A Crosslingual Approach to Dependency Parsing for Middle High German

Cora Haiber
Department of Linguistics
Fakultät für Philologie
Ruhr-Universität Bochum
Cora.Haiber@ruhr-uni-bochum.de

Abstract

This work presents the development and evaluation of a dependency parser for Middle High German Universal Dependencies utilising modern German as a support language for low-resource MHG. A neural dependency parser is trained with Stanza achieving UAS = 92.95 and LAS = 88.06. To ensure the parser's utility in facilitating and speeding up manual annotation to build a scaling UD treebank of MHG, a thorough error analysis shows the model's structural reliability as well as frequently confused labels. Hence, this work constitutes an effort to counterbalance the under-representation of historical languages in dependency treebanks and attend to the need of historical treebanks in contemporary linguistic research by utilising the UD extensions and accordingly annotated corpora published by Dipper et al. (2024).

1 Introduction

Historical linguistics is not only about understanding outdated or long-forgotten languages, but often brings valuable insight to the analysis of linguistic change in contemporary research. However, researchers in the historic field are bound to preserved written resources, which are often limited or of poor quality. Recently, computational linguistics, first and foremost Natural Language Processing (NLP), has become a field of great benefit for historical linguistics enabling the efficient exploitation of given resources in low-resource scenarios. Although the development of Universal Dependencies (UD) as a cross-lingual framework for morphosyntactic annotation encouraged the creation of dependency treebanks for various languages, historic stages of those languages are still underrepresented among syntactically parsed corpora. So far no treebank comparable in size to modern treebanks exists which includes dependency annotations for Middle High German (MHG), the language stage spoken and written in what is today southern and central Germany around the medieval period (1050–1350) and representing the beginnings of Modern High German in phoneme structure as well as syntax (Weddige, 2015).

As manual annotation is costly in time and effort, this work aims at the development of a neural dependency parser for MHG Universal Dependencies to be utilised in pre-annotation and correction when creating a scaling treebank. Due to the limited amount of annotated data, I will treat MHG as a low-resource language and explore modern German as a high-resource support language. Stanza[1] as a Python package known for dealing well with multi-linguality (Qi et al., 2020) is used for training the parser.

The paper is structured as follows. Section 2 introduces contemporary research in the fields of UD and NLP for low-resource languages. The available data published and annotated by Dipper et al. (2024) are described in Section 3. Section 4 introduces the methods of training conducted with Stanza. The results as well as details of the error analysis are presented in Section 5. The discussion of the results and some suggestions for future work on the parser follow in Section 6. The model instance, a script demonstrating its application and a collection of Python scripts developed for model evaluation are available on GitLab[2]. The main contributions of this paper are: (i) a UD parser for Middle High German and (ii) a thorough error analysis ensuring its utility in corpus development.

2 Related Work

The Universal Dependencies framework constitutes the theoretical basis this paper relies on. Since its initial publication by Nivre et al. (2016) it has not only become a widely accepted linguistic frame-

[1] https://stanfordnlp.github.io/stanza/
[2] https://gitlab.ruhr-uni-bochum.de/comphist/konvens-depparsing-mhg

work, but also a community project providing and developing treebanks for over 100 languages. Due to its cross-lingual consistency even across typologically diverse languages, UD treebanks have been enabling (multilingual) parser development as well as research in the field of cross-lingual learning. UD – following the tradition of dependency grammars – provides a closed set of dependency relation types, but allows for custom subtypes to incorporate special cases or specific constructions unique to one or a small set of languages. Several publications propose extensions to the original UD scheme, among which are Dipper et al. (2024) proposing a set of extensions for modern and Middle High German and providing a corpus of 1856 annotated MHG sentences, which will serve as a basis for the development of the dependency parsing model in this paper.

Low-resource NLP provides methods to counterbalance the under-representation of historic languages in quantitative and computational linguistics often being attributed to the lack of sufficient resources. Eckhoff and Berdičevskis (2016) name high variation, e.g. due to non-standardised spelling, and the overall small amount of preserved, digitised and annotated texts as difficulties when working with historical languages. They explore off-the-shelf NLP tools in pre-annotation for treebank production for Old East Slavic and show improvements in annotation speed and no interference with parsing quality when applying parsing models which were not developed specifically for the annotation task at hand. Since 2016, several efforts for developing or adapting tools to support the development of parsed corpora of historical languages have been made, among which are Sapp et al. (2023) exploring automatic constituency parsing to speed up manual annotation and correction of Early New High German. They utilise Middle Low German as a support language and develop a cross-dialectal parser for this low-resource scenario reproducing the improvement in parsing speed obtained by Eckhoff and Berdičevskis (2016). Ortmann (2020, 2021) develops and applies automatic parsing models for topological field identification and phrase recognition in historical German and partly utilises models trained on modern German for parsing historical data. The studies show that training data containing modern and historic passages improve parsing quality compared to the application of purely modern models on historic data, resembling the successful utilisation of cross-lingual training for low-resource NLP.

When researching low-resource languages, one has to not only adapt one's training techniques, but also efficiently exploit the limited amount of available data. Zupon et al. (2022) suggest a method for automatic correction of syntactic dependency annotation differences between different data sets. According to their study, it can be beneficial to automatically detect annotation mismatches between different texts or corpora and convert the mismatches before the training process begins, resulting in a technique one could call automatic curation.

3 Data

Data set	#Sent	#Tok	Annot.	Cur./Mod.
M005	513	9288	A1, A2	✓
M008	435	5836	A1, A3	
M205	480	5024	A1	
M246	11	255	A2	
M335	10	165	A1	
	251	4144	A1, A2	✓
	200	4718	A2	
M340	21	434	A1	
News	50	884	A4, A5	✓
	50	988	A4, A6	✓
Reviews	50	662	A4, A5	✓
	50	679	A4, A6	✓

Table 1: Available data sets reporting number of sentences, number of tokens and annotation as well as curation (MHG) or modification (ModG) status.

The historical data utilised in this paper were obtained from the Reference Corpus of Middle High German (ReM; Klein et al., 2016), annotated[3] according to Dipper et al. (2024) by three annotators as well as partially curated[4] and then cleaned automatically[5]. All annotated MHG data are religious texts or poetry.

The modern data originate from the German GSD treebank (McDonald et al., 2013), were automatically parsed using a modified version of the

[3] Dipper et al. (2024) propose an annotation scheme for modern and historical German, which is based on the original UD scheme for German. They achieve inter-annotator agreement of $\alpha = 0.85$.

[4] Curation of a subset of the data was done by hand by the annotators discussing diverging annotations and finding common solutions.

[5] A heuristic algorithm was run over the historical data to obtain root and period annotations, which had been left out by the annotators. Some fragmentary sentences had to be excluded from the data completely due to Stanza's inability to deal with incomplete dependency structures during parser training.

Stanford typed dependencies for English (de Marneffe et al., 2006; de Marneffe and Manning, 2008) and then corrected manually by three annotators according to the UD augmentations proposed by Dipper et al. (2024). Replacing manual curation, the modern data were modified according to the method for automatic correction of syntactic dependency annotation differences proposed by Zupon et al. (2022). Their algorithm detects (head, relation, dependent)-triples differing between text passages annotated by two annotators and produces a joint version of the text by choosing the triple with the higher overall frequency between every differing pair of triples in question.

Short name	Dev	Test	Train	#Sent	#Tok
MHG-cur	112	111	535	758	13400
MHG-all	228	229	1590	2099	32171
MHG+ModG	303	304	1692	2299	35384

Table 2: Data sets for parser development reporting number of sentences in dev, test and train set as well as total number of sentences and tokens.

Three different data sets were assembled based on the pre-processed data as shown in Table 2. **MHG-cur** consist of only the curated passages of M005 and M335. **MHG-all** unites curated as well as single-annotated MHG data and was split with regard to the principle that the test and development sets consist of only curated MHG data and the single-annotated as well as the remaining curated data are accumulated in the training set. **MHG+ModG** combines all usable data presented in Table 1 including MHG and modern data and was split equivalently to MHG-all. Note that all test sets consist of only curated MHG data as this work focuses on evaluating the parsing of MHG.

4 Methods

Stanza is an open-source library developed by Qi et al. (2020) providing a language-agnostic and data-driven NLP pipeline. It was chosen as the development tool in this paper because of its high-scoring multilingual models reported in Zeman et al. (2018) and it being well-adapted to the UD framework. For example it requires CoNLL-U formatted data and is accustomed to the annotation layers represented by the format as well as provides efficient processing for them. The factor of multi-linguality is especially important to the cross-lingual parsing of two historical stages of German conducted in this paper opposed to training a parsing model for only one language (stage). In addition to publicly available pre-trained models, Stanza provides an interface to train customised models.

The dependency model trained with Stanza[6] is an instance of a graph-based, Bi-LSTM-based deep biaffine neural dependency parser based on the Multi-Layer Perceptron approach by Kiperwasser and Goldberg (2016), augmented by Dozat and Manning (2016) with the concept of biaffine attention and finally adapted for Stanza by Qi et al. (2020). They introduce the linearisation order of two words in a given language and their typical linear distance as additional linguistically motivated features to the former model to improve parsing accuracy. The model is described as generalising well even based on small amounts of training data and is therefore well-suited for the given low-resource scenario. The developers emphasise the thorough regularisation by applying extensive dropout and the overall high performance. By default, optimisation is conducted via the Adam algorithm by Kingma and Ba (2014).

Compared to the default parameters, I set the batch size to 5000 due to technical limitations and decreased the learning rate from 0.003 to 0.002, which resulted in significantly shorter run time and higher accuracy as presented in Table 8 in the appendix.

I experimented with character- and word level embedding models provided by Stanza and pre-trained on modern German data. The evaluation showed that these embeddings do not interfere with model performance (see Appendix A), so they were included in the training of the models presented in the next section and represent another instance of modern German as a support language.

Part-of-speech tags were obtained from the original ReM annotations for all MHG data in training and evaluation and were not automatically produced by the Stanza pipeline. Annotations according to two different schemata were provided: STTS (Schiller et al., 1999) and UPOS (Petrov et al., 2011).

During training the current parsing model is evaluated on the development set after every hundredth iteration by calculating LAS, MLAS, and BLEX

[6]Training was conducted on a Linux workstation equipped with an Nvidia GeForce GTX 980 graphics card with CUDA version 12.1 and 4 GB of memory, an Intel Core i7-5820K processor and 15 GB of RAM.

(see section 5.1) with custom subtypes mapped to the original UD types. After 3000 iterations with no improvement of the LAS, the optimiser is switched from Adam to AMSGrad developed by Reddi et al. (2018). After another 3000 iterations without improvement, training is stopped automatically. The number of training steps needed for each model can be obtained in Appendix A. After training, evaluation of the parsing model is conducted on the test set, of which the results are presented in the following section.

5 Results

This section reports on evaluation scores of the parsing models trained on the three data sets presented in Section 3 as well as an error analysis of the output produced by the highest-scoring model.

5.1 Parser Evaluation

data set	UAS	LAS	CLAS	MLAS	BLEX
MHG-cur	91.99	86.30	78.58	77.37	78.58
MHG-all	91.68	85.63	77.93	76.43	77.93
MHG+ModG	92.95	88.06	81.57	80.75	81.57

Table 3: Evaluation of trained models, reporting UAS, LAS, MLAS, CLAS and BLEX in % calculated on the test set.

All metrics were calculated with the scripts from the CoNLL 2018 UD Shared Task (Zeman et al., 2018) provided by Stanza and mapping custom subtypes to their respective original UD labels.[7] The reported metrics evaluate different dimensions of a dependency parsing model. In addition to the standard metrics labelled attachment score (LAS) and unlabelled attachment score (UAS), three measures in particular relevant to the UD framework have been proposed: content word LAS (CLAS), morphology aware LAS (MLAS) and bi-lexical dependency score (BLEX). They each introduce specific aspects to a basic metric: CLAS only considers content-words when determining LAS; MLAS extends CLAS by part-of-speech tags and morphological features; BLEX scores content-word relations with lemmatisation but does not consider features and tags. Table 3 reports on the results achieved in the presented training effort. Additional results showing model training with different parameter configurations can be obtained in Table 8 in the appendix. A more fine-grained evaluation including custom subtypes of the UD labels is presented in Table 4.

data set	LAS	CLAS	MLAS	BLEX
MHG-cur	82.12	77.56	75.34	77.56
MHG-all	82.34	77.67	75.69	77.67
MHG+ModG	85.22	80.52	79.45	80.52

Table 4: Fine-grained evaluation of trained models reporting LAS, CLAS, MLAS and BLEX in % calculated on the test set and with regard to customised labels.

The results of both calculations imply a superiority of the model trained on mixed historical and modern data (MHG+ModG), referred to as the combined model from now on. Its high scores are presumably not solely due to the substantial increase in data, as the model trained on MHG-all does not score significantly higher than the model trained on MHG-cur, but more so due to the syntactical diversity present in the data, which lead to the model generalising well on unseen data. With UAS $>$ .92, LAS $>$ .88, and all reported scores $>$ .80 in Table 3, the combined model even outperforms state-of-the-art Stanza models for historical language varieties and one for modern German trained on the complete GSD treebank as presented in the Stanza documentation.[8]

5.2 Error Analysis

Stanza provides precision, recall and F1 measures for each label calculated on the test set, which are scores widely used for binary classification tasks, but which can also be applied to dependency parsing.

Table 5 presents the ten most reliably parsed labels, while a complete list of scores for each label as well as label counts on the test set can be obtained in Table 9 in the appendix. As shown there, all basic elements of a German sentence (*root*, *nsubj*, *iobj*, *obj*) reach recall scores of $\geq$ .75, so at least 75% of them are parsed correctly by the evaluated parsing model. Having the basic structure of a sentence parsed correctly in pre-annotation is very beneficial for manual correction especially due to the partly

[7] For example, Dipper et al. (2024) discriminate different subtypes of the original UD label *obl*, among which are *obl:loc* for local, *obl:dir* for directional and *obl:tmp* for temporal oblique arguments. All of these subtypes are mapped to the original label *obl* by the evaluation scripts provided by the CoNLL 2018 UD Shared Task. A more fine-grained evaluation without mapping subtypes to original labels was conducted with a modified version of the script, of which the results can be obtained in Table 4.

[8] https://stanfordnlp.github.io/stanza/performance.html (accessed May 5th 2024)

Label	Precision	Recall	F1
compound:prt	1.000	1.000	1.000
punct	0.999	1.000	0.999
case	0.982	0.986	0.984
det	0.981	0.984	0.983
amod	0.948	0.958	0.953
mark	0.954	0.948	0.951
root	0.938	0.938	0.938
cc	0.936	0.936	0.936
aux	0.938	0.920	0.929
nsubj	0.887	0.876	0.882

Table 5: Top 10 labels with highest F1 scores, reporting precision, recall, and F1 produced by the combined model on the test set.

very long and complex sentences in MHG. Presumably most important is the correct identification of the root and the subject, which is done by the parser with respective F1 scores of .936 for the root (*root*) and .882 for nominal subjects (*nsubj*). Another achievement of the parsing model lies in its ability to reliably parse frequent functional categories such as *det*, *mark*, or *cc*, which all score recall of $> .935$. Stable parsing of categories which do not usually require long consideration but are rather repetitive or even tedious for the human annotator is enormously helpful in preparation of manual annotation, as leaving this task to the hands of a parsing model enables the annotator to concentrate on the more complex decisions during annotation.

no.	Label 1	Label 2	F1
1	advmod:nmod	compound:adv	0.333
2	obl:dir	obl:loc	0.333
3	obl:loc	obl:dir	0.283
4	compound:case	compound:adv	0.222
5	advcl	ccomp	0.188
6	advmod:loc	advmod:dir	0.161
7	xcomp:pred	amod:pred	0.154
8	obl:mod	obl:arg	0.149
9	obl:loc	obl:mod	0.143
10	obl:mod	obl:loc	0.126

Table 6: Top 10 confusions of the combined model on the test set measured by F1. Recall that optimal F1 for different tags is 0.

Being conscious of the weaknesses of a parser and hence the likely errors in a pre-annotated text is important for effectively utilising the parser output in manual annotation. Secondly, the notion of frequently confused labels enables future improvement of the parsing model as new data can be annotated or corrected with special regard to these confusions. Table 6 reports on the 10 most frequent confusions of labels measured by an equivalent of an F1 score.[9] Challenging distinctions seem to lie between directional and locative oblique modifiers and adverbs, in differentiating between argument and modifier status as well as in discriminating the different subtypes of *obl* introduced by Dipper et al. (2024). These three sources of confusion within the parser output resemble in the error analysis in Dipper et al. (2024) reporting on annotation differences between two human annotators. These parallels hint to more fundamental problems than deficient training including uncertainty in meaning and valence of MHG predicates. Further research and familiarisation with these topics by the annotators resulting in higher accuracy in the training data could possibly decrease the F1 scores of these confusions.

5.3 Effects of sentence length

According to Ortmann (2021), Middle High German is known for its complex and deeply embedded syntactic structure and remarkably high variation in sentence length. Presumably, the unusual length of some sentences in the data at hand can also be explained by the text genre being mostly religious texts and poetry. The data contain sentences of up to 88 tokens, as shown in Figure 1. The test set of the combined model reflects this high variation with an average sentence length of 18.23, a median of 15 and a maximum of 88 tokens per sentence.

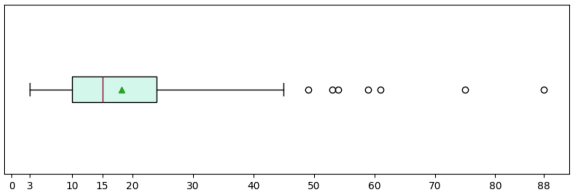

Figure 1: Distribution of sentence length in the test set of the combined model (MHG+ModG).

To gain an understanding of the effects of sentence length on the model's accuracy and to improve the parser's utility in pre-annotation, Table 7 presents the evaluation scores separate for each quantile

[9] F1 is calculated as follows:

$$2 * \frac{a_1 l_1 * a_2 l_2}{a_1 l_1 + a_2 l_2}$$

with a_1, a_2 as the annotators and l_1, l_2 as the labels annotated by the respective annotator. Possible values are between 0 and 1, where 1 means perfect agreement if $l_1 = l_2$, and 0 means perfect disagreement if $l_1 \neq l_2$. Thus, the measure corresponds to the F1 score if one of the annotators is treated as the gold standard.

of sentence length as well as for the outliers as calculated by the scripts from the CoNLL 2018 UD Shared Task (Zeman et al., 2018). As above custom subtypes have been mapped to their original UD labels. The reliability of the parser output for different sentence lengths is important for human annotators as they can decide to concentrate on those sentences with problematic lengths and hence boost efficiency of the annotation.

As can be expected, all scores peak in the first quantile with sentences consisting of three to ten tokens and are lowest in the report for the outliers including sentences with 46–88 tokens. What first attracts attention is the strikingly high UAS in the first quantile, which can be ascribed to the few opportunities for syntactic variation in short sentences and the simple syntactic structures resulting from this circumstance, including the low number of subordinate clauses, which have been presented as a source of confusion before. What is also striking is the development of all scores in between the first quantile and the outliers. Where one could have expected a rather linear decline of all scores proportional to sentence length, Table 7 shows a drop from first to second quantile followed by increasing LAS, CLAS, MLAS, and BLEX up to the fourth quantile from about three points in percentage on each score. Only the UAS is stable at around 92.5 in each of these three quantiles – it then decreases to a score of 90.66 for the outliers. This promises high structural stability of the parser output even across sentences highly varying in length.

Q	SL	UAS	LAS	CLAS	MLAS	BLEX
Q1	3–10	97.28	87.22	89.96	86.87	88.96
Q2	11–15	92.44	83.63	77.98	77.37	77.98
Q3	16–24	92.87	84.88	79.22	78.21	79.22
Q4	25–45	92.42	86.15	80.16	79.16	80.16
OL	46–88	90.66	82.46	74.00	73.00	74.00

Table 7: Evaluation scores of the sentences parsed by the combined model separately for each quantile (Q1-4) of sentence length and outliers (OL). Reported are sentence length (SL) as well as UAS, LAS, CLAS, MLAS and BLEX.

We can conclude that short sentences of up to ten tokens are parsed very reliably regarding arcs as well as labels and that the UAS and therefore the structural quality of the parsed output declines with sentence length, but that labelled scores are not as affected by token counts of up to 45. Outliers with extreme counts of up to 88 tokens have to be handled with care, but even here the parsing model is evaluated with scores of 90.66 for UAS and 82.46 for LAS, which are extraordinarily stable despite the the extreme sentence length. These insights should be kept in mind during manual correction of the parser output.

6 Discussion and Future Work

This paper presented the training and evaluation of a dependency parser of Middle High German in the Universal Dependencies framework. The highest-scoring parsing model reaches state-of-the-art results in all reported evaluation metrics and hence is a satisfactory achievement of the initial goal. As this parser is the first of its kind for MHG and only one of the few for historical languages in general, it constitutes a striking progress for the representation of historical languages in contemporary linguistic frameworks such as UD. A growing MHG treebank emerging from a reliable cycle of automatic parsing and manual correction will bring great benefit to linguistic research. That includes historic as well as diachronic research on German syntax and on the development of the German language in general. Parsing unseen data and replacing annotation from scratch with manual correction of the automatically parsed output will speed up data production and benefit treebank development. The main strengths of the presented model are its structural stability represented in high UAS scores and the reliable parsing of basic syntactic elements as well as particularly repetitive parts of the annotation task. An additional success is the utilisation of modern German as a support language for syntactically parsing low-resource MHG. This cross-lingual approach raises hopes for a joint multi-lingual parser for various stages of historical German paving the way for treebanks of all stages of historical German within the same theoretical framework. Aside from all success, the error analysis points out room for improvement on some frequently confused labels, which demonstrate problematic decisions concerning some more fundamental linguistic distinctions between argument and modifier status. Further manual annotation and correction efforts on MHG data need to be made to achieve reliable predictions concerning this question as well as expand the set of potential training data.

Further efforts on improving the parser could include a delexicalised approach to cross-lingual parsing or training customised embedding mod-

els on historical data instead of utilising the ones trained on modern German, if delexicalisation does not emerge as the method of choice. Incorporating further historic stages as represented in the reference corpus of Early New High German (ReF Wegera et al., 2021) by for example mapping the syntactic annotations of the Indiana Corpus (Sapp et al., 2023) or the Mercurius Treebank (Demske et al., 2004) to the UD schema could pave the way to a joint parsing model for different historic stages of the German language. A more practical approach for future improvements is the usage of an updated version of the utilised corpus to eliminate outdated labels as well as incorporate clarifications for the problematic distinctions within the proposed subtypes.

This work is part of the beginning of the development of a Middle High German treebank embedded in the Universal Dependencies framework. The first manual annotations published by Dipper et al. (2024) and the first parsing model published with this paper constitute the starting point of the cyclic process of treebank development to fill the void of dependency treebanks of historical German.

Limitations

Aside from all success, even the highest-scoring dependency parsing model presented in this paper has its limitations. The fine-grained error analysis presented in Section 5.2 illustrates frequent confusions and hints at likely errors present in automatically parsed data. On a larger scale, these errors reflect unresolved linguistic discussions or ambiguities as for example the distinction between argument and modifier status. Unresolved questions in contemporary research are of course represented in the data and therefore reproduced by the model, so the output has to be evaluated and utilised with regard to these conflicts.

On a higher level, automatic parsing models in their early phases – especially when trained on limited amounts of data – can not replace manual efforts. This paper made it very clear that these models are designed for pre-annotation and not for purely automatic parsing. To reach this goal, the cycle of parser and treebank development first has to be repeated time and again.

Ethical Considerations

This paper complies with the ACL Ethics Policy[10]. The development of parsing models aims at facilitating manual annotation efforts and therefore motivate further scientific research and debate. In this case, it even supports counter-balancing the underrepresentation of historical treebanks in modern frameworks. Of course it has to be kept in mind that automatic pre-annotation can reproduce biases represented in the utilised data and therefore has to be applied with care. The thorough error analysis and evaluation presented in this paper should support the sensible application of the trained models.

Acknowledgements

Special thanks go to my mentor Stefanie Dipper as well as Madeleine Landsberg-Scherff, Anna-Maria Schröter, Alexandra Wiemann and Alona Solopov, who were involved in the annotation and curation of the presented data. I am also very thankful to Adam Roussel, who helped with some technical issues and provides support for the workstation I conducted the training process on.

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – SFB 1102 – Project ID 232722074 and – SFB 1475 – Project ID 441126958. / Gefördert durch die Deutsche Forschungsgemeinschaft (DFG) – SFB 1102 – Project ID 232722074 und – SFB 1475 – Projektnummer 441126958.

References

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The Stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8, Manchester, UK. Coling 2008 Organizing Committee.

Ulrike Demske, Nicola Frank, Stefanie Laufer, and Hendrik Stiemer. 2004. Syntactic Interpretation of an Early New High German Corpus.

[10] https://www.aclweb.org/portal/content/acl-code-ethics

Stefanie Dipper, Cora Haiber, Anna Maria Schröter, Alexandra Wiemann, and Maike Brinkschulte. 2024. Universal Dependencies: Extensions for Modern and Historical German. In *LREC 2024 Conference Proceedings*.

Timothy Dozat and Christopher D. Manning. 2016. Deep biaffine attention for neural dependency parsing. *ArXiv*, abs/1611.01734.

Hanne Martine Eckhoff and Aleksandrs Berdičevskis. 2016. Automatic parsing as an efficient pre-annotation tool for historical texts. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 62–70, Osaka, Japan. The COLING 2016 Organizing Committee.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Thomas Klein, Klaus-Peter Wegera, Stefanie Dipper, and Claudia Wich-Reif. 2016. Referenzkorpus Mittelhochdeutsch (1050–1350), version 1.0.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal Dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97, Sofia, Bulgaria. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

Katrin Ortmann. 2020. Automatic topological field identification in (historical) German texts. In *Proceedings of the The 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 10–18, Online. International Committee on Computational Linguistics.

Katrin Ortmann. 2021. Automatic phrase recognition in historical German. In *Proceedings of the 17th Conference on Natural Language Processing (KONVENS 2021)*, pages 127–136, Düsseldorf, Germany. KONVENS 2021 Organizers.

Slav Petrov, Dipanjan Das, and Ryan T. McDonald. 2011. A universal part-of-speech tagset. *ArXiv*, abs/1104.2086.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Sashank J. Reddi, Satyen Kale, and Surinder Kumar. 2018. On the convergence of adam and beyond. *ArXiv*, abs/1904.09237.

Christopher Sapp, Daniel Dakota, and Elliott Evans. 2023. Parsing early New High German: Benefits and limitations of cross-dialectal training. In *Proceedings of the 21st International Workshop on Treebanks and Linguistic Theories (TLT, GURT/SyntaxFest 2023)*, pages 54–66, Washington, D.C. Association for Computational Linguistics.

Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. 1999. Guidelines für das Tagging deutscher Textcorpora mit STTS (Kleines und großes Tagset). *Technischer Bericht. Universität Stuttgart, Institut für Maschinelle Sprachverarbeitung*.

Hilkert Weddige. 2015. *Mittelhochdeutsch - Eine Einführung*, 9. edition. C. H. Beck Studium. Beck, München.

Klaus-Peter Wegera, Hans-Joachim Solms, Ulrike Demske, and Stefanie Dipper. 2021. Referenzkorpus Frühneuhochdeutsch (1350-1650), version 1.0.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.

Andrew Zupon, Andrew Carnie, Michael Hammond, and Mihai Surdeanu. 2022. Automatic correction of syntactic dependency annotation differences. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 7106–7112, Marseille, France. European Language Resources Association.

A Appendix

The following tables present more detailed evaluation scores of all trained models as well as of all labels present in the data sets. The first five rows of Table 8 illustrate parameter tuning with different combinations of learning rate and utilised pre-trained embedding models.

no.	data set	lr	emb	char	min	steps	UAS	LAS	CLAS	MLAS	BLEX
1	MHG-cur	0.003	✓	✓	41	12,100	91.74	86.65	79.48	78.19	79.48
2	MHG-cur	0.002	✓	✓	29	8,700	91.99	86.30	78.58	77.37	78.58
3	MHG-cur	0.003	✓	×	25	11,800	91.89	86.05	78.44	77.06	78.44
4	MHG-cur	0.003	×	✓	46	12,800	91.28	85.84	78.26	76.79	78.26
5	MHG-cur	0.003	×	×	20	10,100	91.99	86.40	78.86	77.48	78.86
6	MHG-all	0.003	✓	✓	53	15,200	91.68	85.63	77.93	76.43	77.93
7	MHG+ModG	0.003	✓	✓	112	30,600	92.15	86.85	79.69	78.90	79.69
8	MHG+ModG	0.002	✓	✓	50	13,700	92.95	88.06	81.57	80.75	81.57

Table 8: Evaluation scores of trained models (data sets), reporting learning rate (lr), usage of word (emb) or character (char) embeddings, run time (min), number of steps (steps) as well as UAS, LAS, MLAS, CLAS and BLEX calculated on the test set. Model 1, 6, and 8 are the ones presented in Section 5.

Label	Precision	Recall	F1	#Label	Label	Precision	Recall	F1	#Label
compound:prt	1.0000	1.0000	1.0000	15	det:predet	0.5000	1.0000	0.6667	2
punct	0.9987	1.0000	0.9994	771	parataxis	0.6154	0.6423	0.6286	137
case	0.9823	0.9858	0.9841	282	compound:pav	0.5556	0.7143	0.6250	7
det	0.9810	0.9842	0.9826	631	expl:pv	0.5333	0.7273	0.6154	11
amod	0.9482	0.9581	0.9531	191	appos	0.6053	0.5476	0.5750	42
mark	0.9538	0.9483	0.9510	174	flat	0.6667	0.5000	0.5714	4
root	0.9375	0.9375	0.9375	304	vocative	0.6098	0.5208	0.5618	48
cc	0.9355	0.9355	0.9355	124	aux:pass	0.5000	0.6364	0.5600	11
aux	0.9384	0.9195	0.9288	149	nmod:det	0.5000	0.6250	0.5556	8
nsubj	0.8874	0.8758	0.8816	612	xcomp:pred	0.6000	0.5000	0.5455	12
xcomp	0.8571	0.8824	0.8696	34	ccomp	0.5957	0.4308	0.5000	65
advmod	0.8113	0.8889	0.8483	324	obl:loc	0.4615	0.4545	0.4580	66
advmod:tmp	0.8438	0.8438	0.8438	96	expl	0.4643	0.4483	0.4561	29
cop	0.8571	0.8276	0.8421	87	dislocated	0.6667	0.3158	0.4286	19
aux:cop	0.8571	0.8182	0.8372	22	obl:compar	0.6667	0.2857	0.4000	7
discourse	0.8333	0.8333	0.8333	18	obl:dir	0.3333	0.3415	0.3373	41
nummod	1.0000	0.7143	0.8333	7	advmod:dir	0.4444	0.2353	0.3077	17
obl:tmp	0.9048	0.7600	0.8261	25	obl:arg	0.4444	0.1739	0.2500	23
nmod	0.8444	0.7308	0.7835	104	acl	0.6667	0.1053	0.1818	19
iobj	0.7711	0.7485	0.7596	171	obl	0.0	0.0	0.0	1
obj	0.6988	0.8109	0.7507	349	nmod:part	0.0	0.0	0.0	9
conj	0.7241	0.7500	0.7368	112	nmod:arg	0.0	0.0	0.0	1
acl:relcl	0.6909	0.7451	0.7170	51	csubj	0.0	0.0	0.0	5
compound:case	0.7143	0.7143	0.7143	7	orphan	0.0	0.0	0.0	3
advmod:loc	0.7778	0.6034	0.6796	58	compound:adv	0.0	0.0	0.0	2
obl:mod	0.6232	0.7350	0.6745	117	amod:pred	0.0	0.0	0.0	3
advcl	0.6220	0.7315	0.6723	108	advmod:nmod	0.0	0.0	0.0	7
hypopara	0.5000	1.0000	0.6667	1	advcl:relcl	0.0	0.0	0.0	0

Table 9: Evaluation scores of labels sorted by F1, reporting precision, recall, F1 and label count produced by the combined model (model 8) on the test set.