility of training lemmatizer models for processing German (bio)medical text via machine-processable treebanks. Moreover, it investigates the suitability of the resulting models for real-world, unstructured clinical data. All model files are freely available for NLP software in the medical domain.

2. Methods

Expert-curated linguistic resources exist for the processing of natural language, referred to as treebanks. These can be used to implement ML components for processing un- or semi-structured text in information systems. The *Universal Dependencies* (UD) project [5] provides a specification for consistent annotation of grammar (part of speech, morphological features, and syntactic dependencies) in different human languages [6].

2.1. Material

Several available treebanks, in CoNLL-U [7], or CoNLL-X [8] format, were identified and selected as candidates for training German lemmatizer models, see Table 1.

Both UD-labeled treebanks, UD-GSD [9] and UD-HDT [10], are constructed from text corpora of German newspapers and other freely available text materials.

The annotation levels in TüBa-D/DP [11] and TüBa-D/W [12] contain information about word types, morphology, lemmas, as well as dependency relations. TüBa-D/W is a huge corpus: It is based on Wikipedia text material including 36.1 million sentences. For the training phase, CoNLL-X treebanks had to be converted into the CoNLL-U format which is better supported in most NLP tools.

Table 1. Basic properties for a selection of available German treebanks; note: *mil = million*

Name	Label	# Sentences	# Words	# Tokens
Universal dep. treebank v2.0 (legacy)	UD-GSD	15.590	292.773	287.725
Hamburg dep. treebank	UD-HDT	189.928	~3.46 mil	~3.4 mil
TüBa-D/DP release 5, *German political speeches*	TüBa-D/DP	619.152	~12.8 mil	-
TüBa-D/W release 0 *Wikipedia corpus*	TüBa-D/W	~36.1 mil	~615 mil	-

2.2. Training

The training was conducted based on the open-source framework Apache OpenNLP [13], in version 2.1.0. Conveniently, in OpenNLP, a robust *Maximum Entropy* (ME) [14] implementation, and a solid and well-tested software stack to train and practically use models, are already provided built in. For this reason, the treebanks listed in Table 1 were used to train a corresponding ME model, each.

For the generation of lemmatizer models with smaller treebanks (UD-GSD, UD-HDT, TüBa-D/DP-political), the OpenNLP training parameters were chosen as follows:

```
training.algorithm=maxent; training.iterations=100; training.cutoff=5;
training.threads=16; language=de; use.token.end=false;
sentences.per.sample=5; upos.tagset=upos
```

The training for TüBa-D/W was conducted with these parameters:

```
training.algorithm=maxent;training.iterations=20; training.cutoff=5;
training.threads=4; language=de; use.token.end=false;
sentences.per.sample=5; upos.tagset=upos
```

The resulting binary model files were persisted for evaluation and later re-use in NLP applications with a lemmatizer component.

The execution environment of the training program was a Java Runtime Environment (JRE), a 64bit OpenJDK in version 8 build 292.

2.3. Evaluation

From clinical text material, $n = 100$ sentences were randomly extracted from discharge letters of the *Chest Pain Unit* at the Heidelberg University Hospital. Part-of-speech tagging was used to extract one word each that represented a medical noun. From a technical point of view, nouns offer high value for the realization of IE or IR systems, as nouns often act as the most important terms for identifying relevant documents, for instance, in a database or for matching against an index.

For the sentence splitting and POS tagging, a default German OpenNLP model was used which can be downloaded on the project's website [13].

After randomized sampling sentences and identification of the medical nouns, German dictionaries were used to identify the corresponding lemma as reference. Thereby, each medical noun, as documented by a physician in a real-world discharge letter, was mapped to the correct lemma of that noun. Subsequently, each trained model was evaluated for its accuracy[2] and via the F_1 measure [15].

3. Results

Since the training of a lemmatizer model (LM) required between ~32 GB (UD-GSD) and ~1,100 GB (TüBa-D/W) of RAM at runtime, these tasks could not be performed on conventional workstation hardware. Therefore, the training of each model was conducted on the mainframe environment of the *bwUniCluster* [16] during October 2022. For the smaller models (UD-GSD, UD-HDT, TüBa-D/DP/political) a few hours were required; the training with the TüBa-D/W took 2 days, 7 hours and 1 minute on the *bwHPC* cluster.

3.1. Training

The main, technical characteristics of the resulting LM models are listed in Table 2.

Table 2. Technical properties of the *DE-Lemma* models (LM) trained with Apache OpenNLP.

Model	Name	Disk required (MB)	RAM required[a] (GB)
LM_{GSD}	DE-Lemma_UD-gsd-2022-maxent.bin	0.9	~0.15
LM_{HDT}	DE-Lemma_UD-hdt-2022-maxent.bin	14.8	~0.50
$LM_{TÜDP}$	DE-Lemma_Tue-BuReg-2022-maxent.bin	4.1	~0.30
$LM_{TÜDW}$	DE-Lemma_Tue-Wiki-2022-maxent.bin	137.7	~2.00

[a] Note: estimates for loading the model into the main memory (RAM) in a JRE 11 application.

All models presented in Table 2 are freely available and can be downloaded from: https://github.com/mawiesne/DE-Lemma

3.2. Evaluation

The evaluation outcome for all obtained models (LM_{GSD}, LM_{HDT}, $LM_{TÜDP}$, $LM_{TÜDW}$) against a random sample of $n = 100$ inflected medical nouns is shown in Table 3.

Table 3. Accuracy and F_1 scores for the trained *DE-Lemma* models (LM); n = 100.

Model	Accuracy	F_1
LM_{GSD}	.500	.667
LM_{HDT}	.410	.582
$LM_{TÜDP}$	.670	.802
$LM_{TÜDW}$	.880	.936

Both models trained via the UD treebanks (LM_{GSD}, LM_{HDT}) showed an unsatisfactory performance in recognizing correct lemmas for given medical nouns. In contrast, the models based on the TüBa treebanks ($LM_{TÜDP}$, $LM_{TÜDW}$) detected those

[2] Here: proportion of correct lemma predictions among the total number of samples.

much more accurately. The $LM_{TÜDW}$ model is outperforms all other models with an accuracy of 88%.
News-based treebanks tended to be too generic for biomedical scenarios. A brief error analysis revealed: several medical nouns were not contained in the $UD_{GSD\,/\,HDT}$ treebanks. Consequently, models trained with those resources are less suitable for the target domain.

4. Discussion

Publicly available treebanks containing lemma annotations are suitable for training lemmatizer models. These can be used in NLP software components to reduce (bio)medical nouns in inflected form to their corresponding lemmas. However, evaluation showed that the accuracy is satisfactory only for one treebank and the associated model ($LM_{TÜDW}$). The TüBa-D/W treebank, in contrast to the other selected treebanks, includes Wikipedia articles on medical topics. This resulted in a greater thematic variety and thus a broader scope of $LM_{TÜDW}$. Consequently, $LM_{TÜDW}$ constitutes an applicable model that can be integrated into NLP-focused software systems.

4.1. Comparison with Prior Work

A systematic literature review by Pramana et al. [17] found that lemmatization techniques excel in many application scenarios for a broad range of natural languages.

Gleim et al. surveyed lemmatization and tagging techniques in 2019 [18]. They reported that "LemmaTag [19] performs best in most cases" with lemmatization accuracies between 49.6% to 98.7%, depending on different evaluation sets (cf. Table 21 in [18]).

A study by Ortmann et al. reported lemmatization accuracies for different lemmatizer components [20] with a set of German text material with broader topics than in this study. They investigated 9 lemmatizers (cf. Table 4 in [20]), with accuracies ranging from 86.5% to 97.5%. In comparison with the results from [18, 20], any OpenNLP-based lemmatizer using the $LM_{TÜDW}$ model could be considered a competitive player.

4.2. Limitations & Future Directions

The study made use of existing third-party treebanks of different size and covered topics. The topical variety, the quality of the linguistic annotations, and the recency could not be controlled for this study, due to the sheer size of the corpora (see Table 1).

The evaluation sample was restricted to 100 (bio)medical nouns. However, those were randomly sampled from discharge letters. Clearly, the evaluation should be expanded to a larger set of nouns and/or named entities. Yet, no broadly accepted gold standard exists for the German (bio)medical domain. Constructing a large ground truth evaluation set provides potential for future research and joint efforts in the German NLP community.

The Apache OpenNLP is a solid and well-known software framework for training and evaluating NLP models. However, other commercial and open-source frameworks exist suited for tasks conducted in this study. Consequently, a comparative evaluation with the TüBa-D/W treebank and the exact same, or extended evaluation sample remains an open task. Moreover, a comparison of recent lemmatization techniques, such as

contextualized embeddings, might also be a valuable contribution for the German NLP community.

5. Conclusions

This study demonstrated the feasibility of training German lemmatizer models intended for the use in (bio)medical domain. The trained model files, available in binary OpenNLP format, are a contribution for scientific comparisons, or practical use in NLP software components. The *DE-Lemma* model $LM_{TÜDW}$ achieves an accuracy of 88.0% for an evaluation set constructed from nouns in real-world discharge letters.

Acknowledgements

The author acknowledges support by the state of Baden-Württemberg through *bwHPC*.

References

[1] Wartena C. A Probabilistic Morphology Model for German Lemmatization. Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019). 2019; Erlangen, Germany: German Society for Computational Linguistics & Language Technology. p. 40–49.
[2] Bay M, Bruneß D, Herold M, Schulze C, Guckert M, Minor M. Term Extraction from Medical Documents Using Word Embeddings. Proceedings of the 6th IEEE Congress on Information Science and Technology (CiSt). 2020. p. 328–333.
[3] Perera P, Witte R. A Self-Learning Context-Aware Lemmatizer for German. Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing. 2005; Vancouver, BC, Canada: Association for Computational Linguistics. p. 636–643.
[4] Eger S, vor der Brück T, Mehler A. Lexicon-assisted tagging and lemmatization in Latin: A comparison of six taggers and two lemmatization methods. Proceedings of the 9th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH). 2015; Beijing, China: Association for Computational Linguistics. p. 105–113.
[5] de Marneffe M-C, Manning CD, Nivre J, Zeman D. Universal Dependencies. Computational Linguistics. 2021; **47**:255–308.
[6] Rosa R, Mašek J, Mareček D, Popel M, Zeman D, Žabokrtský Z. HamleDT 2.0: Thirty Dependency Treebanks Stanfordized. Proceedings of the Ninth International Conference on Language Resources and Evaluation. 2014; Reykjavik, Iceland: European Language Resources Association; p. 26-31.
[7] Nivre J, de Marneffe M-C, Ginter F, Goldberg Y, Hajič J, Manning CD, et al. CoNLL-U Format [Internet]. Universal Dependencies. [cited 2023 Mar 31]. Available from: https://universaldependencies.org/format.html.
[8] Buchholz S, Marsi E. CoNLL-X Shared Task on Multilingual Dependency Parsing. Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X). 2006; New York City, USA: Association for Computational Linguistics. p. 149–164.
[9] McDonald R, Nivre J, Quirmbach-Brundage Y, Goldberg Y, Das D, Ganchev K, et al. Universal Dependency Annotation for Multilingual Parsing. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Vol. 2: Short Papers). 2013; Sofia, Bulgaria: Association for Computational Linguistics. p. 92–97.
[10] Borges Völker E, Wendt M, Hennig F, Köhn A. HDT-UD: A very large Universal Dependencies Treebank for German. Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019). 2019; Paris, France: Association for Computational Linguistics. p. 46–57.
[11] de Kok D, Pütz S. TüBa-D/DP stylebook [Internet]. 5th ed. Tübingen, Germany: Seminar für Sprachwissenschaft; 2019 [cited 2023 Mar 31]. Available from: https://www.sfs.uni-tuebingen.de/resources/tueba-ddp-stylebook.pdf.
[12] de Kok D. TüBa-D/W: A large Dependency Treebank for German [Internet]. Tübingen, Germany; 2014. p. 271–278. Available from: http://tlt13.sfs.uni-tuebingen.de/tlt13-proceedings.pdf.

[13] Apache Software Foundation. Apache OpenNLP [Internet]. Apache OpenNLP. 2023 [cited 2023 Mar 26]. Available from: https://opennlp.apache.org/.
[14] Ratnaparkhi A. A Maximum Entropy Model for Part-Of-Speech Tagging. Proceedings of the Conference on Empirical Methods in Natural Language Processing. 1996; Philadelphia, PA, USA: Association for Computational Linguistics. p. 133–142.
[15] Zhang, E., Zhang, Y. (2009). F-Measure. In: LIU, L., ÖZSU, M.T. (eds) Encyclopedia of Database Systems. Springer, Boston, MA.
[16] Steinbuch Centre of Computing (SCC) at Karlsruhe Institute of Technology. BwUniCluster2.0 [Internet]. Baden-Württemberg's high-performance computing (HPC). [cited 2023 Mar 31]. Available from: https://wiki.bwhpc.de/e/BwUniCluster2.0.
[17] Pramana R, et al. Systematic Literature Review of Stemming and Lemmatization Performance for Sentence Similarity. Proceedings of the 7th International Conference on Information Technology and Digital Applications (ICITDA). 2022; Yogyakarta, Indonesia: IEEE, pp. 1–6.
[18] Gleim R, Eger S, Mehler A, Uslu T, Hemati W, Lücking A, Henlein A, Kahlsdorf S, Hoenen A. A practitioner's view: a survey and comparison of lemmatization and morphological tagging in German and Latin. Journal of Language Modelling. 2019; **7**:1–52.
[19] Kondratyuk D, Gavenčiak T, Straka M, Hajič J. LemmaTag: Jointly Tagging and Lemmatizing for Morphologically Rich Languages with BRNNs. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 2018; Brussels, Belgium: Association for Computational Linguistics. p. 4921–4928.
[20] Ortmann K, Roussel A, Dipper S. Evaluating Off-the-Shelf NLP Tools for German. Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019). 2019; Erlangen, Germany: German Society for Computational Linguistics & Language Technology. p. 212–222.