Chunking Historical German

Katrin Ortmann
Department of Linguistics
Fakultät für Philologie
Ruhr-Universität Bochum
ortmann@linguistics.rub.de

Abstract

Quantitative studies of historical syntax require large amounts of syntactically annotated data, which are rarely available. The application of NLP methods could reduce manual annotation effort, provided that they achieve sufficient levels of accuracy. The present study investigates the automatic identification of chunks in historical German texts. Because no training data exists for this task, chunks are extracted from modern and historical constituency treebanks and used to train a CRF-based neural sequence labeling tool. The evaluation shows that the neural chunker outperforms an unlexicalized baseline and achieves overall F-scores between 90% and 94% for different historical data sets when POS tags are used as feature. The conducted experiments demonstrate the usefulness of including historical training data while also highlighting the importance of reducing boundary errors to improve annotation precision.

1 Introduction

The analysis of linguistic phenomena in historical language requires large amounts of annotated data. For example, to study the development of syntactic phenomena like object order or extraposition in German, syntactically annotated texts from all relevant time periods are needed. To date, however, only very few historical corpora provide annotations beyond the morpho-syntactic level, thus limiting syntactic research to qualitative studies on small data sets. Using NLP methods for the automatic creation of relevant annotations could support the annotation process and reduce the necessary manual effort for quantitative studies. But the application of standard tools to historical data faces a variety of challenges, as there is less or no training data, the data is less standardized, etc.

The present study investigates the automatic recognition of chunks in historical German. Section 2 gives a short introduction to the chunking task and explains peculiarities about chunking German concerning complex pre-nominal modification. Section 3 presents previous approaches to automatic chunking, which have not yet been applied to historical data, likely because no manually annotated data is available. In this study, to address the lack of chunked historical data, chunks are extracted from modern and historical constituency treebanks. Section 4 describes the training data as well as the additional test data sets before Section 5 introduces the selected methods for automatic chunking: a regular expression-based baseline and a neural CRF chunker. Finally, Section 6 details the evaluation process and presents the results, followed by a conclusion in Section 7.

2 Chunking (German)

Chunking is also referred to as partial or shallow parsing. The concept of chunks was introduced by Abney (1991), who defines them as non-recursive phrases from a sentence's parse tree ending with the head of the phrase. According to this definition, a chunk may contain chunks of other types but not of the same type, and post-nominal modifiers start a new chunk. Example (1) shows the annotation of an English sentence following Abney's chunk definition:

(1) [S [NP The woman] [PP in [NP the lab coat]] [VP thought]] [S [NP you] [VP had bought] [NP an [ADJP expensive] book]].

(Kübler et al., 2010, p. 147)

The CoNLL-2000 shared task on chunking (Sang and Buchholz, 2000), which is still widely used as a benchmark, has popularized a more restricted definition of chunks and only allows for non-recursive, non-overlapping chunks, i.e. a word belongs to a maximum of one chunk while keeping the restriction that a chunk ends at the head token. When applied to sentence (1), this results in the annotation in example (2).

(2) [NP The woman] [PP in] [NP the lab coat] [VP thought] [NP you] [VP had bought] [NP an expensive book].

Defining chunks this way makes them suitable for the automatic annotation with sequence labeling methods and is especially useful for tasks that do not require a complete syntactic analysis but profit from an easy and fast annotation, e.g. agreement checking in word processors (Fliedner, 2002; Mahlow and Piotrowski, 2010). Furthermore, it may serve as a basis for deeper syntactic analyses (cf. Van Asch and Daelemans, 2009; Daum et al., 2003; Osenova and Simov, 2003) and thus could build the foundation for the automatic syntactic annotation of historical data.

However, applying the standard definition of chunks is problematic when chunking German because of possibly complex pre-nominal modification. The phrase in example (3) violates Abney's chunk definition due to the embedded noun chunk and, when annotated according to the CoNLL-style definition, it would contain an article *der* that is separated from its noun chunk as in example (4).

(3) [NC der [NC seinen Sohn] liebende Vater]
the his son loving father
'the father who loves his son'
(Kübler et al., 2010, p. 148)

(4) **der** [NC seinen Sohn] [NC liebende Vater]

While in some German corpora, these stranded tokens are left unannotated, e.g. DeReKo (Dipper et al., 2002), Kübler et al. (2010) introduce a special category for stranded material, marked with an initial 's', e.g. `sNC` for a stranded noun chunk. They also suggest including the head noun chunk in the prepositional chunk while leaving post-nominal modifiers separate. In the following, their approach is adopted for chunking German.

Of the eleven original chunk types from the CoNLL-2000 shared task, four main types are considered in this study: noun chunks (`NC`), prepositional chunks (`PC`), adjective chunks (`AC`), and adverb chunks (`ADVC`), and, in addition, stranded noun (`sNC`) and prepositional chunks (`sPC`). Example (5) shows the annotation of a sentence from an 1871 newspaper taken from one of the historical data sets in this study. For better readability, the relation of stranded articles to their respective noun chunks is indicated by subscripts.

(5) [sNC$_1$ die] [sNC$_2$ den] [PC an Deutschland] [NC$_2$ abgetretenen Landestheilen] [NC$_1$ angehörenden Kriegsgefangenen] [...] werden [ADVC sofort] [PC in Freiheit] gesetzt;
the the to Germany transferred territories belonging prisoners of war will be immediately to freedom set
'Prisoners of war belonging to the territories transferred to Germany will be released immediately'
Allgemeine Zeitung, no. 72, 1871
(DTA; BBAW, 2021)

3 Related Work

Since chunking can be understood as both a shallow parsing and a sequence labeling task, depending on the chunk definition, there have been many different approaches to the automatic identification of chunks. For non-recursive Abney-style chunking, Abney (1991) uses finite-state cascades, yet similar techniques have also been applied to CoNLL-style chunking. Müller (2005) gives an overview of chunking studies on German, many of which use finite state-based methods, but also other parsing approaches. For his FSA-based chunker, he reports an overall F_1-score of 96%.

For non-recursive, non-overlapping CoNLL-style chunking, there have been experiments with different classification and sequence labeling methods, including the application of taggers (e.g. Osborne, 2000; Molina and Pla, 2002; Shen and Sarkar, 2005) with F_1-scores between 92% and 94% as well as machine learning, e.g. with Conditional Random Fields yielding F_1-scores of 93% to 94% (cf. Sun et al., 2008; Roth and Clematide, 2014). More recently, there have also been experiments with neural sequence labeling using bidirectional LSTMs (Akhundov et al., 2018; Zhai et al., 2017), RNNs (Peters et al., 2017), and neural CRFs (Huang et al., 2015; Yang and Zhang, 2018) with F_1-scores of about 95%.

As chunks of a given type can only contain certain part-of-speech sequences, most of the studies use POS tags as features. However, lexicalization of models can also improve chunking results (cf. Shen and Sarkar, 2005; Indig, 2017) and current contextual word representations already seem to have some awareness of shallow syntactic structures like chunks (Swayamdipta et al., 2019). In general, van den Bosch and Buchholz (2002) find that POS tags are most relevant if the training data is small, while words become more helpful with increasing amounts of data, and a combination of both features yields the best results.

For evaluation, most studies still use the data set from the CoNLL-2000 shared task (Sang and Buchholz, 2000), i.e. WSJ data from the Penn Treebank, and written news data also serves as the evaluation basis for most studies on German. However, when Pinto et al. (2016) compare tools on English CoNLL-2000 data with their performance on Twitter data, they find that for standard toolkits, F_1-scores decrease by 17 to 38 percentage points to 45%–54% on social media text. A similar drop in performance might also occur for other non-standard data like historical language and would underline the importance of methods and models that are specifically tailored to a particular language variety.

But to date, there has only been a small number of studies on the automatic syntactic analysis of historical German, all of which have to deal with a lack of syntactically annotated historical data. In the absence of a gold standard, some studies develop rule-based approaches, e.g. Chiarcos et al. (2018) for topological field identification in Middle High German. But without the possibility for evaluation, the accuracy of such systems remains unclear. Other studies try to compensate for the lack of historical data by falling back on modern German. Petran (2012) approximates historical language by removing punctuation and capitalization from a modern German news corpus. Using CRFs, he tries to identify segments of increasing length, chunks, clauses, and sentences, in this artificial data set and concludes that smaller units are easier to identify. For chunking, he reports an F_1-score of 93.3%, but since capitalization and punctuation are not the only differences between modern and historical German, it is unclear how well these results generalize to real historical data. Nevertheless, the exploitation of modern data can be conducive for automatically annotating historical language by reducing the need for large annotated historical data sets. As a previous study has shown, models trained on modern newspaper text can successfully be transferred to historical German with F_1-scores $>92\%$ when POS tags are used as input unless the historical language structures differ too much from modern German (Ortmann, 2020).

4 Data

As already mentioned, most German corpora and especially historical corpora do not offer a manual chunk annotation that could be used for training and evaluating automatic models. However, Kübler et al. (2010) notice that chunks can be extracted directly from constituency trees by converting the lowest phrasal projections with lexical content to chunks. Using this method, they automatically transform the constituency annotations from the TüBa-D/Z treebank (Telljohann et al., 2017) into chunks. The resulting corpus[1] comprises 3,816 newspaper articles with more than 100k sentences and almost 2M tokens. In total, it contains over 743k instances of the six chunk types considered in the present study.

Since the extracted chunks might be influenced by the structure of the constituency trees and, hence, may differ between treebanks with different syntactic annotation schemes, a second German treebank is included in the present study. The Tiger corpus (Brants et al., 2004)[2] contains about 50k sentences with about 888k tokens from 2,263 German news articles, but the annotation of certain syntactic phenomena deviates significantly from those in the TüBa-D/Z corpus (Dipper and Kübler, 2017). Most notably, the Tiger treebank includes discontinuous annotations. Therefore, all sentences must be linearized first[3] before chunks of the six different types can be extracted from the constituency trees similar to the procedure described by Kübler et al. (2010).

Besides accounting for possible influences of the annotation scheme on the extracted chunks, including the Tiger treebank offers another advantage: While annotated historical data sets rarely exist for syntactic annotation tasks, there are two

[1] Release 11.0, chunked version, http://www.sfs.uni-tuebingen.de/ascl/ressourcen/corpora/tueba-dz.html

[2] Version 2.2, TIGER-XML format, https://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/tiger

[3] As only the lowest phrasal projections are used to derive chunks from the tree, the broader structure of the tree is irrelevant for the task at hand. Therefore, discontinuous nodes are simply duplicated and re-inserted at the correct position inside the tree according to the linear order of terminal nodes in the sentence.

Corpus	#Docs	#Sents	#Toks	#Chunks
Training				
TüBa-D/Z	3,075	83,225	1,564,840	593,735
Tiger	1,863	39,976	726,811	255,077
Mercurius	2	6,709	150,354	53,831
ReF.UP	26	16,761	415,934	163,438
Development				
TüBa-D/Z	377	10,702	196,308	74,780
Tiger	200	4,567	81,593	28,615
Mercurius	2	820	18,287	6,570
ReF.UP	26	2,112	53,836	21,245
Test				
TüBa-D/Z	364	10,491	196,636	74,982
Tiger	200	4,445	78,018	27,253
Modern	78	547	7,605	2,829
Mercurius	2	818	18,740	6,691
ReF.UP	26	2,173	54,005	21,120
HIPKON	53	342	4,210	1,529
DTA	29	606	18,515	6,651

Table 1: Overview of the data sets. The number of chunks refers to the six chunk types evaluated in this study. Only sentences containing at least one chunk of the given types are included.

treebanks for historical German, which are annotated according to the Tiger scheme and thus, fortunately, can also be used for chunk extraction. The Mercurius corpus (Demske, 2005)[4] contains semi-automatic annotations of approximately 8k sentences with 187k tokens from newspaper text from the 16^{th} and 17^{th} centuries. The second treebank, ReF.UP, is a subcorpus of the Reference Corpus of Early New High German (Wegera et al., 2021)[5] and includes annotations of 26 documents with 21k sentences and 500k tokens from different language areas from the 14^{th} to 17^{th} century. Like with the Tiger corpus, the constituency trees from both historical treebanks must be linearized before chunks can be extracted from them. In total, the two corpora contain about 67k chunks and over 205k chunks of the six relevant types, respectively. While the Tiger corpus is already provided with a training, development, and test section, the other three corpora were split into a training (80%), development (10%), and test set (10%) for this study. Also, the historical POS tagsets in the Mercurius and ReF.UP treebanks were mapped to the German standard tagset STTS (Schiller et al., 1999).

Compared to previous studies on historical data, the two modern and historical treebanks form a solid basis for training and evaluating automatic chunking methods on historical German. However, Osborne (2002) notes that distributional differences between training and test data can be even more problematic for chunking performance than noise in the data itself. Therefore, three additional data sets from a previous study (Ortmann, 2020),[6] which are unrelated to the training data, are used for evaluation. The first data set is a collection of about 550 sentences with 7.6k tokens from five modern registers with a varying degree of formality: Wikipedia articles, fiction texts, Christian sermons, TED talk subtitles, and movie subtitles. In total, the modern data set contains about 2.8k chunks of the six types and is used to test the applicability of annotation methods to non-newspaper registers.

The two other data sets comprise historical data from two different corpora. The HIPKON corpus (Coniglio et al., 2014) contains 342 manually annotated sentences from 53 sermons from the 12^{th} to the 18^{th} century. Originally, the corpus only includes a partial annotation of chunks, which was completed for the present study. Also, the mapping of the historical POS tags to STTS tags from Ortmann (2020) was used. The second historical data set consists of 600 sentences with 18.5k tokens from 29 texts from the German Text Archive DTA (BBAW, 2021). The texts were published in a variety of genres[7] from the 16^{th} to the 20^{th} century and were manually enriched with chunks for this study, using the corrected POS tags and sentence boundaries from Ortmann (2020). Table 1 gives an overview of the data sets. The annotated data sets and additional resources can be found in this paper's repository.[8]

Table 2 shows the distribution of the six chunk types in the test data. As could be expected, noun chunks (`NC`) are the most frequent chunk type, followed by prepositional chunks (`PC`) and adverb chunks (`ADVC`). Stranded chunks make up about 1% of the chunks in all data sets, except for the TüBa-D/Z data with 0.6% and the modern non-standard data with only 0.4% stranded chunks. While stranded noun chunks (`sNC`) are more frequent in the modern data, the opposite can be observed for most of the historical data sets where

[4] Mercurius Baumbank (version 1.1), https://doi.org/10.34644/laudatio-dev-VyQiCnMB7CArCQ9CjF3O

[5] https://www.linguistics.rub.de/ref

[6] https://github.com/rubcompling/latech2020

[7] The DTA subset contains five newspaper texts and three texts each from the genres: funeral sermon, language science, medicine, gardening, theology, chemistry, law, and prose.

[8] https://github.com/rubcompling/nodalida2021

Corpus	NC	PC	AC	ADVC	sNC	sPC
TüBa-D/Z	54.2	24.6	5.9	14.8	0.4	0.2
Tiger	55.2	30.7	4.6	8.5	0.6	0.4
Modern	60.3	21.2	5.5	12.5	0.3	0.1
Mercurius	51.5	29.5	4.4	13.5	0.4	0.7
ReF.UP	57.7	20.6	5.9	15.1	0.2	0.5
HIPKON	56.4	25.1	2.4	15.3	0.1	0.9
DTA	56.4	24.4	5.2	12.8	0.6	0.6

Table 2: Distribution of chunk types in the test data reported as percentage of the total number of chunks per data set.

stranded prepositional chunks (sPC), as in example (6) from the Mercurius corpus, are more common.

(6) [sPC von] [NC der Frantzosen] [PC Vorhaben]
of the French's plan
'of the plan of the French'

5 Methods

As detailed in Section 3, various methods have been applied to the automatic recognition of chunks in modern text. In the present study, two different approaches are tested: an unlexicalized regular expression-based chunker, which serves as a baseline, and a neural state-of-the-art sequence labeling tool.

The regular expression-based approach is comparable to the finite-state chunkers mentioned in Section 3. For this study, a simple RegExp chunker as implemented in the NLTK[9] is used, which successively applies a set of manually created context-sensitive regular expressions to an input POS sequence to identify non-recursive, non-overlapping chunks of the six different types.

The neural sequence labeling tool NCRF++ (Yang and Zhang, 2018)[10] achieves state-of-the-art results for several tasks, including chunking. On the English CoNLL-2000 data, the best model reaches an F_1-score of 95% (Yang et al., 2018). The toolkit consists of a three-layer architecture with a character sequence layer, a word sequence layer, and a CRF-based inference layer. While the RegExp chunker relies on expert knowledge in the form of manually compiled rules, NCRF++ must be trained on annotated data to perform the task. For this study, the tool is trained on the two different modern treebanks: model News1 is trained on the TüBa-D/Z training set, and model News2 on the Tiger training set. Also, the two historical treebanks are used to train a joined model Hist, which might be more suitable for the analysis of historical data and its peculiarities. Finally, since the historical data sets are smaller than the modern training sets, a model News2+Hist is trained on a combination of the modern and historical treebanks that follow the same annotation scheme. During training, the tool is provided with the corresponding development data and each of the models is trained with and without POS tags as an additional feature. Since current contextual word representations seem to be aware of shallow syntactic structures (Swayamdipta et al., 2019), each model is also trained with GloVe embeddings pretrained on German Wikipedia.[11] To ensure comparability, all models are trained with the same default settings.[12] While the News2 and Hist training sets only contain annotations of the six chunk types considered in this study, the News1 model is trained on all chunk types included in the TüBa-D/Z corpus, although only the six types described in Section 2 are evaluated here. For each token, both selected methods, i.e. the RegExp chunker and the NCRF++ toolkit, output the single most likely chunk label encoded as a BIO tag.

6 Evaluation and Results

To assess the performance of the automatic methods introduced in the previous section, their output is compared to the gold standard annotation. As already mentioned, every token is annotated with a BIO tag, i.e. either B-XC (beginning of chunk), I-XC (inside chunk), or O (outside chunk). However, the number of tokens inside and outside of chunks provides little information about the quality of the automatic chunk annotation. Instead, it is of interest whether the boundaries of chunks align between automatic and gold annotation. Therefore, the evaluation is carried out chunk-wise instead of token-wise and each chunk in the gold

[9] http://www.nltk.org/api/nltk.chunk.html

[10] https://github.com/jiesutd/NCRFpp

[11] GloVe embeddings trained on German Wikipedia and provided by deepset, https://deepset.ai/german-word-embeddings

[12] The experiments of Yang et al. (2018) suggest that the default combination of character CNN, word LSTM, and a CRF-based inference layer gives the best result for the chunking task with good model stability for random seeds (mean F_1: 94.86 ± 0.14). However, the present study is only a first investigation of chunking historical German and further experiments should be conducted to test for model stability and to explore fine-tuning of parameters for optimal results.

Model	*Words*	*POS*	*GloVe*	**TüBa-D/Z**	**Tiger**	**Modern**	**Mercurius**	**ReF.UP**	**HIPKON**	**DTA**
`RegExp`	-	+	-	85.46	86.75	90.35	85.70	86.83	91.76	88.20
`News1`	+	-	-	93.46	87.80	89.63	72.52	49.77	47.69	72.07
	+	-	+	94.30	88.16	90.12	73.48	51.94	48.43	71.50
	+	+	-	97.07	90.33	92.91	90.34	91.01	93.71	90.11
	+	+	+	**97.17**	90.89	93.68	90.37	90.66	92.92	90.15
`News2`	+	-	-	85.02	91.41	86.67	71.15	49.09	43.25	67.75
	+	-	+	86.19	92.76	87.77	72.05	50.01	46.90	69.59
	+	+	-	90.96	94.70	**94.04**	88.58	89.84	94.20	88.76
	+	+	+	91.22	**95.44**	93.97	88.55	88.77	92.50	88.35
`Hist`	+	-	-	n.a.	n.a.	n.a.	11.68	16.10	12.81	13.86
	+	-	+	n.a.	n.a.	n.a.	85.53	81.28	69.41	73.61
	+	+	-	n.a.	n.a.	n.a.	92.37	93.48	93.29	89.89
	+	+	+	n.a.	n.a.	n.a.	**92.80**	**93.64**	93.85	**90.37**
`News2` `+Hist`	+	-	-	n.a.	n.a.	n.a.	82.56	79.42	60.47	73.24
	+	-	+	n.a.	n.a.	n.a.	83.40	79.02	65.05	74.77
	+	+	-	n.a.	n.a.	n.a.	91.94	93.03	**94.49**	90.15
	+	+	+	n.a.	n.a.	n.a.	92.19	93.41	93.99	90.29

Table 3: Overall F_1-scores for the RegExp chunker and all NCRF++ models for the seven corpora. Models trained on historical data are only applied to historical corpora. All numbers are given in percent and the best result for each corpus is highlighted in bold.

standard is compared to the system output and vice versa concerning chunk type and chunk boundaries. Only sentences for which the gold standard contains at least one of the six relevant chunk types are considered. Chunks with identical labels and boundaries are counted as true positives, whereas chunks only existing in the gold standard are considered false negatives, and chunks only present in the system output count as false positives.

In addition to these common categories, there can be additional types of errors, though, which are not captured by the three categories and usually are penalized as multiple errors in a single unit. For example, a system could identify a chunk spanning the correct token sequence but label it as a different chunk type, e.g. `ADVC` instead of `AC`, which would count as a false positive `ADVC` and a false negative `AC`. Also, a system can get the boundaries of a chunk wrong, e.g. miss the first word of an `ADVC`, which would correspond to a false positive and a false negative `ADVC`. And finally, the system can make both errors at once, for example by missing the initial preposition and classifying a `PC` as `NC`, resulting in a false positive `NC` and a false negative `PC`. To account for these types of errors, in the following, seven different categories are distinguished during evaluation: true positives (`TP`), false positives (`FP`), labeling errors (`LE`), boundary errors (`BE`), labeling-boundary errors (`LBE`), and false negatives (`FN`).[13] Because labeling and boundary errors mean that the system recognized some chunk, although not entirely correctly, and not that it missed a chunk, `LE`, `BE`, and `LBE` errors are counted as false positives for the calculation of precision and recall while preventing multiple penalties for a single unit. As the evaluation is carried out chunk-wise, sensible true negatives cannot be determined and are not evaluated here. Table 3 gives an overview of the results for the different annotation methods and models.

The evaluation shows that the RegExp parser, which operates on POS tags only, reaches F_1-scores between 85% and 92% for all data sets, setting a high baseline for the task. The best results are achieved for the modern non-newspaper data and the HIPKON corpus. The NCRF++ models outperform this baseline by several percentage points on each data set, achieving F_1-scores between 90% and 97%. The recall lies between

[13] The idea for this distinction between error types stems from a blog post by Chris Manning about a similar problem with named entity evaluation (`https://nlpers.blogspot.com/2006/08/doing-named-entity-recognition-dont.html`). The problem with double penalties when using F-scores has also been recognized in the literature. For example, in the context of word tokenization, Shao et al. (2017) show that precision favors under-splitting systems, suggesting that recall, i.e. the proportion of correctly segmented units, gives a more realistic impression of system performance and should be used as the only evaluation metric. However, for tasks that require segmentation and labeling such as chunking or NER, almost correct chunks/entities may still provide useful information for certain purposes. Thus, the more fine-grained distinction of errors and adjusted calculation of precision and recall seem appropriate for a thorough evaluation of these annotations.

97% and 99% for the best models on all data sets and is always higher than the precision with 84% to 95%. As already observed in other studies (van den Bosch and Buchholz, 2002), models that include POS as additional features generally perform better than models purely based on characters and word forms. Also, adding pre-trained word embeddings improves the results in almost all cases, especially for models without POS tags.

The modern newspaper data is analyzed with the highest F_1-scores of 97% and 95% respectively. Unsurprisingly, models trained on the training section of the same corpus perform better on the test data than models trained on another data set. This may be a result of distributional differences between data sets (Osborne, 2002) but could, in part, also be due to differences between the constituency trees from which the chunks were extracted.

The results for the modern non-newspaper data are slightly lower than for the news corpora with a maximum F_1-score of 94%. Interestingly, the overall F_1-scores are higher for the more informal registers than for the formal ones. Probably, informal sentences are generally easier to chunk because they contain more simple (noun) chunks and less pre-nominal modification.

While models purely based on words still perform well on the modern data, POS tags prove to be especially relevant for the historical data. Even the `Hist` model must be complemented with (modern) pre-trained word embeddings for acceptable performance on the historical corpora, possibly reflecting problems with the non-standardized spelling in historical German. For the Mercurius and ReF.UP corpora, the `Hist` model with POS and word embeddings achieves the best results with F_1-scores of about 93%, followed by the `News2+Hist` model. For the HIPKON corpus, the `News2+Hist` model with POS reaches the highest F_1-score of 94.5%, closely followed by the `News2` model. The DTA data is analyzed with the highest F_1-score of 90.4% by the `Hist` model with POS and word embeddings, followed by the `News2+Hist` and the `News1` models with F_1-scores of about 90% as well.

These results are in line with the observations of Ortmann (2020) that models trained on modern news data can successfully be transferred to historical German with overall F_1-scores >90% when POS tags are used as input. However, the evaluation also shows that historical training data further improves the automatic annotation of historical language.[14]

Corpus	NC	PC	AC	ADVC	sNC	sPC
TüBa-D/Z	95.6	97.9	86.8	97.0	77.2	70.0
Tiger	94.4	95.0	85.2	84.7	84.7	68.4
Modern	93.3	91.3	85.4	83.7	80.0	0.0
Mercurius	90.6	90.8	84.3	86.0	0.0	36.7
ReF.UP	92.9	92.3	81.1	85.1	5.6	40.3
HIPKON	94.1	90.4	87.0	87.4	0.0	26.7
DTA	87.5	90.0	80.4	81.8	10.3	16.7

Table 4: Overall F_1-scores per chunk type (in percent) for the best performing model on each data set.

In Table 4, the results per chunk type are displayed for the best performing model on each data set. Here, no distinction is made between true positives, labeling, and boundary errors, i.e. one unit can correspond to multiple errors in one or two of the categories as exemplified above. For all data sets, the best results are observed for noun and prepositional chunks with F_1-scores mostly above 90%, while the results for adjective and adverb chunks range mostly between 80% and 87%. The stranded chunk types are recognized much less reliably, especially in the historical data where the majority of errors in these categories result from structures with a pre-nominal modifying noun chunk `NC` inside a prepositional chunk `PC` like in example (6) above. These structures are more frequent in historical German, causing the higher proportion of stranded prepositional chunks compared to modern data. When confronted with a structure like this, in most cases, instead of annotating a stranded preposition `sPC` preceding a pre-nominal noun chunk `NC`, the models identify a joined `PC`, followed by an `NC` as in example (7).

[14] It is important to note that the experiments in this paper were conducted with gold standard POS tags and using automatically assigned POS can be expected to negatively influence the results. For example, Müller (2005) reports a chunking F_1-score of only 90% instead of 96% when using automatic POS. Applying the Stanza tagger (Qi et al., 2020, German `hdt` model) to the modern data sets in this study results in POS error rates of 4% (TüBa-D/Z) to 6% (Modern) and reduces the F_1-scores of the RegExp chunker by 1 (TüBa-D/Z) to 4 (Modern) percentage points. The F_1-scores of the best NCRF++ models with POS as feature decrease by 3 (TüBa-D/Z) to 3.7 (Tiger, Modern) percentage points. It can be assumed that similar reductions would be observed for historical data if a comparable tagger model for the relevant language stages was available and used to tag the data automatically.

Corpus	FP	LE	BE	LBE	FN
TüBa-D/Z	10.9	4.4	60.1	4.8	19.8
Tiger	17.7	6.0	59.8	4.2	12.3
Modern	11.3	5.5	63.6	2.4	17.1
Mercurius	22.6	10.3	53.2	7.1	6.7
ReF.UP	17.5	8.1	56.0	7.1	11.3
HIPKON	11.7	10.4	55.8	12.3	9.8
DTA	13.9	6.5	58.9	6.7	14.0

Table 5: Proportion of the five different error types: false positives (FP), labeling errors (LE), boundary errors (BE), labeling-boundary errors (LBE), and false negatives (FN). Numbers are given in percent for the best performing model on each data set.

(7) **Gold:** [sPC von] [NC der Frantzosen] [PC Vorhaben]

NCRF++: [PC von der Frantzosen] [NC Vorhaben]

Since, in these cases, the embedded noun chunk cannot be recognized based on STTS POS tags, a morphological analysis is necessary to distinguish structures with a pre-nominal genitive from prepositional chunks with a post-modifying noun chunk. When the genitive form is not syncretized, i.e. the word form differs from the morphological realization in other cases like nominative or dative, lexicalized models could, in theory, identify the correct structure. But as stranded chunks constitute only about one percent of all chunks in the data sets, there is not enough training data to recognize them reliably.

Finally, Table 5 shows the distribution of error types in the data sets, including the more fine-grained distinction of labeling and boundary errors. Interestingly, for all corpora, boundary errors constitute more than half of the errors, i.e. the models identified the chunks but did not achieve an exact match of the boundaries. One could argue that this type of error is less severe than completely missing (FN) or made-up chunks (FP), which are the second and third most frequent error types for most data sets. The evaluation approach in this study, which does not multiply penalize a model for boundary errors, thus seems appropriate to get a more realistic impression of model performance.

7 Conclusion

The present study has investigated the automatic recognition of chunks in historical German. To address the main problem of analyzing historical language, namely a lack of manually annotated data for training and evaluation, chunks of six different types were derived from modern and historical constituency treebanks. Using the extracted chunks, the state-of-the-art neural sequence labeling tool NCRF++ was trained on modern news articles, Early New High German corpora, as well as a combination of modern and historical data.

The evaluation has shown that models that include POS tags as features can be transferred successfully from modern to historical language, with F_1-scores >90%, thereby outperforming a regular expression-based baseline. By adding historical training data, the results can be improved further, yielding F_1-scores between 90.4% and 94.5% for the different historical corpora.

Regarding the evaluation of chunks, the present study has argued for a distinction between different types of errors that are commonly penalized as multiple errors in a single unit. An analysis of the occurring error types showed that the majority of errors are boundary errors, meaning that the system identified the chunks, but the boundaries do not exactly match those in the gold standard. Since this type of error can be considered less severe than pure false positives or negatives, the presented results give a more realistic impression of the actual system performance.

Future studies should focus primarily on a reduction of incorrect chunk boundaries to increase the annotation precision, as well as further investigate and improve the analysis of stranded chunks and complex pre-nominal modification in (historical) German.

Acknowledgments

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 232722074 – SFB 1102 (Project C6). I am grateful to the student annotators Anna Maria Schroeter and Larissa Weber for the annotations and Jennifer Wodrich for help with the various data sets. Also, I would like to thank the anonymous reviewers for their helpful comments.

References

Steven P. Abney. 1991. Parsing by chunks. In Robert C. Berwick, Steven P. Abney, and Carol Tenny, editors, *Principle-based parsing*, volume 44 of *Studies in Linguistics and Philosophy*, pages 257–278. Springer.

Adnan Akhundov, Dietrich Trautmann, and Georg Groh. 2018. Sequence labeling: A practical approach. *arXiv preprint arXiv:1808.03926*.

BBAW. 2021. Deutsches Textarchiv. Grundlage für ein Referenzkorpus der neuhochdeutschen Sprache. Berlin-Brandenburgische Akademie der Wissenschaften.

Antal van den Bosch and Sabine Buchholz. 2002. Shallow parsing on the basis of words only: a case study. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 433–440.

Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. 2004. TIGER: Linguistic interpretation of a German corpus. *Research on language and computation*, 2(4):597–620.

Christian Chiarcos, Benjamin Kosmehl, Christian Fäth, and Maria Sukhareva. 2018. Analyzing Middle High German syntax with RDF and SPARQL. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Marco Coniglio, Karin Donhauser, and Eva Schlachter. 2014. HIPKON: Historisches Predigtenkorpus zum Nachfeld (Version 1.0). Humboldt-Universität zu Berlin. SFB 632 Teilprojekt B4.

Michael Daum, Kilian A. Foth, and Wolfgang Menzel. 2003. Constraint based integration of deep and shallow parsing techniques. In *10th Conference of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary.

Ulrike Demske. 2005. Mercurius-Baumbank (Version 1.1). Universität Potsdam.

Stefanie Dipper, Hannah Kermes, Dr. Esther König-Baumer, Wolfgang Lezius, Frank H. Müller, and Tylman Ule. 2002. DEREKO (DEutsches REferenzKOrpus) German Reference Corpus Final Report (Part I).

Stefanie Dipper and Sandra Kübler. 2017. German treebanks: TIGER and TüBa-D/Z. In Nancy Ide and James Pustejovsky, editors, *Handbook of linguistic annotation*, pages 595–639. Springer.

Gerhard Fliedner. 2002. A system for checking NP agreement in German texts. In *Proceedings of the ACL Student Research Workshop*, pages 12–17.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Balázs Indig. 2017. Less is more, more or less... Finding the optimal threshold for lexicalization in chunking. *Computación y Sistemas*, 21(4):637–646.

Sandra Kübler, Kathrin Beck, Erhard Hinrichs, and Heike Telljohann. 2010. Chunking German: an unsolved problem. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 147–151, Uppsala, Sweden. Association for Computational Linguistics.

Cerstin Mahlow and Michael Piotrowski. 2010. Noun phrase chunking and categorization for authoring aids. In *10. Konferenz zur Verarbeitung Natürlicher Sprache (KONVENS 2010)*. University of Zurich.

Antonio Molina and Ferran Pla. 2002. Shallow parsing using specialized HMMs. *The Journal of Machine Learning Research*, 2:595–613.

Frank Henrik Müller. 2005. *A finite-state approach to shallow parsing and grammatical functions annotation of German*. Ph.D. thesis, Seminar für Sprachwissenschaft, Universität Tübingen.

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.

Miles Osborne. 2000. Shallow parsing as part-of-speech tagging. In *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*, pages 145–147.

Miles Osborne. 2002. Shallow parsing using noisy and non-stationary training material. *The Journal of Machine Learning Research*, 2(Mar):695–719.

Petya Osenova and Kiril Simov. 2003. Between chunk ideology and full parsing needs. In *Proceedings of the Shallow Processing of Large Corpora (SProLaC 2003) Workshop*, pages 78–87.

Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. *arXiv preprint arXiv:1705.00108*.

Florian Petran. 2012. Studies for segmentation of historical texts: Sentences or chunks? In *Proceedings of the Second Workshop on Annotation of Corpora for Research in the Humanities (ACRH-2)*, pages 75–86.

Alexandre Pinto, Hugo Gonçalo Oliveira, and Ana Oliveira Alves. 2016. Comparing the performance of different NLP toolkits in formal and social media text. In *5th Symposium on Languages, Applications and Technologies (SLATE'16)*, pages 3:1–3:16. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

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*.

Luzia Roth and Simon Clematide. 2014. Tagging complex non-verbal German chunks with Conditional Random Fields. In *Proceedings of the 12th Edition of the KONVENS Converence*, pages 48–57, Hildesheim, Germany. University of Zurich.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task: Chunking. In *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*, pages 127–132.

Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. 1999. Guidelines für das Tagging deutscher Textcorpora mit STTS (Kleines und großes Tagset).

Yan Shao, Christian Hardmeier, and Joakim Nivre. 2017. Recall is the proper evaluation metric for word segmentation. In *Proceedings of the The 8th International Joint Conference on Natural Language Processing*, pages 86–90, Taipei, Taiwan.

Hong Shen and Anoop Sarkar. 2005. Voting between multiple data representations for text chunking. In Balázs Kégl and Guy Lapalme, editors, *Advances in Artificial Intelligence. Canadian AI 2005.*, pages 389–400. Springer.

Xu Sun, Louis-Philippe Morency, Daisuke Okanohara, Yoshimasa Tsuruoka, and Jun'ichi Tsujii. 2008. Modeling latent-dynamic in shallow parsing: a latent conditional model with improved inference. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 841–848, Manchester, UK.

Swabha Swayamdipta, Matthew Peters, Brendan Roof, Chris Dyer, and Noah A. Smith. 2019. Shallow syntax in deep water. *arXiv preprint arXiv:1908.11047*.

Heike Telljohann, Erhard W. Hinrichs, Sandra Kübler, Heike Zinsmeister, and Kathrin Beck. 2017. *Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z)*. Seminar für Sprachwissenschaft, Universität Tübingen, Germany.

Vincent Van Asch and Walter Daelemans. 2009. Prepositional phrase attachment in shallow parsing. In *Proceedings of the International Conference RANLP-2009*, pages 12–17. Association for Computational Linguistics.

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

Jie Yang, Shuailong Liang, and Yue Zhang. 2018. Design challenges and misconceptions in neural sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, pages 3879–3889, Santa Fe, New Mexico, USA.

Jie Yang and Yue Zhang. 2018. NCRF++: An open-source neural sequence labeling toolkit. In *Proceedings of ACL 2018, System Demonstrations*, pages 74–79, Melbourne, Australia.

Feifei Zhai, Saloni Potdar, Bing Xiang, and Bowen Zhou. 2017. Neural models for sequence chunking. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3365–3371.