Bochumer
Linguistische
Arbeitsberichte
25

Computational Methods for Investigating Syntactic Change: Automatic Identification of Extraposition in Modern and Historical German

Katrin Ortmann

Bochumer Linguistische Arbeitsberichte

Herausgeberin: Stefanie Dipper

Die online publizierte Reihe „Bochumer Linguistische Arbeitsberichte“ (BLA) gibt in unregelmäßigen Abständen Forschungsberichte, Abschluss- oder sonstige Arbeiten der Bochumer Linguistik heraus, die einfach und schnell der Öffentlichkeit zugänglich gemacht werden sollen. Sie können zu einem späteren Zeitpunkt an einem anderen Publikationsort erscheinen. Der thematische Schwerpunkt der Reihe liegt auf Arbeiten aus den Bereichen der Computerlinguistik, der allgemeinen und theoretischen Sprachwissenschaft und der Psycholinguistik.

The online publication series “Bochumer Linguistische Arbeitsberichte” (BLA) releases at irregular intervals research reports, theses, and various other academic works from the Bochum Linguistics Department, which are to be made easily and promptly available for the public. At a later stage, they can also be published by other publishing companies. The thematic focus of the series lies on works from the fields of computational linguistics, general and theoretical linguistics, and psycholinguistics.

Band 25 (February 2023)

Herausgeberin: Stefanie Dipper
Sprachwissenschaftliches Institut
Ruhr-Universität Bochum
Universitätsstr. 150
44801 Bochum

Erscheinungsjahr 2023
ISSN **2190-0949**

Katrin Ortmann

Computational Methods for Investigating Syntactic Change: Automatic Identification of Extraposition in Modern and Historical German

---

2023

Bochumer Linguistische Arbeitsberichte

(BLA 25)

Computational Methods for Investigating Syntactic Change: Automatic Identification of Extraposition in Modern and Historical German

Inaugural-Dissertation

zur

Erlangung des Grades eines Doktors der Philosophie

in der

Fakultät für Philologie

der

RUHR-UNIVERSITÄT BOCHUM

vorgelegt

von

Katrin Ortmann

Gedruckt mit der Genehmigung der Fakultät für Philologie der Ruhr-Universität Bochum

Referent:	Frau Prof. Dr. Stefanie Dipper
Korreferent:	Frau Prof. Dr. Heike Zinsmeister
Tag der mündlichen Prüfung:	01.02.2023

Abstract

The linguistic analysis of historical German and diachronic syntactic change is traditionally based on small, manually annotated data sets. As a consequence, such studies lack the generalizability and statistical significance that quantitative approaches can offer. In this thesis, computational methods for the automatic syntactic analysis of modern and historical German are developed, which help to overcome the natural limits of manual annotation and enable the creation of large annotated data sets. The main goal of the thesis is to identify extraposition in modern and historical German, with extraposition being defined as the movement of constituents from their base position to the post-field of the sentence (Höhle 2019; Wöllstein 2018).

For the automatic recognition of extraposition, two annotation steps are combined: (i) a topological field analysis for the identification of post-fields and (ii) a constituency analysis to recognize candidates for extraposition. The thesis describes experiments on topological field parsing (Ortmann 2020), chunking (Ortmann 2021a), and constituency parsing (Ortmann 2021b). The best results are achieved with statistical models trained on Part-of-Speech tags as input. Contrary to previous studies, all annotation steps are thoroughly evaluated with the newly developed *FairEval* method for the fine-grained error analysis and fair evaluation of labeled spans (Ortmann 2022). In an example analysis, the created methods are applied to large collections of modern and historical text to explore different factors for the extraposition of relative clauses, demonstrating the practical value of computational approaches for linguistic studies.

The developed methods are released as the CLASSIG pipeline (**C**omputational **L**inguistic **A**nalysis of **S**yntactic **S**tructures **I**n **G**erman) at `https://github.com/rubcompling/classig-pipeline`. Data sets, models, and evaluation results are provided for download at `https://github.com/rubcompling/classig-data` and `https://doi.org/10.5281/zenodo.7180973`.

Acknowledgments

When I began my studies at Ruhr-University Bochum, I had no idea that one day I would be sitting here, writing my dissertation. What a journey it has been! And it is for sure that the time wouldn't have been as much fun without the amazing people I met. Thank you, Ronja, Helena, Anna, Doreen, Pia, and Julia for joining me on this way.

Thanks also to my colleagues and former colleagues from Bochum, Adam, Ilka, and my fantastic student assistants Anna Maria, Jenni, Larissa, and Madeleine. Your support made this work possible.

A special thank you goes to Sophia. It has been a pleasure to share the virtual office and yoga mat with you! You made it so easy to forget the 300 kilometers between us.

A big thank you to my supervisor Stefanie Dipper for all the opportunities, the positive working atmosphere, and for giving me the freedom to 'do my thing'.

I am also particularly grateful to the wonderful people from Saarbrücken and the SFB, Augustin Speyer, Elke Teich, Marie-Ann, Patricia, and Sabine. Thank you to Heike Zinsmeister for reviewing my thesis. And thanks also to Pauline, Stefania, Lisa, Tom, Iona, and everyone I had the pleasure to connect with in the past years.

Last but not least, I want to thank my family, who always had my back. Without you, I wouldn't be here today. Thank you so much!

My work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 232722074 – SFB 1102 (Project C6).

Contents

Figures

Tables

Chapter 1

Introduction

Human language is not a static construct but rather is characterized by continuous, dynamic change on all linguistic levels, from phonetics and phonology to morphology, lexical and semantic change to syntactic structures (Bybee 2015). The present thesis emerged from a project that deals with the latter, namely the investigation of syntactic variation in the history of German. In particular, the focus was on the diachronic development of extraposition. By this, we mean the phenomenon of constituents being moved from their base position in the middle field to the post-field of the sentence behind the right sentence bracket. Consider the following examples. Each sentence (1–3a) contains a different type of constituent that is placed in the post-field (Höhle 2019; Wöllstein 2018; sentence brackets are marked in boldface, and the extraposed constituents are underlined). In (1–3b), the same constituents are placed in their base position in the middle field.

(1) a. **Hör** endlich **auf** mit dem Quatsch!

b. **Hör** endlich mit dem Quatsch **auf**!
'Stop the nonsense!'

(2) a. Es **ist** echt schön **gewesen** gestern.

b. Es **ist** gestern echt schön **gewesen**.
'Yesterday was really nice.'

(3) a. Ich **muss** die Bücher **abholen**, die ich bestellt habe.

b. Ich **muss** die Bücher, die ich bestellt habe, **abholen**.
'I have to pick up the books that I ordered.'

Linguistic studies have found that the frequency of extraposition has changed over time (e.g., Schildt 1976). Even though certain constituents can still be extraposed in modern German, as demonstrated by the examples above, there has been an increasing tendency to realize more and more information in the middle field. In the literature, a range of factors is discussed, which may influence extraposition, including the length and complexity of the extraposed elements, discourse mode, or informational aspects.

Our project aimed to investigate whether information density in the sense of Shannon (1948) plays a relevant role in the (diachronic) development of extraposition. We hypothesized that moving complex, highly informative constituents to the post-field could be beneficial for sentence processing by preventing excess memory strain on the middle field. To test this hypothesis, we manually annotated Early New High German texts with syntactic information to compare properties of extraposed vs. non-extraposed constituents.[1] Studies on this data set suggest that information density indeed affects extraposition, albeit to varying degrees for different constituents and time periods (Voigtmann and Speyer 2021a; Voigtmann and Speyer 2021b; Voigtmann and Speyer forthcoming). One problem, however, is the limited generalizability of the results. Although our data set is significantly larger than those of previous studies with a total amount of ~1.8 million annotated tokens, conclusions can be drawn only for specific genres (medicine, theology) and time periods (17^{th} to 20^{th} century), for which annotated data exists. A more comprehensive investigation is not possible unless further data is enriched with the necessary information, which would require costly and time-consuming manual annotation.

In order to address this issue and resolve the ubiquitous lack of data, the goal of my dissertation project was to explore the automatic identification of extraposition in historical and modern German. When I first started this venture, it was still uncharted territory. Automatic annotations of historical German, especially beyond the morpho-syntactic level, are scarce, and the peculiarities of the data pose several challenges to the application of computational linguistic methods, such as data sparsity, high variability, and a lack of trained tools and models. With this project, I set out to address and overcome the inherent challenges of working with historical language with the ultimate goal of automatically detecting extraposition in texts of arbitrary length from various genres and time periods.

Along the way, I explored several syntactic annotations and created gold standard data and models that did not previously exist for historical German. Among other things, I worked on automatic topological field analysis to detect the post-field in modern and historical German (Ortmann 2020), and trained freely available models for constituency parsing of Early New High German that can be used for detecting (potentially) extraposed elements (Ortmann 2021b). Although these were, in a way, byproducts of my pioneering work, they can serve as a starting point for future projects of their own. In this thesis, I incorporate them in the order of my exploration and put the puzzle pieces together to arrive at the final goal of automatically detecting extraposition. In an example application, I demonstrate the benefits of the developed methods for linguistic studies by automatically analyzing different factors for the extraposition of relative clauses.

[1] The data set is available at `https://github.com/rubcompling/C6Samples`. All links in this document were last checked on October 19, 2022.

Nevertheless, this thesis is only a starting point for the investigation of extraposition and other diachronic syntactic analyses. Instead of a deep dive into the causes of extraposition, I focused on laying the foundation for such studies by figuring out what is or is not (yet) possible with the resources available. In doing so, I adopted a rather pragmatic approach, creating methods and models of practical value for linguistic research. I shall point out that this exploratory nature of my project entails that I did not focus on fine-tuning the latest NLP models to improve the results down to the last detail. The field of computational linguistics is rapidly evolving, and future studies can experiment with replacing the individual components from this thesis with more accurate tools or training better models with additional data as it becomes available. I see the main contribution of this thesis in providing the groundwork for exploring these topics further, raising awareness of the problems, and suggesting ways for future improvement.

Structure of this Thesis

The remainder of this thesis is structured as follows: Chapter 2 describes the background information necessary to follow the contents of this thesis. That includes a more detailed look at the phenomenon of extraposition and the factors that are hypothesized to influence it, as well as a brief overview of previous work on the automatic analysis of historical (German) language.

Chapter 3 explains difficulties of working with historical data and introduces the modern and historical data sets that are used as gold standard and training data for the different annotation studies.

In Chapter 4, a new evaluation method called *FairEval* for the evaluation of labeled spans is presented, which I developed during my dissertation project and which is used to evaluate the annotations in this thesis.

Chapters 5–7 include the actual annotation studies. Chapters 5 and 6 describe the different annotations that serve as prerequisites for the recognition of extraposition. Chapter 5 focuses on topological field analysis with the intention of finding the post-field as the location of extraposed elements. Chapter 6, aims to identify candidates for extraposition, starting with simple chunks before shifting to the recognition of more complex phrases and clauses based on a constituency analysis. Chapter 7 is the heart of this thesis, where all threads come together to identify extraposed elements for the automatic analysis of extraposition.

To demonstrate the benefits of the developed methods, Chapter 8 provides an example application, which takes a quantitative look at different factors that presumably influence the extraposition of relative clauses, namely length, orality, and information density. The thesis concludes with a summary and an outlook in Chapter 9.

Previous Publications

Several of the enlisted chapters are based on previous publications, in particular:

- Chapter 4 corresponds to **Ortmann (2022)**. Fine-Grained Error Analysis and Fair Evaluation of Labeled Spans. In *Proceedings of the 13th Language Resources and Evaluation Conference (LREC)*. Marseille, France, pp. 1400–1407.
- Chapter 5 is based on the study from **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 (LaTeCH-CLfL)*. Barcelona, Spain (online), pp. 10–18.
- Chapter 6.1 corresponds to **Ortmann (2021a)**. Chunking Historical German. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*. Reykjavik, Iceland (online), pp. 190–199.
- Chapter 6.2 corresponds to **Ortmann (2021b)**. Automatic Phrase Recognition in Historical German. In *Proceedings of the 17th Conference on Natural Language Processing (KONVENS 2021)*. Düsseldorf, Germany, pp. 127–136.

I will also refer to several joint publications with Stefanie Dipper on the automatic identification of orality:

- **Ortmann and Dipper (2019)**. Variation between Different Discourse Types: Literate vs. Oral. In *Proceedings of the NAACL-Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*. Minneapolis, MN, pp. 64–79.
- **Ortmann and Dipper (2020)**. Automatic Orality Identification in Historical Texts. In *Proceedings of the 12th Language Resources and Evaluation Conference (LREC)*. Marseille, France, pp. 1293–1302.
- **Ortmann and Dipper (forthcoming)**. Nähetexte automatisch erkennen: Entwicklung eines linguistischen Scores für konzeptionelle Mündlichkeit in historischen Texten.

Wherever I incorporate text or material from previous publications, this is indicated with a footnote at the beginning of the chapter.

Additional Resources

To support the practical usefulness of this thesis, it is accompanied by a range of additional material, including data, models, and code:

Data The gold data sets and automatically created annotations from Chapters 5–7 of this thesis are provided with the evaluation results and documentation in a Git repository at `https://github.com/rubcompling/classig-data`. The repository includes all gold data sets that can be legally redistributed. For data sets with restrictive licenses, information about the selected train/dev/test splits is given. For access to the original data, contact information is provided.

Due to space constraints, the large compilation of annotated data sets from the example application in Chapter 8 cannot be stored in the same repository. Instead, all data sets and language models can be downloaded from the Zenodo repository at `https://doi.org/10.5281/zenodo.7180973`.

Models & Code The Python code to run the experiments in this thesis and annotate new data sets is released as the CLASSIG pipeline (**C**omputational **L**inguistic **A**nalysis of **S**yntactic **S**tructures **I**n **G**erman). It is made available in a Git repository at `https://github.com/rubcompling/classig-pipeline` under a free, permissive license so that results can be reproduced and built upon in future work. The repository also contains the trained models and the Java and Python libraries required to use them. In addition, I provide the R code used to create the plots and statistics for this thesis.

Due to space constraints, the chunker models (Chapter 6.1) cannot be stored in the same repository. Instead, all chunker and parser models are also included in the data package and can be downloaded from Zenodo (`https://doi.org/10.5281/zenodo.7180973`).

I hope that these resources prove fruitful for future (computational) linguistic studies on syntactic structures in (historical) German.

CHAPTER 2

Background

This chapter describes the background information that is necessary to follow the contents of this thesis. I will start with a more detailed look at the phenomenon of extraposition in Section 2.1 and describe different factors that are assumed to influence it in Section 2.2. I will pay special attention to the factors that were at the core of our project's research agenda. Specifically, these are length (2.2.1), orality (2.2.2), and information density (2.2.3). The three factors will play an essential role in Chapter 8, where their influence on the extraposition of relative clauses is examined in a large collection of modern and historical data. The chapter closes with a general overview of related work on the automatic analysis of historical (German) language in Section 2.3. The discussion will be complemented with details on individual annotations in the respective chapters.

2.1. The Phenomenon of Extraposition

As already mentioned in the introduction, this thesis emerged from a project on the diachronic development of extraposition. In German, different types of 'extraposition' can be distinguished, such as right dislocation, apposition, repetition, etc. (Altmann 1981). However, these distinctions are based on criteria like prosody that cannot (always) be inferred from written text, especially for historical data. In our project, we subsumed all of these types under the term 'extraposition', which we define as the movement of constituents from the middle field to the post-field of the sentence.[2] Some constituents can also be extraposed from the pre-field, e.g., attributive clauses, but this is the exception rather than the rule (Zifonun et al. 1997).

Example (4a) shows a relative clause located in the middle field, directly adjacent to its nominal antecedent. I will refer to such constituents in their base position as being placed *in situ*. In (4b), the relative clause has been *extraposed*, i.e., it has been moved away from its antecedent and behind the right sentence bracket.

[2] Even though I use the word 'movement' here, this thesis is not intended to make any claims about underlying grammatical processes that lead to a specific surface structure. Instead, I focus on a theory-neutral description and computational analysis of the syntactic variation that can be observed between the unmarked, default position of constituents in the middle field and an alternative, 'exceptional' positioning in the post-field. The annotations that are created with the methods from this thesis can, of course, be used to find data-based evidence from the perspective of specific grammatical theories, cf., e.g., Voigtmann and Speyer (forthcoming).

(4) a. Er **will** das Buch, das ich empfohlen habe, **lesen**.

b. Er **will** das Buch **lesen**, das ich empfohlen habe.
'He wants to read the book that I recommended.'

The definition of extraposition as moving elements from one specific field to another is based on the topological field model (Höhle 2019; Wöllstein 2018). According to this framework, German sentences are structured into a linear sequence of fields organized around the verbal material. A detailed explanation of the topological field model will be presented in Chapter 5, which dives into the automatic recognition of fields in modern and historical German. Here, I will only briefly illustrate the idea with a simplified analysis of example (4a):

Pre-field	**Left bracket**	Middle field	**Right bracket**	Post-field
Er	**will**	das Buch, das ich empfohlen habe,	**lesen**.	
He	*wants*	*the book that I recommend*	*to read.*	

In the example, there are five different fields: pre-field, left sentence bracket, middle field, right sentence bracket, and post-field. The left and right brackets contain the verbal elements (here: modal verb and infinitive). Together they form the basic skeleton of the sentence, also called the sentence frame. The remaining fields are named according to their position before (pre-field), in between (middle field), or behind (post-field) the sentence brackets. In the example, the pre-field contains the subject, while the remaining information is put in the middle field, and the (optional) post-field remains empty.

For some constituents, the post-field is considered the default position, e.g., certain types of subordinate clauses (Zifonun et al. 1997, p. 1651ff.). If other elements are moved to the post-field, this is called extraposition. The simplified analysis of example (4b) with the extraposed relative clause in the post-field would look like this:

Pre-field	**Left bracket**	Middle field	**Right bracket**	Post-field
Er	**will**	das Buch	**lesen**,	das ich empfohlen habe.
He	*wants*	*the book*	*to read*	*that I recommend.*

When working with historical data, it is important to note that the topological field model is based on modern German sentence structure, which only evolved over time and can differ substantially from historical language. Even though the sentence frame is already attested for Old and Middle High German, it was finally established only in the Early New High German period (Schildt 1976; Sahel 2015, among others). Especially the position of the right sentence bracket was still subject to change in older data, and the model may not be able to capture the full complexity of historical syntactic structures. As a consequence, several scholars argue that post-field placement in modern and historical German is not directly comparable (see, e.g., the discussion in Sahel 2015).

Nevertheless, it seems reasonable in practice to apply the framework to historical German, as this allows to investigate the emergence of modern sentence structure and the diachronic development of extraposition.

Diachronic Development

As linguistic studies have shown, the relative frequency and the types of elements that can be extraposed have changed over time (e.g., Schildt 1976; Coniglio and Schlachter 2015; Sapp 2014; Sahel 2015; Speyer 2016). When the sentence frame was still developing, various constituents used to be placed at the borders of the clause for different reasons (Paul 2007). Over time, especially the post-field placement of phrases has been significantly reduced, leading to less extraposition overall (Coniglio and Schlachter 2015; Sapp 2014, among others). In modern (standard) German, extraposition is restricted mainly to complex constituents like relative and complement clauses, comparative elements, and some prepositional phrases (Zifonun et al. 1997), cf. examples (5)–(7).

(5) **Hör** endlich **auf** mit dem Quatsch!
'Stop the nonsense!'

(6) Das Wetter **ist** noch schöner **gewesen** als gestern.
'The weather has been even better than yesterday.'

(7) Ich **muss** die Bücher **abholen**, die ich bestellt habe.
'I have to pick up the books that I ordered.'

Other constituents in modern German are extraposed almost only in spoken or oral-like language (Zifonun et al. 1997), e.g., noun, adjective, or adverb phrases as in examples (8)–(10) from the TüBa-D/S treebank of spoken German (Hinrichs et al. 2000).

(8) **Da** ich immer in die Stadt **muß** jeden Morgen, wäre Innenstadtnähe ziemlich günstig.
'Since I always have to go to the city every morning, downtown would be very convenient.'

(9) **Wenn** Sie dann Lust **hätten** spontan, könnten wir auch noch in das Theater gehen.
'If you would like spontaneously, we could also go to the theater.'

(10) **Sollen** wir dann abends noch was **machen** zusammen?
'Should we do something in the evening together?'

In earlier stages of German, these constituents were more regularly extraposed in written language, too (cf., e.g., Coniglio and Schlachter 2015; Schildt 1976; Speyer 2016), albeit to varying degrees depending on several influencing factors (cf. Section 2.2). Since extraposition is an optional process, the next section will address the question of what causes extraposition and discuss several factors that have been identified as possible triggers.

2.2. Influencing Factors

In the literature, various explanations for the phenomenon of extraposition exist. What many of them have in common – even if not stated explicitly – is a processing perspective: The movement of constituents to the post-field is assumed to somehow facilitate communication (e.g., Hawkins 1992; Gibson 1998). However, different hypotheses exist regarding the influencing factors that underlie this process. For example, processing the sentence might be facilitated by specifically extraposing longer or more complex constituents (e.g., Uszkoreit et al. 1998; Wasow 1997) or by extraposing constituents that convey high amounts of information (e.g., Vinckel 2006; Voigtmann and Speyer 2021a). Reducing dependency lengths between the sentence brackets through extraposition could also help sentence processing in accordance with syntactic locality principles (Gibson 1998; Futrell et al. 2015). It seems likely that there is an interplay of these different factors (e.g., long constituents may be more complex, contain more information, and cause longer dependencies), which especially comes into play in oral language where processing constraints are more relevant than in written language (Weiß 2005).

In this section, I will focus on three factors that played a central role in our research project: length, orality, and information density. Sections 2.2.1–2.2.3 describe each of the factors in detail, including how they are calculated and what has been observed about their effects in previous studies. In Chapter 8, their influence on the extraposition of relative clauses will be explored with the methods that are developed in this thesis. Other possible triggers of extraposition are briefly discussed in Section 2.2.4.

2.2.1. Length

The observation that the order of constituents is influenced by their length is commonly attributed to Behaghel (1932). His 'law of increasing constituents' (*Gesetz der wachsenden Glieder*) states that shorter elements precede longer elements. In the literature, it is generally assumed that this principle also applies to extraposition. Studies agree that longer constituents are extraposed more often than shorter ones, and this observation holds for phrasal and clausal extraposition as well as across different genres and time periods (Uszkoreit et al. 1998; Wasow 1997; Speyer 2016; Sapp 2014; Voigtmann and Speyer forthcoming, among others).

For example, Sapp (2014) finds that only 10% to 33% of phrases with 1–3 words in his data set of Middle High German and Early New High German are extraposed vs. 50% to 86% of phrases with 5–10 words. Similarly, Uszkoreit et al. (1998) report that extraposed relative clauses (RelCs) in modern German are about 1.3 words longer than *in situ* RelCs, and extraposition is rated as more acceptable for longer RelCs.

Often the terms length, heaviness, and complexity are used synonymously. While Zifonun et al. (1997) claim that structural complexity is more relevant than sheer length, Wasow (1997) finds that length and complexity are correlated: Longer constituents also tend to have more complex structures, and counting words is equally useful as counting nodes or phrasal nodes. The experiments by Weber (2019) suggest that the number of words may even be more relevant than the number

of phrasal nodes for the acceptability of PP extraposition. I will follow these considerations in understanding length as the number of words in a constituent.

From a processing perspective, extraposing longer elements makes sense not only for the listener but also for the speaker because it allows to post-pone the planning of the constituent (Wasow 1997). Concerning the observed differences between constituent types (Section 2.1), it also seems plausible that the importance of length for extraposition explains why long, complex constituents like relative clauses are often found in the post-field, also in standard written German. In modern newspaper text, 25% of the relative clauses are (unambiguously) extraposed (Chapter 7, cf. also Uszkoreit et al. 1998). In contrast, phrases, which are usually shorter and less complex, are mainly placed *in situ*. Overall, less than 2% of all phrases are extraposed. Even for prepositional phrases, which are the most frequently extraposed phrase type, less than 5% of the instances in modern newspaper text are found in the post-field (cf. Chapter 7).

2.2.2. Orality

The second influencing factor I want to consider in this thesis is referred to by various names like genre, register, discourse mode, or formality. Essentially, these terms point to the underlying notion of *conceptual orality* as a trigger of extraposition. The distinction between conceptually oral and literate language was established by Koch and Oesterreicher (1985), who observed that linguistic utterances, independently of the medium, can show characteristics that are typically attributed to written or spoken language (cf. also Halliday 1989). For example, despite its spoken realization, a scientific talk may resemble prototypical written language, while chat communication, although realized in the written medium, exhibits characteristics of spoken language.

A register analysis by Biber (1995) suggests that this distinction is a universal linguistic phenomenon, even though it may be realized differently in different languages. In German, extraposition is considered a typical feature of orality (Müller 1990; Richter 1985; Weiß 2005; Vinckel-Roisin 2015), i.e., spoken and spoken-like language shows more extraposition than written and literate-style language. For example, modern spoken German allows to extrapose more diverse constituent types, including several types of (typically short) phrases (Zifonun et al. 1997; Richter 1985; Tomczyk-Popińska 1987; Weiß 2005, among others; cf. Section 2.1). And it generally shows higher rates of extraposition than modern written German. In the TüBa-D/S treebank of spoken German (Hinrichs et al. 2000), about 10% of the prepositional phrases are extraposed, compared to only 3% in newspaper text from the TüBa-D/Z corpus (Telljohann et al. 2017; see Chapter 7) – probably because processing constraints are more relevant in spoken than written language.

For historical German, obviously, no spoken data exists, but the same difference can be observed within the written medium when comparing conceptually oral and literate data. For example, orally-oriented texts like women's letters or fictional dialogues show more phrasal extraposition than administrative texts (Ebert 1980; Dipper and Schultz-Balluff 2013).

One hypothesis is that the diachronic decrease of (phrasal) extraposition may, at least in part, be attributed to a stylistic change of written language and, hence, to orality. Older writings are said to be closer to spoken language (Betten 1989), i.e., more orally-oriented, and therefore may contain higher rates of extraposition – just like modern spoken language contains more extraposition than modern written language. With the development of a literate style, middle fields may have become denser with fewer constituents being extraposed because the written medium does not depend as much on working memory capacity. Over time, due to the 'prestige' of written language, this development may have reflected back into spoken language (Halliday 1989), leading to less extraposition overall.

To investigate such hypotheses and the general interplay of orality and extraposition, the degree of orality must be quantified. Koch and Oesterreicher (2007) propose mainly extra-linguistic criteria to situate texts on the literate-to-oral continuum, such as publicity vs. privacy, weak vs. strong emotional involvement, distance vs. proximity, or monologicity vs. dialogicity. The problem is that these criteria are rather vague and difficult to operationalize. One possible approach is to group texts into registers with prototypical extra-linguistic characteristics, such as scientific publications or spontaneous spoken communication. That is also what studies on extraposition do when they compare, e.g., the proportion of extraposed phrases in letters and administrative texts. As we have shown in previous studies (Ortmann and Dipper 2019; Ortmann and Dipper 2020), this approach can be a good approximation of orality.

However, there is still a significant amount of variation within registers. In response to this, we developed a method to objectively measure the degree of orality of individual texts (Ortmann and Dipper forthcoming). Our approach is based on the model by Ágel and Hennig (2006) who proposed to determine linguistic features like deixis, ellipsis, interjections, or the proportion of complete sentences in a text and compare them with the distribution in a prototypical text to calculate the relative degree of orality. Since recognizing their features requires a careful manual analysis of every single sentence, we came up with a new set of features that can be determined fully automatically based on standard annotations. We include features from the areas of complexity, variance, reference/deixis, syntax, and lexis. For an overview of the original feature set, see Ortmann and Dipper (2019). When we tested the features on modern German (Ortmann and Dipper 2019) and historical German (Ortmann and Dipper 2020; Ortmann and Dipper forthcoming), we found that especially simple features like word length or the frequency of certain pronouns are good predictors of orality.

Based on these findings, we developed a statistical model that takes the most informative features to calculate an objective orality score for individual texts.[3] Table 2.1 lists the features that are included in the score with their respective weights and definitions. In order to determine the degree of orality, the features are automatically identified in a given text. Since individual features can take on very different values (e.g., an average word length of 5 letters vs. a proportion of interjections

[3] A Python implementation to calculate our orality score is provided at `https://github.com/rubcompling/COAST`. The functionality is also integrated in the code that comes with this thesis and is used for the example analysis in Chapter 8.

Feature	Weight	Definition
mean_word	-0.819	Mean word length.
subord	-0.314	Ratio of subordinating conjunctions (tagged as KOUS or KOUI) to full verbs.
V:N	0.528	Ratio of full verbs to nouns.
PRON1st	0.717	Ratio of 1st person pronouns with lemmas *ich* 'I' and *wir* 'we' to all words.
DEM	0.060	Ratio of demonstrative pronouns (tagged as PDS) to all words.
DEMshort	0.365	Proportion of demonstrative pronouns (tagged as PDS) with lemmas *diese* or *die* 'this/these', which are realized as the short form (lemma *die*).
PTC	0.104	Proportion of answer particles (*ja* 'yes', *nein* 'no', *bitte* 'please', *danke* 'thanks') to all words.
INTERJ	0.276	Proportion of primary, i.e., one-word interjections (e.g., *ach*, *oh, o*, *bravo*, *halleluja*, *hmm*) to all words.

Table 2.1.: List of features included in our orality score with their respective weights and definitions (Ortmann and Dipper forthcoming). A positive weight means that the feature indicates conceptual orality, a negative weight indicates conceptual literacy.

of 0.1%), values are scaled to the standardized area between 0 and 1 with a linear transformation (Eq. 2.1). Because sensible minimum and maximum values are difficult to determine for most features, the lowest value for a given feature is mapped to 0 and the highest value to 1.

$$x'_0 = \frac{x_0 - min}{max - min} \tag{2.1}$$

To calculate the orality score, the scaled features are then factored in with their respective weights, as shown in Eq. 2.2. In our experiments, we found a high correlation ($r = 0.92$) of the orality score with expert judgments, making it a promising method to explore the effects of orality on the phenomenon of extraposition (Chapter 8).

$$\textit{Orality Score} = \sum_{i=1}^{N} w_i * x_i \tag{2.2}$$

2.2.3. Information Density

The third factor I want to discuss in more detail was the main focus of our research project. We hypothesized that extraposition is a form of information management, understanding information in the sense of Shannon (1948) as predictability in context. If a word is highly predictable from the context, it conveys less information than less predictable, *surprising* words. The surprisal of a word is calculated as the negative log of the word's probability in a given context (Eq. 2.3).

$$surprisal(word) = -log_2(p(word|context)) \tag{2.3}$$

Word probability can be estimated with language models (Jurafsky and Martin 2021). In our project, we used n-gram and skip-gram models, which condition the probability of a word on n preceding words. Words that are infrequent in the given context have a low probability and, therefore, a high surprisal, which has been linked to perceiving and production difficulty (Hale 2001; Jaeger 2010).

According to information theory, there is an upper limit to how much information can be reliably communicated through a specific channel at any given time (Shannon 1948). If the channel capacity is exceeded, this can result in loss of information and failed communication. Language users are assumed to structure their messages in a way that ensures successful communication. They do so by distributing information as evenly as possible across their utterances to prevent peaks and troughs in the information profile, which has become known as the Uniform Information Density (UID) hypothesis (Levy and Jaeger 2007). The most prominent syntactic example is the phenomenon of *that*-omission in English relative clauses. Levy and Jaeger (2007) show that language users are more likely to insert the optional relativizer *that* to smooth the information signal if the beginning of the relative clause is highly surprising. Otherwise, the relativizer is more likely to be omitted.

For extraposition in German, we assume a similar effect: Highly informative, surprising constituents tend to be extraposed more often to prevent overloading the middle field with peaks of information. Postponing the constituent to the post-field could reduce memory strain on the middle field and distribute the information more evenly across the sentence. In German literature, this is also referred to as *Informationsentflechtung* ('information disentanglement', Zifonun et al. 1997).

There are (at least) two different ways to measure this effect. Looking at the mean surprisal of a constituent (i.e., the average surprisal of all words comprising the constituent) corresponds to the idea that too informative units may exceed channel capacity and should be moved behind the sentence frame where more cognitive resources are available again. Voigtmann and Speyer (2021a), Voigtmann and Speyer (2021b), and Voigtmann and Speyer (forthcoming) inspect cumulative and mean surprisal on our manually annotated data set and find, when controlling for length, that higher surprisal can favor extraposition of relative clauses and (attributive) PPs in Early New High German.

The second option is to look at the information profile of the entire sentence. If we assume that extraposing surprising constituents facilitates communication, this may be explained by a smoothing effect on the overall information profile. Compared to studies like Levy and Jaeger (2007), this is more difficult to quantify, though, because the movement of a constituent simultaneously creates changes in several places. Therefore, we started to experiment with the DORM measure (**D**eviation

Of the **R**olling **M**ean), introduced by Cuskley et al. (2021) to objectively quantify the uniformity of information profiles and compare the differences between different orders of elements like words or constituents.

Given a list of surprisal values (e.g., n-gram word surprisal), DORM takes the arithmetic means of every two adjacent surprisal scores (Eq. 2.4) and calculates the sample variance of these rolling means, as shown in Equation (2.5). Lower DORM values correspond to a more uniform information profile.

$$\text{for } i \text{ in} (1..n-1) : \textit{rolling mean}_i = \frac{surprisal_i + surprisal_{i+1}}{2} \tag{2.4}$$

$$DORM = s^2 = \frac{\sum_{i=1}^{n} (\textit{rolling mean}_i - \bar{x})^2}{n-1} \tag{2.5}$$

Since DORM values depend on the surprisal scores of individual items, they are not directly comparable between sentences. Instead, Cuskley et al. (2021) introduce UIDO, which is the most uniform information distribution that can be achieved by scrambling the elements in a sentence (e.g., words or constituents). When comparing sentences with their respective UIDO variants, they find that information profiles of human language are closer to the optimal distribution than would be expected by chance, speaking in favor of the UID hypothesis.

However, arbitrary permutations of words or constituents to determine the optimal information profile of a sentence are not linguistically plausible because the possible orders that language users can (realistically) choose from are restricted by the language's grammar. In our project, we introduced the concept of a variant corpus, which allows to explore the effects of plausible word order variation. In a corpus of variants, only specific aspects of a sentence are manipulated, e.g., by 'un-doing' the extraposition, while other factors are kept constant. Chapter 7.3 explains how to create a variant corpus for extraposition. In contrast to UIDO, these variant sentences are still valid grammatical expressions in the given language and provide a basis for comparing the information profiles of original and variant sentences to study the effects of extraposition (or any other manipulated phenomenon) on the uniformity of the information distribution.

We propose the $\text{DORM}_{\text{diff}}$ value (Ortmann et al. 2022) to be the difference between the $\text{DORM}_{\text{orig}}$ value of the original sentence and $\text{DORM}_{\text{variant}}$ of the variant sentence (cf. Eq. 2.6). An example calculation of $\text{DORM}_{\text{diff}}$ values can be found in Chapter 8.

$$DORM_{\text{diff}} = DORM_{\text{orig}} - DORM_{\text{variant}} \tag{2.6}$$

If the $\text{DORM}_{\text{diff}}$ value is negative, the original sentence has a smoother information profile than the variant, suggesting that the phenomenon under investigation positively influences sentence processing. In Chapter 8, I will exemplarily test whether the extraposition of RelCs may be associated with high mean surprisal values or overall improvements of the information profile.

2.2.4. Other Factors

Besides the already mentioned effects of constituent type, time period, length/complexity, orality, and information density, there is a variety of other factors, which are said to influence the occurrence of extraposition. In modern German, the most important aspects seem to be distance and pragmatic, discourse-related factors. Several studies report that constituent length competes with the distance between base position and post-field (Uszkoreit et al. 1998; Weber 2019, among others). Extraposition occurs primarily over short distances or for very long material. In a corpus study on modern German, Uszkoreit et al. (1998) find that, on average, relative clauses are extraposed over only 1.6 words. They also note that extraposition is much more likely over verbal material than over other intervening elements.

Regarding the function of constituents, adjuncts are more readily extraposed than arguments (Zifonun et al. 1997). Other influencing factors include the depth of embedding, definiteness, citations, or the presence of demonstratives (Strunk 2014; Weber 2019; Coniglio and Schlachter 2015). For relative clauses, it is also assumed that restrictiveness facilitates extraposition (Zifonun et al. 1997, but see Poschmann and Wagner (2016) and Voigtmann and Speyer (2021a) for opposing evidence).

In older stages of German, information structural aspects played a major role for extraposition, including focus (Sapp 2014; Poschmann and Wagner 2016) and givenness/newness (Speyer 2016). In modern German, the post-field can still be used to introduce new topics and serve as an area of emphasis (Vinckel-Roisin 2015). However, while many properties of extraposition are comparable between modern and historical German (Sapp 2014), the importance of information structural factors like givenness decreased over time (Coniglio and Schlachter 2015; Speyer 2016).

This list of additional factors that go beyond the focus of this thesis is certainly condensed, but it gives a general impression of the various influences that interact when it comes to extraposition.

2.3. Related Work

The evidence on influencing factors of extraposition, especially for historical German, stems mainly from qualitative studies on small, manually annotated data sets. For example, Sapp (2014) analyzes 683 extraposed phrases from five centuries, which corresponds to about 1.4 cases of extraposition per year. Similarly, Sahel (2015) bases his study on 1.108 relative clauses from three registers and 150 years, i.e., about 2.5 RelCs per register and year. Although such detailed analyses are very precise and provide valuable insights, they cannot achieve the same statistical significance and generalizability as studies on large (modern) data sets. In addition, the strong expert involvement during data selection and/or qualitative investigation always comes with the risk of introducing biases into the analysis. Applying quantitative methods to complement these traditional approaches seems valuable to verify hypotheses and perhaps even discover previously overlooked patterns in the data.

However, the application of natural language processing (NLP) to historical language faces several obstacles. Historical data is characterized by a high degree of variation, which is certainly interesting in and of itself, consider, e.g., studies on diatopic and diachronic variation as reflected in the graphemic variation (Dipper and Waldenberger 2017; Waldenberger et al. 2021). However, this variation also hinders automatic analyses, aggravating the already existing problem of data sparsity. While modern NLP tools usually need large amounts of annotated data to train accurate models, such resources rarely exist for historical language. As a result, most previous work on the automatic analysis of historical language has focused on automatically retrieving texts from handwritten sources with OCR and creating low-level annotations like sentence and word tokenization, normalization, lemmatization, and morpho-syntactic analysis or Part-of-Speech (POS) tagging. To date, there are several automatically annotated corpora of historical German, including the German Text Archive (DTA, BBAW 2021), GerManC (Bennett et al. 2007), TüBa-D/DC (Hinrichs and Zastrow 2012), and parts of the Reference Corpus of Early New High German (ReF, Wegera et al. 2021).

Beyond the morpho-syntactic level, much fewer resources and previous studies exist. Data sets that include syntactic annotations are mostly small and/or very specific to a certain research project (e.g., HIPKON, Coniglio et al. 2014; Deutsche Diachrone Baumbank, Hirschmann and Linde 2010), or the accuracy of the annotations remains unclear (TüBa-D/DC, Hinrichs and Zastrow 2012). ReF.UP (Demske 2019), the syntactically annotated part of ReF (Wegera et al. 2021), is a notable exception, but was released only recently.

The small corpus basis of highly varied language data qualifies historical German as a less-resourced language (Cieri et al. 2016) and makes training and using statistical or neural NLP tools difficult or impossible. To overcome this challenge, previous studies have experimented with different solutions. One possible approach is to use the automatic tools only for pre-annotation, which can significantly speed-up the manual annotation process and help to create large data sets more quickly (cf. Eckhoff and Berdičevskis (2016) for dependency parsing of Old East Slavic). Another approach is to pre-process the historical data, e.g., by normalizing word forms and punctuation, to increase the accuracy of a modern parser. Hinrichs and Zastrow (2012) train the Berkeley parser (Petrov et al. 2006) on modern German and apply it to German texts from the 13th to 20th century but do not provide evaluation results. Krielke et al. (2022) annotate scientific texts from the DTA with dependencies. After filtering out problematic sentences, they observe labeled attachment scores of about 80% on the subset of 'good' sentences.

A possibility to entirely bypass the need for training data is to fall back on (more or less) complex rules (Chiarcos et al. 2018; cf. also my experiments on sentence bracket recognition in Chapter 5). Contrary to statistical models, this approach requires expert knowledge to create rules and may be less robust, especially in rare cases. The topological field parser for Middle High German by Chiarcos et al. (2018) also relies on rich pre-annotations, which do not exist for most data sets. Another strategy that does not require any historical data is manipulating modern data. Petran (2012) takes a modern German data set and approximates historical language by removing punctuation and capitalization. Since these orthographic changes are not the only difference between modern and historical language, it is unclear if this yields realistic results.

In my annotation studies, I use modern data (and historical data, if available) to train statistical models that can be transfered to non-standard registers and historical time periods by operating on the shared level of POS tags. In Chapters 5–6, I will describe this approach in more detail and complement it with a discussion of related work on the specific annotations. Contrary to studies like Chiarcos et al. (2018) and Hinrichs and Zastrow (2012), I also conduct detailed evaluations and error analyses for each annotation to provide a realistic impression of annotation accuracy.

Chapter 3

Corpus Data

This chapter introduces the data sets that serve as training and evaluation data throughout the experiments in Chapters 5–7. As already alluded to, there are several challenges when working with historical data:

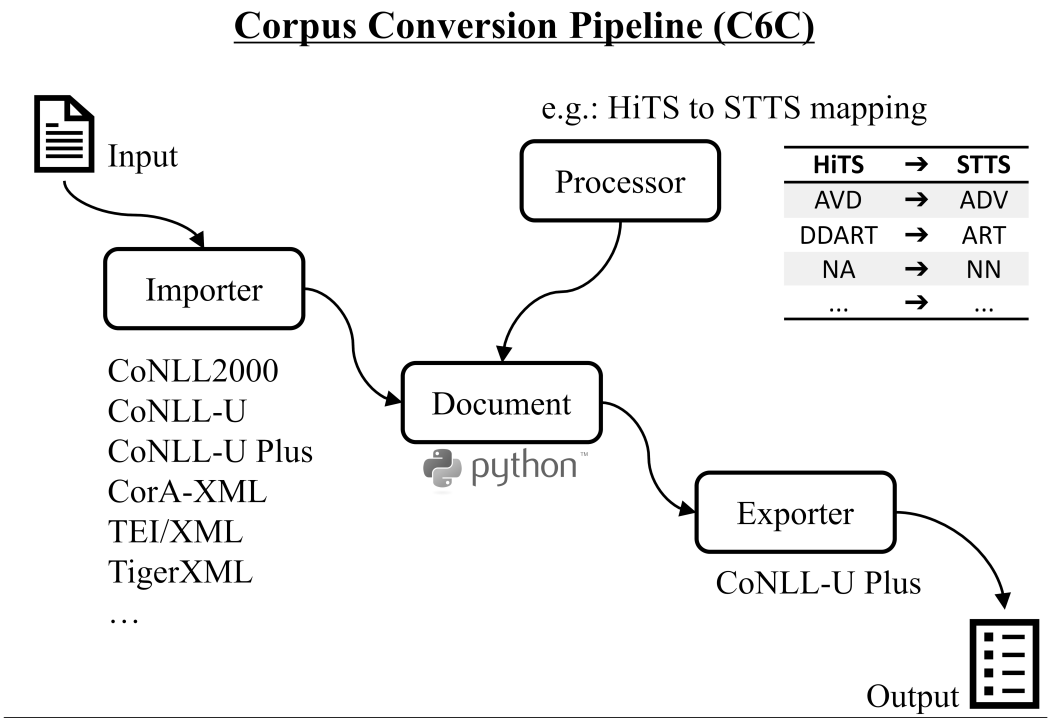
Data Formats Despite standardization attempts, e.g., by the Text Encoding Initiative (TEI),[4] there are no generally accepted standards regarding the annotation and storage of historical data that are used by all research projects (yet). As a consequence, corpora of historical German are supplied in a wide variety of data formats, including in particular:

- different versions of the column-based CoNLL formats
- various XML formats such as TIGER-XML, TEI/XML (e.g., DTA "Base Format", Geyken et al. 2012), CorA-XML (as produced by the CorA annotation tool, Bollmann et al. 2014),[5] EXMARaLDA's Basic Transcription (exb) format (Schmidt and Wörner 2014),[6] WebLicht TCF (Hinrichs et al. 2010), etc.

To deal with this multitude of formats, we implemented a Python pipeline that can be used to convert different historical corpora (and also modern data sets) to a uniform data format (see Figure 3.1). The converter is freely available under an MIT license.[7] An enhanced version is also integrated into the CLASSIG pipeline that accompanies this thesis, enabling the automatic syntactic analysis of different data sets with the developed methods. As shared data format, I have chosen the column-based CoNLL-U Plus format[8] because it can represent the necessary annotations for our project in a convenient and human-readable form. Besides default word-based annotations, we mainly want to store syntactic annotations, i.e., spans and trees.

Span annotations like chunks, phrases, or clauses can be represented with the simple BIO format that is commonly used for sequence labeling. Every token is annotated with a BIO tag, consisting of the token's position in the span (begin `B`, inside `I`, outside `O`), a dash (`-`), and the span label. So each token is tagged as either `B-S` (beginning of span), `I-S` (inside of span), or `O` (outside of span),

[4] https://tei-c.org/
[5] https://cora.readthedocs.io/en/latest/coraxml/
[6] https://exmaralda.org
[7] https://github.com/rubcompling/C6C
[8] https://universaldependencies.org/ext-format.html



1	Es	es	PPER	_	B-NC	B-NP-insitu	B-NP	B-VF
2	geben	geben	VVFIN	_	O	_	O	B-LK
3	ſich	sich	PRF	_	B-NC	B-NP-insitu	B-NP	B-MF
4	jezuweilen	jezuweilen	ADV	_	B-ADVC	B-ADVP-insitu	B-ADVP	I-MF
5	glůckliche	glücklich	ADJA	B-Antec-1	B-NC	B-NP-insitu	B-NP	I-MF
6	Umſtände	Umstand	NN	I-Antec-Head	I-NC	I-NP	I-NP	I-MF
7	in	in	APPR	_	B-PC	B-PP-insitu	B-PP	I-MF
8	unſerm	unsere	PPOSAT	_	I-PC	I-PP	I-PP	I-MF
9	Leben	Leben	NN	_	I-PC	I-PP	I-PP	I-MF
10	an	an	PTKVZ	_	O	_	O	B-RK
11	,	,	$,	_	O	_	O	O
12	welche	welche	PRELS	_	B-NC	B-RELC-extrap-1|B-NP-insitu	B-NP	B-NF|B-LK
13	…							

Achtzig erläuterte Grund-Fragen von Petrus Hanssen (1731; DTA)

Figure 3.1.: Illustration of the corpus conversion pipeline from our project. The converter can be used to transform corpora from different input formats to a standardized CoNLL-U Plus output format. We also provide processors, e.g., to map historical POS tagsets like HiTS (Dipper et al. 2013) to the German standard tagset STTS (Schiller et al. 1999). The internal document representation serves as a basis for the automatic analyses with the CLASSIG pipeline in this thesis. The image at the bottom shows an excerpt from the DTA gold data set in the column-based target format. The span annotations that are created in this thesis are encoded as BIO tags (see also Figure 3.2). The converter pipeline is available at `https://github.com/rubcompling/C6C`.

(S (NP Das) (VP ist (NP ein (AP einfacher) Satz)) .)

Das	*This*	`B-S|B-NP`
ist	*is*	`I-S|B-VP`
ein	*a*	`I-S|I-VP|B-NP`
einfacher	*simple*	`I-S|I-VP|I-NP|B-AP`
Satz	*sentence*	`I-S|I-VP|I-NP`
.	.	`I-S`

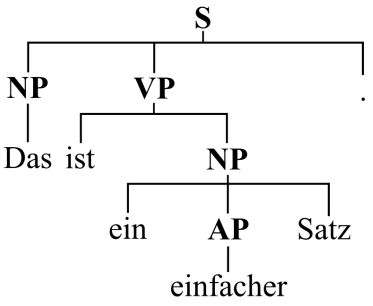


Figure 3.2.: Different options to represent tree structures in the CoNLL-U Plus format. The tree (right) can either be represented with a bracket structure (top) or with stacked BIO tags using vertical pipes (left). For example, the `S`-node dominates the complete sentence, i.e., it starts with the `B-S` tag for the first token and ends with the `I-S` tag for the last token. The first `NP` is dominated by the `S` node and dominates the word *Das*, so the `B-NP` tag is appended to the `B-S` tag of the first token. The `VP` is dominated by `S` and dominates *ist ein einfacher Satz*. Consequently, the `B-VP` and `I-VP` tags are appended to the `I-S` tags of the four tokens, etc.

with `S` being a place-holder for different labels. BIO tags can be stacked to represent multi-level annotations, e.g., several stacked entities. Also, the BIO format can be easily extended to store additional information like the position of a constituent (extraposed vs. *in situ*) by appending it to the BIO tag, e.g., `B-NP-extrap`. A detailed documentation is provided with the data sets and program code.

For the representation of tree structures, such as topological field annotations and constituency trees, the CoNLL-U Plus format offers (at least) two different possibilities, provided that there are no discontinuous nodes.[9] Trees can either be represented with a traditional bracket structure or be treated like span annotations, using several stacked BIO tags to represent the hierarchical tree structure (cf. the example illustration of the two options in Figure 3.2).

Tagsets Besides the multitude of data formats, there are no commonly agreed upon standard tagsets or guidelines for the syntactic annotation of historical German. Consequently, different corpora use different tagsets and annotation schemes, e.g., STTS (Schiller et al. 1999), HiTS (Dipper et al. 2013), and other custom tagsets for POS tagging. With our corpus conversion pipeline, we provide mappings for the different historical POS tagsets to STTS tags, the de-facto standard tagset for modern German (Schiller et al. 1999). In some cases, this mapping may lead to a loss of information (cf., e.g., Chapter 5 on relative adverbs and particles), but it allows to create a shared

[9] The ability to represent discontinuous structures is one of the main advantages of XML-based formats like TIGER-XML. However, standard NLP tools cannot handle such discontinuities, so I have removed them from the trees as described in Chapter 6. The provided data sets in CoNLL-U Plus format only contain the modified, continuous trees.

basis across data sets and time periods, which can serve as input for standard NLP tools (see the discussion on variation, below).

Corpus Size The third problem when working with historical language is a lack of (annotated) data, which is generally required for training and evaluating automatic annotation methods. Ideally, large amounts of annotated text from all time periods of German would be available that could be exploited for this purpose. However, the manual annotation of corpus data is effortful, time-consuming, and thus expensive. As a consequence, most historical data sets are provided only with basic or highly specific linguistic annotations that were compiled for a particular research project. Especially annotations beyond the morpho-syntactic level are rare and often exist only for small data sets.

As described in Chapter 2.3, there are different approaches to deal with this problem. In my dissertation project, I have decided to exploit large modern corpora (and historical data where available) to train statistical models and transfer the results to other registers and time periods. Smaller, manually annotated data sets are only used for evaluation. Contrary to previous studies (e.g., Chiarcos et al. 2018; Hinrichs and Zastrow 2012), this approach enables me to train powerful, flexible probabilistic models for modern and historical German and – maybe even more importantly – evaluate the results.

Variation Another difficulty for the automatic analysis of historical German is the inherently high variation. Especially the deviation from modern orthography makes the direct application of modern, usually word-based NLP tools problematic. Bollmann (2018) finds 80 different spelling variants for the word *Frau(en)* 'woman/women' in one Early New High German corpus. This degree of variation aggravates the already existing problem of data sparsity (cf. the discussion on corpus size above) and makes it difficult to train probabilistic models. One possible solution would be to normalize word forms, i.e., map them to a modern word form (cf. example (11) from the HIPKON data set). It has been shown that this approach can increase accuracy for dependency parsing of Middle English (Schneider et al. 2015) or tagging historical German and Dutch (Bollmann 2013; Tjong Kim Sang et al. 2017). In my studies, I choose an even higher level of abstraction and use primarily the POS tags, which are shared across data sets, as input for the models. This approach not only bypasses the problem of unstandardized word forms but may also mitigate the effects of lexical change and reduce the problem of data sparsity by shrinking the 'vocabulary' to a set of only 54 tags.

(11)

Orig:	vñ	wo̊lte	gan	zů	ſínem	vatt'	vnd	ſprechē
Norm:	und	wollte	gehen	zu	seinem	Vater	und	sprechen
POS:	KON	VMFIN	VVINF	APPR	PPOSAT	NN	KON	VVINF
	and	*wanted*	*to go*	*to*	*his*	*father*	*and*	*speak*

Finally, I expect relevant variation between different language registers, not only for historical but also for modern data. Most NLP tools are trained on newspaper or web data, for which abundant resources exist. But system performance can drop significantly when the models are applied to non-standard data (cf., e.g., Pinto et al. 2016). To get an impression of how well models can be transferred to out-of-domain data, I include data from different registers in the evaluation.

In total, I used four modern and four historical gold data sets for my experiments in Chapters 5–7. During the course of the project, these data sets have continuously evolved and were successively enriched with more layers of annotation – some derived automatically from existing annotations, others created manually by student annotators.[10] In this thesis, I use the final version of the data sets and reproduce the results from my previous studies with these data sets wherever possible. In combination with the new evaluation method, which I only developed after the last published study (see Chapter 4), this is the reason why numbers may differ from previously published results.

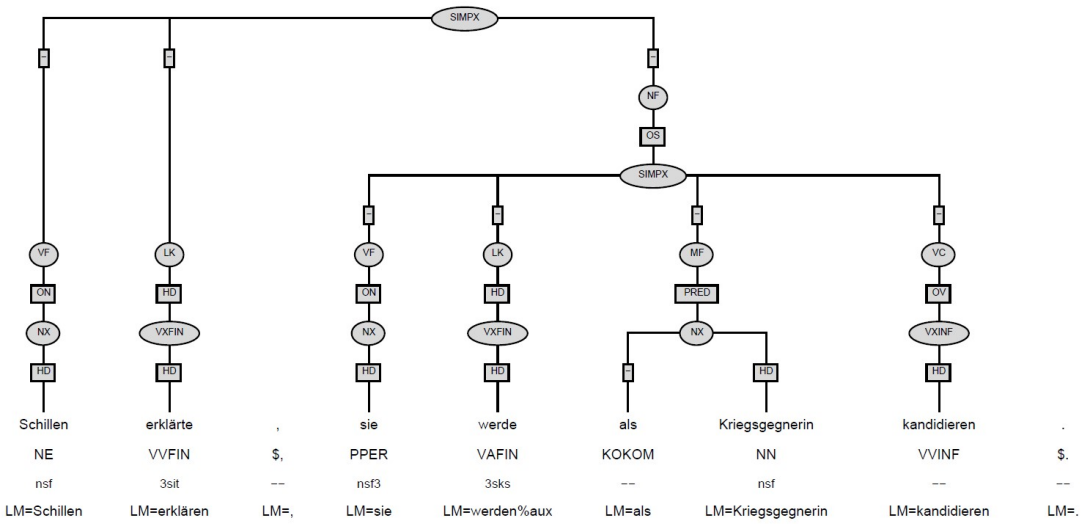
The following sections introduce the different modern (Section 3.1) and historical data sets (Section 3.2), followed by an overview of the corpora and available annotations in Section 3.3. As already mentioned, the data sets are provided for download at https://github.com/rubcompling/classig-data.

3.1. Modern Data Sets

For modern German as one of the twenty most spoken languages in the world,[11] ample language resources exist. For example, the German Wikipedia is the second largest active Wikipedia after English, with more than 2.7M articles and over 1.4B words.[12] However, most of the available data consists of raw text, for instance, Wikipedia articles or other web pages, and large corpora are often provided with automatically generated annotations, e.g., SdeWaC (Faaß and Eckart 2013). Also, syntactic analyses often focus on dependency annotation,[13] whereas only a few data sets include topological field annotations or constituency parses. To ensure a high quality of the training data and trained models, I selected two manually annotated German treebanks (TüBa-D/Z, Tiger) for training and evaluation in my studies. In addition, I include evaluation data from other registers (Spoken, Modern) to judge the transferability of models to out-of-domain data. In the following paragraphs, each of the modern data sets is briefly introduced.

[10] Manual annotations were performed with WebAnno, version 3.6.6 (Eckart de Castilho et al. 2016; https://webanno.github.io/webanno).
[11] Ethnologue, 25th edition, https://www.ethnologue.com/ (September 14, 2022)
[12] Considering official Wikipedias with at least 200 active users according to Wikimedia, https://meta.wikimedia.org/wiki/List_of_Wikipedias (September 14, 2022)
[13] https://universaldependencies.org/

TüBa-D/Z The TüBa-D/Z corpus (Telljohann et al. 2017)[14] is a collection of 3,816 German newspaper articles from 'die tageszeitung' (taz) with approx. 100k sentences and almost 2M tokens. The articles are semi-automatically annotated with morphology, POS tags, lemmas, and constituency trees that include topological field annotations. I will refer to these trees as TüBa-style trees, in distinction from the Tiger-style trees, which do not include topological fields (see below). The following image shows an example tree from the documentation (Telljohann et al. 2017, p. 155):

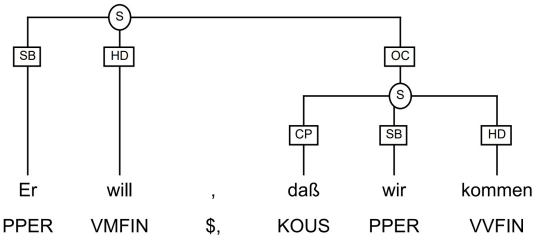


The corpus also contains automatically derived chunks (Kübler et al. 2010) and additional annotations like dependencies, named entities, and coreference resolution. For my studies, I added automatically derived phrases (cf. Chapter 6.2) and annotations of extraposition (cf. Chapter 7). The corpus is free to use for academic research but may not be redistributed. For my annotation experiments, it is split into a training (80%), development (10%), and test set (10%). The split is used consistently across all of my papers and the chapters of this thesis.

Tiger The Tiger treebank (Brants et al. 2004)[15] is the second modern newspaper corpus included in my thesis. It consists of 2,263 German news articles from the 'Frankfurter Rundschau' and contains about 50k sentences with 888k tokens. The articles are semi-automatically annotated with POS tags and constituency trees, henceforth referred to as Tiger-style trees. Contrary to TüBa-style trees, they do not contain information about topological fields. Discontinuous annotations were removed from the trees as described in Chapter 6. The following image shows an example tree from the annotation manual (TIGER Project 2003, p. 49):

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

[15] Version 2.2, https://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/tiger



The corpus also includes lemmas, morphology, and automatically derived dependency annotations. For my studies, I added automatically derived chunks (cf. Chapter 6.1), phrases (cf. Chapter 6.2), and the extraposition of relative clauses (cf. Chapter 7). The corpus is provided with a training, development, and test section. It is freely available for academic research but comes with a non-disclosure agreement and may not be publicly redistributed.

Spoken As discussed above, most NLP tools and models are trained and evaluated on modern standard language. To test the accuracy on other modern data, I include two non-standard data sets. The first one is the TüBa-D/S corpus (Hinrichs et al. 2000), a treebank of spoken German. It contains 14 transcribed dialogues from a business context with about 28k sentences and almost 300k tokens. The corpus is manually annotated with POS tags and TüBa-style constituency trees, which include topological fields. For this thesis, I added automatically derived phrases (cf. Chapter 6.2) and extrapositions (cf. Chapter 7). To prevent confusion with TüBa-D/Z, I will refer to the corpus as 'Spoken' in Chapters 5–7. The use is free for academic research but does not permit redistribution.

Modern The second non-newspaper corpus was compiled for our study on the evaluation of different NLP tools (Ortmann et al. 2019).[16] It contains about 500 sentences with 7.6k tokens from five different registers with varying degrees of formality: Wikipedia articles,[17] narrative text,[18] Christian sermons,[19] TED talk subtitles,[20] and movie subtitles.[21] The data was manually annotated with POS tags, lemmas, morphology, dependencies, topological fields, chunks, phrases, and extraposition. It is licensed under CC BY-SA 3.0, except for the source text of the TED talks, which are licensed under CC BY–NC–ND 4.0.

[16] https://github.com/rubcompling/konvens2019

[17] Sample from the Wikipedia subcorpus of DeReKo, http://corpora.ids-mannheim.de/pub/wikipedia-deutsch/2015/wpd15_sample.i5.xml.bz2

[18] Genre 'novelette' from GutenbergDE corpus, edition 14 (https://gutenberg.abc.de/), published after 1900.

[19] http://www.sermon-online.de

[20] German translations of English talks, https://www.ted.com/talks?language=de

[21] Genres 'Action, Adventure, Drama' and 'Comedy, Drama' from the OpenSubtitles database, http://www.opensubtitles.org/

3.2. Historical Data Sets

For historical German, the only large data set that includes TüBa-style trees and, hence, topological field annotations is the TüBa-D/DC corpus (Hinrichs and Zastrow 2012). However, the annotations were created fully automatically with a parser model trained on modern German (TüBa-D/Z corpus), and no evaluation results are reported, so the accuracy remains unclear. As a consequence, it does not seem reasonable to train or evaluate newly developed methods on this corpus. Instead, I selected four manually (or semi-automatically) annotated data sets for my studies, which are briefly described in the following paragraphs.

Mercurius The Mercurius treebank (Demske 2005)[22] contains approx. 8k sentences and 187k tokens of newspaper text from the 16th and 17th centuries. It is semi-automatically annotated with POS tags and Tiger-style constituency trees, from which I automatically derived chunks (cf. Chapter 6.1) and phrases (cf. Chapter 6.2). Discontinuous annotations were removed as described in Chapter 6. The POS tagset was mapped to the modern standard tagset STTS (Schiller et al. (1999); see Table A.1 in the appendix). The corpus is licensed under CC BY 3.0 and was split into a training (80%), development (10%), and test set (10%) for my studies. The split is used consistently across my papers and the chapters of this thesis.

ReF.UP The ReF.UP treebank (Demske 2019) is a subcorpus of the Reference Corpus of Early New High German (Wegera et al. 2021).[23] It includes 26 documents from different registers with 21k sentences and 600k tokens from several language areas from the 14th to 17th century. The corpus is semi-automatically annotated with POS tags and Tiger-style constituency trees, from which I automatically derived chunks (cf. Chapter 6.1) and phrases (cf. Chapter 6.2). Again, discontinuous annotations were removed, and the POS tagset was mapped to the modern standard tagset STTS (Schiller et al. (1999), see Table A.2 in the appendix). The corpus is licensed under CC BY-SA 4.0 and was split into a training (80%), development (10%), and test set (10%) for my studies. The split is used consistently across my papers and the chapters of this thesis.

HIPKON Besides the two treebanks, I include two smaller, manually annotated data sets for evaluation. The HIPKON corpus (Coniglio et al. 2014) contains sermons from the 12th to the 18th century (except 15th century). Only sentences with a post-field are annotated, yielding 342 annotated sentences with 4.2k tokens. Annotations include POS tags, basic syntactic structures, and a non-recursive topological field analysis. For my studies, recursive topological fields, chunks, phrases, and extrapositions were added manually. Also, the custom POS tagset was mapped to the German standard tagset STTS (Schiller et al. (1999); see. Table A.3 in the appendix). The corpus is licensed under CC BY 3.0.

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

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

DTA The German Text Archive (DTA, BBAW 2021) contains large amounts of German texts from a variety of registers from the 16th to the early 20th century. The data is provided with automatically annotated sentences, STTS POS tags, lemmas, and orthographic normalization. However, the annotation accuracy turned out to be too low for a meaningful evaluation.[24] Therefore, I decided to use a smaller, manually annotated subset of sentences for my experiments.[25] My data set consists of 600 sentences with 18.8k tokens from 29 texts. The texts were published from the 16th to 20th century in various genres. Included are five newspaper texts and three texts from each of the following genres: funeral sermon, language science, medicine, gardening, theology, chemistry, law, and prose. Sentence boundaries and POS tags were manually corrected for my studies. In addition, the data set was manually enriched with annotations of topological fields, chunks, phrases, and extrapositions. The DTA is licensed under CC BY-SA 4.0.

3.3. Overview of Corpora and Annotations

Table 3.1 gives an overview of the gold data sets that were presented in the previous sections, including sentence and token counts. As mentioned, the smaller gold data sets (Modern, HIPKON, and DTA) were annotated manually, whereas annotations for the other corpora were already provided with the corpus or automatically derived from existing annotations. More information on the automatically derived annotations can be found in the respective Chapters 5–7. Whether a data set is included in a particular annotation study depends on the availability of gold annotations for this data set. For example, the Mercurius and ReF.UP corpora can only be used to evaluate chunking and phrase recognition because no topological field annotations exist for these data sets. In the Tiger corpus, only the extraposition of relative clauses can be identified with the provided annotations. The spoken data set includes a constituency analysis with topological fields, but it is not possible to derive consistent chunks from it because relevant annotations of coordination are missing. So, this data set cannot be used to evaluate chunking methods. Table 3.2 gives an overview of which annotations are available for which corpus.

[24] In my topological field study (Ortmann 2020), I found that the POS error rate in the DTA sample ranges between 1.3% and 15% for the different texts (avg: 6.3%). For the sentence boundaries, I found F_1-scores between 54.1% and 100.0% (avg: 86.7%). Of course, it would be more realistic to evaluate on imperfect annotations because many corpora are not annotated manually. However, evaluating on incorrect data makes it difficult to judge whether an error is actually caused by the evaluated system or by errors in the data. To get a conclusive impression of the system's performance at a specific task, clean and correct gold data is indispensable.

[25] The manually annotated data set from our research project is also a subset of the DTA. However, there are multiple reasons why I do not use this sample here. Firstly, it is restricted to only two scientific genres (medicine, theology), which limits the generalizability of evaluation results. Secondly, the annotations that were created in our project are not exhaustive enough for my evaluation purposes. For example, to reduce manual annotation effort, only one *in situ* phrase per extraposed phrase is annotated, and not *all* phrases as required for a complete evaluation. Thirdly, no annotations of topological fields and chunks were created. And finally, the automatic POS annotations were corrected only toward the end of our project (cf. Chapter 8) and the sentence boundaries are not corrected at all, which makes them unsuitable for evaluation (see the previous footnote).

Corpora	#Docs	#Sents	#Toks	#Words	License
Modern					
TüBa-D/Z	3,816	104,787	1,959,450	1,671,198	Academic
Tiger	2,263	50,461	888,076	768,534	Academic
Spoken	14	28,696	296,942	239,897	Academic
Modern	78	559	7,642	6,369	CC BY-SA 3.0 and CC BY–NC–ND 4.0
Historical					
Mercurius	2	8,387	187,423	163,873	CC BY 3.0
ReF.UP	26	21,432	600,569	477,306	CC BY-SA 4.0
HIPKON	53	342	4,210	3,747	CC BY 3.0
DTA	29	629	18,885	16,114	CC BY-SA 4.0

Table 3.1.: Overview of the gold data sets that are used in this thesis. The number of words refers to all tokens that are not tagged as punctuation. The size of the respective train/dev/test sections is given in the annotation chapters.

Corpus	Sentence Brackets	Topological Fields	Chunks	Phrases	RelCs	Extra-position
Modern						
TüBa-D/Z	✓	✓	✓	✓	✓	✓
Tiger	x	x	✓	✓	✓	(✓ RelCs only)
Spoken	✓	✓	x	✓	✓	✓
Modern	✓	✓	✓	✓	✓	✓
Historical						
Mercurius	x	x	✓	✓	x	x
ReF.UP	x	x	✓	✓	x	x
HIPKON	✓	✓	✓	✓	✓	✓
DTA	✓	✓	✓	✓	✓	✓

Table 3.2.: Overview of the annotations that are available for each gold data set in this thesis.

CHAPTER 4

FairEval: Error Analysis and Fair Evaluation of Labeled Spans[26]

The annotations that are created in this thesis can be subsumed under the term 'labeled spans'. During my studies, I noticed that evaluating this type of annotation with traditional approaches can lead to undesirable consequences and requires additional effort to gain a real understanding of annotation quality. For example, low recall or precision values for the recognition of post-fields do not reveal if the annotation tool recognized any part of a post-field or no post-field at all, if the post-field was confused with a middle field or something else, etc. In addition, partly correct annotations are traditionally penalized as multiple errors, complicating the interpretation of the resulting scores.

In response to these observations, I developed a new evaluation method called *FairEval* (Ortmann 2022),[27] which produces more meaningful results and, at the same time, provides useful insights into the actual errors, strengths, and weaknesses of annotation systems. In this chapter, the details of the method are presented and its usage is demonstrated with different annotation examples. The method will be used for evaluation throughout the following chapters of this thesis.

4.1. Traditional Evaluation Metrics

Evaluation in NLP serves two main purposes: (i) determining how good a system is at a given task and comparing its performance to other systems, and (ii) analyzing the errors a system makes to be able to improve it. For 1:1 mapping tasks like POS tagging, the procedure is clear-cut. Every token receives exactly one tag, and the number of correctly assigned tags is compared to the incorrect tags and reported as accuracy, possibly accompanied by a confusion matrix.

For tasks that include the annotation of spans or do not require every token to receive an annotation, e.g., tokenization, named entity recognition (NER), or chunking, the incorrect annotations can be further divided into false positives (`FP`, superfluous annotations) and false negatives (`FN`, missing annotations). System performance, in this case, is measured by comparing the two types of incorrect annotations with the number of true positives (`TP`). The results are reported as recall (Eq. 4.1; 'how many annotations that should be present are actually there') and precision (Eq. 4.2;

[26] The content of this chapter is taken from my paper Ortmann (2022): *Fine-Grained Error Analysis and Fair Evaluation of Labeled Spans.*

[27] https://github.com/rubcompling/FairEval

'how many of the annotations that are present are actually correct'). Usually, there is a trade-off between precision and recall because improving one likely worsens the other.

$$Recall = \frac{TP}{TP + FN} \tag{4.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

The harmonic mean of precision and recall, better known as F_1-score, is consulted to compare different systems based on a single number (Eq. 4.3).

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4.3}$$

There are certain issues with these evaluation metrics, though (cf., e.g., Shao et al. (2017) for the task of word tokenization). In this chapter, I target the yet unsolved problem of double penalties for the annotation of labeled spans, which is highly relevant for the evaluation of all subsequent annotations in this thesis. In the next section (4.2), I will first give a short overview of the problem before I suggest a new approach to the evaluation of labeled spans in the remainder of the chapter.

4.2. The Problem with Traditional Evaluation

When labeled spans are evaluated in the traditional way, in trivial cases as displayed in example (12), one (missing) annotation counts as one true positive or one error, respectively.

(12)

	Target:	| A |	| A |	_
	System:	| A |	_	| A |
		1 TP	1 FN	1 FP

However, if a system annotates a span that overlaps with the correct annotation but is not identical to it, one annotation is counted as two errors as in example (13) because the target annotation is missing (FN), while another annotation is present (FP).

(13)

	Target:	| A |	| A |
	System:	| B |	| A |
		1 FN + 1 FP	1 FN + 1 FP

This phenomenon is especially undesirable since the annotations in (13) are closer to the target annotation than completely missing or superfluous annotations, i.e., FNs and FPs as in (12). Optimizing a system based on these metrics could thus encourage the system to skip difficult or uncertain cases because missing an annotation (FN) is punished less than getting it almost right (FN+FP). Intuitively, these close-to-correct errors should be punished equally or maybe even less than the errors in (12), and not vice versa.

Also, the traditional evaluation with only two error categories does not provide information about the actual weaknesses of a system, which is critically important for improving performance (Braşoveanu et al. 2018; Manning 2011). Instead, a manual error analysis would be necessary to distinguish between the two very different error types in example (13).

As most researchers are likely aware of this problem (cf., e.g., Jurafsky and Martin 2021), there have been various attempts to deal with it, e.g., by performing qualitative error analyses (Braşoveanu et al. 2018; Manning 2011), counting overlapping tokens or characters (Potthast et al. 2010), or introducing partial annotation scores or relaxed evaluation metrics (Röder et al. 2018; Ji and Nothman 2016). However, there is no universal solution yet, and the traditional metrics are still widely used for evaluating labeled spans despite their drawbacks.

In this chapter, I suggest an approach to a fairer evaluation of labeled spans that prevents double penalties for a single annotation and, at the same time, allows for a more fine-grained error analysis. First, Section 4.3.1 introduces new error types that help to distinguish between different kinds of overlapping spans. Section 4.3.2 then discusses ways to calculate precision, recall, and F_1-scores based on these error types. Afterwards, in Section 4.4, an algorithm for the identification of the different error types in flat and multi-level annotations is presented. Finally, Section 4.5 compares the results of traditional evaluation to fair evaluation for different types of annotations. The chapter concludes with a discussion in Section 4.6.

4.3. Fair Evaluation

The enterprise of this chapter was inspired by Manning (2006), who explicitly brings up the problem of double penalties in NER evaluation. Similar to the remarks above, he argues that one should not optimize NER systems for F_1 because the metric is dysfunctional for sparse annotations. Although he focuses on named entity recognition, the same also holds for other types of labeled spans, e.g., chunks or syntactic constituents. As an alternative, Manning (2006) suggests the distinction of different error types, which will be picked up and expanded upon in the next section (4.3.1). From his considerations, it remains unclear, though, how these error types should be used to compare different NLP systems, which is the topic of Section 4.3.2.

4.3.1. Fine-Grained Error Types

The traditional evaluation only considers true positives (`TP`), false positives (`FP`), and false negatives (`FN`). However, example (13) already pointed out that a restriction to the latter two error types does not reflect the actual annotation quality in the case of overlapping spans. `FP`s and `FN`s should therefore be used exclusively to refer to 1:0 and 0:1 mappings as displayed in example (12). For cases in which the system annotation overlaps with the target annotation but is not identical to it, Manning (2006) suggests the distinction of three additional error types:

`LE` (labeling error): Identical span, different label

`BE` (boundary error): Identical label, different (overlapping) span

`LBE` (labeling-boundary error): Different label, different (overlapping) span

The three additional error types are illustrated in example (14).[28] As intended, their application resolves the problem of double penalties because one annotation now counts as one error instead of two. Moreover, they enable a more detailed error analysis and allow to distinguish between entirely missing or superfluous annotations and almost correct annotations, which are often more frequent than actual `FPs` and `FNs` (Manning 2006; Ortmann 2021a; Ortmann 2021b).

(14)

Target:	| A |	| A |	| A |
System:	| B |	| A |	| B |
	1 `LE`	1 `BE`	1 `LBE`

In the case of boundary errors, it is possible to make the evaluation even more fine-grained by distinguishing whether the system's annotation is smaller ($\texttt{BE}_\text{s}$) or larger ($\texttt{BE}_l$) than the target span or whether it overlaps with it ($\texttt{BE}_\text{o}$). Example (15) displays the three sub-types of boundary errors, which provide even more details on a system's weaknesses, indicating possible starting points for improvement.[29]

(15)

Target:	| A |	|A|	| A |
System:	|A|	| A |	| A |
	1 $\texttt{BE}_\text{s}$	1 $\texttt{BE}_l$	1 $\texttt{BE}_\text{o}$

Annotations that overlap with two (or more) spans, at least one of which has the same label, should be counted as `BE` and not `LBE`. In total, for n target annotations and m system annotations, the number of true positives plus errors always lies between $max(n, m)$ and $n + m$. Both examples in (16) should thus yield three errors.

(16)

Target:	| A | B |	| A | B | B |
System:	| A | B | B |	| A | B |
	2 $\texttt{BE}_\text{s}$ + 1 $\texttt{BE}_\text{o}$	2 $\texttt{BE}_l$ + 1 $\texttt{BE}_\text{o}$

[28] In the literature, even more error types have been introduced. While some of them are only relevant to a specific annotation type (e.g., Braşoveanu et al. (2018) with an error taxonomy for Named Entity Linking), other categories like errors in the gold standard (Manning 2011) can only be recognized with a manual analysis. For practical reasons, these error types are not further discussed here. But if their frequency is known for a given data set, they could be integrated into the analysis and calculation of metrics similar to the error types presented in this chapter.

[29] Depending on the intended application, it would also be possible to distinguish whether one of the system boundaries, left or right, is identical to the target boundary to provide even more insight into the actual errors. For example, the evaluation of antecedents in Chapter 7.1 is a case where the alignment of the right boundary is highly relevant. The same distinctions as for boundary errors could also be made for labeling-boundary errors, but they would not provide much additional information since label and span of the system annotation both differ from the target. Therefore, `LBE` sub-types are not considered here.

4.3.2. Fair Precision, Recall, and F_1-Score

The fine-grained distinction of error types as described in Section 4.3.1 solves the problem of double penalties and enables a more detailed error analysis. However, the raw number of errors is unsuitable for comparing different systems, especially across different data sets. Instead, it would be desirable to include these error types in the calculation of precision, recall, and F_1-score. In Ortmann (2021a), I argued that the additional error types refer to an existing annotation and should therefore count as false positives for the calculation of F_1-scores. Read et al. (2012), instead, count these kinds of errors as false negatives to prevent double penalties. For the resulting F_1-score, the decision makes no difference since, mathematically, F_1 only depends on the number of true positives and errors (cf. Eq. 4.4).

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{(2 * TP) + errors} \tag{4.4}$$

However, counting the overlapping errors as either `FPs` or `FNs` makes recall and precision values hard to interpret in a meaningful way. As each of the error types indicates a (partly) missing target annotation and, at the same time, a (partly) incorrect system annotation, it seems more appropriate to count the new error types as half `FP` and half `FN` (cf. Eq. 4.5).

$$1LE = 1BE = 1LBE = 0.5FP + 0.5FN \tag{4.5}$$

As explained above, this does not change the F_1-score, but it renders precision and recall values more meaningful again.

Weighted Evaluation Depending on the application, it could also be useful to make the evaluation more nuanced by introducing specific weights for different error types. For example, boundary errors could be considered less severe, e.g., in a search context because the target span is still found by the system. In this case, `BEs` could, for example, be counted as 50% true positives as in equation (4.6).

$$1BE = 0.5TP + 0.25FP + 0.25FN \tag{4.6}$$

When different types of boundary errors are distinguished, the evaluation could be even more differentiated (cf., e.g., Eq. 4.7) to more precisely reflect true annotation quality in precision and recall. I will use weighted metrics for the evaluation of antecedents in Chapter 7.1. It is important to note, though, that contrary to equation (4.5), the weighting in equations (4.6) and (4.7) also affects F_1-scores because it increases the total number of `TPs`.

$$\begin{aligned} 1BE_\mathrm{s} &= 0.5TP + 0.5FN \\ 1BE_l &= 0.5TP + 0.5FP \\ 1BE_\mathrm{o} &= 0.5TP + 0.25FP + 0.25FN \end{aligned} \tag{4.7}$$

Algorithm 1: Traditional error type identification

Input: A set of target spans T and system spans S. Spans are triples of label l, begin b, and end e

Output: Number of TP, FP, and FN per label and overall

1: Count every span $t \in T \cap S$ as TP for l_t
2: Count every span $t \in T \setminus S$ as FN for l_t
3: Count every span $s \in S \setminus T$ as FP for l_s
4: Sum up TPs, FPs, and FNs across labels
5: **Return** results per label and overall

4.4. Algorithm for Error Identification

For traditional evaluation with only two error categories (false positives and false negatives), the algorithm to identify error types is simple. If a target span was recognized by the system, it counts as TP. Spans only present in the system output are FPs, and target spans missing in the system annotation are FNs (cf. Algorithm 1). The different categories can be identified for individual labels or all labels overall. Identifying the fine-grained error types is more complicated. In particular, there are the following difficulties:

(i) One target span can overlap with more than one system span and vice versa. Nevertheless, the number of TPs plus errors should always lie between $max(n, m)$ and $n + m$ for n target and m system annotations, i.e., every system annotation and every target annotation should count exactly once. To achieve this, spans are removed from the input list as soon as their first matching counterpart is found. To ensure that other potential counterparts are also matched to the correct span, the algorithm must keep track of the already matched tokens in each span. In combination, these two steps allow matching multiply overlapping spans to their correct counterparts without counting any span twice.

(ii) There are cases in which one span could correspond to different error types, e.g., BE and LBE as in example (16). As described in Section 4.3.1, BEs should be preferred over LBEs. The algorithm will therefore proceed in four incremental steps, starting with easy to identify spans with 1:1 mappings (TPs and LEs, step 1), followed by boundary errors (step 2), and labeling-boundary errors (step 3), and finally the remaining 1:0 and 0:1 mappings (FNs and FPs, step 4).

(iii) Per-label evaluation is also less straightforward for the more fine-grained error types. There are two main problems:

1. Which error type should be assigned to multiply overlapping spans? In the following, each span is counted as the first matching error type.
2. Should LE and LBE count as errors for the target or the system label? One possible solution is to put the focus on the target labels, i.e., to evaluate how the target labels

were annotated by the system. In this case, all errors (except false positives) count for the label of the target span they are matched to. The resulting error distribution then gives a detailed picture of how well the target spans were identified by the system. If the focus is on the system annotation, the same process could be applied to the system labels. A confusion matrix can represent both directions at the same time.

(iv) In evaluating hierarchical annotations (e.g., constituency trees), it is common practice to compare the annotated spans and labels while ignoring the hierarchical structure.[30] For example, an `NP` is considered correct if it spans across the correct tokens, independently of the presence or absence of intervening nodes like adjective phrases, etc. The same also applies to other multi-level annotations, e.g., several stacked entities. Hence, the traditional evaluation from Algorithm 1 works just the same for nested spans as for flat annotations.

The identification of the fine-grained error types, specifically `BEs` and `LBEs`, in nested structures is more complicated because it is not always clear which spans should be compared with each other. While the classification is likely no problem for humans in most of the cases, an algorithm will sometimes only approximate the optimal match of system and target spans if it shouldn't become too complex or slow. Here, two practical decisions are made:

1. It is known that systems are usually more accurate at identifying shorter spans compared to longer ones (cf. Bastings and Sima'an (2014) on constituency parsers). Therefore, in each step, the algorithm starts with the shortest span to speed up the search for the correct match of system and target annotation.
2. If one span can be matched to two (or more) other spans, the most similar one is considered first. Similarity, here, is defined as the maximum number of shared tokens and the fewest differing tokens. If multiple spans are equally similar, the shortest one is chosen. If multiple spans are still equally similar, the first one in the input is taken, which corresponds to the left-most one if sentences are read from left to right.

Based on the previous considerations, Algorithm 2 identifies the fine-grained error types from Section 4.3.1 in flat and hierarchical spans. The resulting error counts can be used to calculate precision, recall, and F_1-score as detailed in Section 4.3.2. Table 4.2 shows an example of the algorithm's output.[31]

[30] For the evaluation of tree structures, other approaches also exist that take into account the complete paths within the tree, e.g., the leaf-ancestor metric or dependency-based metrics, cf. Rehbein and Genabith (2007). Although these metrics are more robust against differences in annotation schemes, the PARSEVAL metric (Black et al. 1991) is still commonly used for parser evaluation.

[31] A reference implementation of the algorithm as well as the data sets and detailed results from Section 4.5 were provided in the paper's repository at `https://github.com/rubcompling/FairEval`. The code is also integrated into the CLASSIG pipeline that accompanies this thesis.

Algorithm 2: Identification of fine-grained error types in labeled spans

Input: A list of target spans T and system spans S, sorted by span length from shortest to longest. Each span is a 4-tuple of label l, begin b, end e, and a set of included tokens $toks$ $(1..n)$. $b = e$ for spans of length 1.

Output: Number of TP, FP, LE, BE, $\texttt{BE}_\texttt{s}$, $\texttt{BE}_l$, $\texttt{BE}_\texttt{o}$, and FN per label and overall

Function definitions:

Let $\texttt{BE}_{\text{type}}(t, s)$ return the correct type of $\texttt{BE}_\texttt{s}$, $\texttt{BE}_l$, and $\texttt{BE}_\texttt{o}$ for spans t and s
Let $get_{\texttt{BE}}(t, s \in S)$ return the most similar span $s \in S$ for t with $l_\text{t} = l_\text{s}$ and $|toks_\text{t} \cap toks_\text{s}| \geq 1$
Let $get_{\texttt{LBE}}(t, s \in S)$ return the most similar span $s \in S$ for t with $l_\text{t} \neq l_\text{s}$ and $|toks_\text{t} \cap toks_\text{s}| \geq 1$
Let $update_{\text{toks}}(toks_\text{t}, toks_\text{s})$ set $toks_\text{t} = toks_\text{t} \setminus toks_\text{s}$ and $toks_\text{s} = toks_\text{s} \setminus toks_\text{t}$
Let $move(t, T \rightarrow M)$ remove t from T and add it to M

Step 1: Count 1:1 mappings (true positives and labeling errors)

Count identical spans $t \in T = s \in S$ as TP for l_t and remove t from T and s from S
Count spans with $l_\text{t} \neq l_\text{s}$, $b_\text{t} = b_\text{s}$, $e_\text{t} = e_\text{s}$ as LE for l_t and remove t from T and s from S

Step 2: Count boundary errors

Create empty lists M_t and M_s for matched spans

For each $t \in T$:
    Count $get_{\text{BE}}(t, s \in S)$ as $\texttt{BE}_{\text{type}}(t, s)$ for l_t
    Update matches: $update_{\text{toks}}(toks_\text{t}, toks_\text{s})$, $move(t, T \rightarrow M_\text{t})$, and $move(s, S \rightarrow M_\text{s})$

For each $t \in T$:
    Count $get_{\text{BE}}(t, s \in M_\text{s})$ as $\texttt{BE}_{\text{type}}(t, s)$ for l_t
    Update matches: $update_{\text{toks}}(toks_\text{t}, toks_\text{s})$ and $move(t, T \rightarrow M_\text{t})$

For each $s \in S$:
    Count $get_{\text{BE}}(s, t \in M_\text{t})$ as $\texttt{BE}_{\text{type}}(t, s)$ for l_t
    Update matches: $update_{\text{toks}}(toks_\text{t}, toks_\text{s})$ and $move(s, S \rightarrow M_\text{s})$

Calculate $\texttt{BE} = \texttt{BE}_\texttt{s} + \texttt{BE}_l + \texttt{BE}_\texttt{o}$

Step 3: Count labeling-boundary errors

For each $t \in T$:
    Count $get_{\text{LBE}}(t, s \in S)$ as LBE for l_t
    Update matches: $update_{\text{toks}}(toks_\text{t}, toks_\text{s})$, $move(t, T \rightarrow M_\text{t})$, and $move(s, S \rightarrow M_\text{s})$

For each $t \in T$:
    Count $get_{\text{LBE}}(t, s \in M_\text{s})$ as LBE for l_t
    Update matches: $update_{\text{toks}}(toks_\text{t}, toks_\text{s})$ and $move(t, T \rightarrow M_\text{t})$

For each $s \in S$:
    Count $get_{\text{LBE}}(s, t \in M_\text{t})$ as LBE for l_t
    Update matches: $update_{\text{toks}}(toks_\text{t}, toks_\text{s})$ and $move(s, S \rightarrow M_\text{s})$

Step 4: Count false positives and negatives

Count every $t \in T$ as FN for l_t
Count every $s \in S$ as FP for l_s

Step 5: Return results per label and the overall sum across labels

4.5. Example Evaluation

To illustrate the application and results of the new evaluation algorithm, in this section, it is applied to three different tasks that require the identification of labeled spans: NER, chunking, and topological field parsing.

1. Named Entity Recognition (NER)

- **Annotation:** Named entities are phrases that refer to entities such as people or places by means of a proper name (Tjong Kim Sang and De Meulder 2003). The annotation is sparse, i.e., the majority of tokens does not receive a label. Multi-level annotations are possible but not considered here.
- **NLP tool:** The NER component of the Stanza pipeline (Qi et al. 2020) with the `germeval2014` model.[32]
- **Data:** The test partition of the GermEval 2014 data set (Benikova et al. 2014). Since Stanza does not support multi-level annotation, only top-level entities from the four main classes are evaluated, yielding 6.178 named entities.

2. Chunking

- **Annotation:** Chunks are non-recursive, non-overlapping constituents from a sentence's parse tree (Sang and Buchholz 2000). Contrary to NER, most tokens receive a label. The annotation is non-hierarchical per definition.
- **NLP tool:** The neural sequence labeling tool NCRF++ (Yang and Zhang 2018)[33] with a model from Ortmann (2021a). The model was trained on the German newspaper corpus TüBa-D/Z (Telljohann et al. (2017); 80% training, 10% development data)[34] with characters, tokens, and POS tags as features and pre-trained word embeddings (cf. Chapter 6.1).
- **Data:** The remaining 10% of the TüBa-D/Z corpus with 101.304 chunks of 16 different types.

3. Topological Field Parsing

- **Annotation:** Topological fields are linear syntactic structures on the clause level of German sentences. Fields can be understood to form a tree structure, i.e., the annotation is hierarchical, and most tokens receive a label.

[32] Stanza version 1.2, `https://stanfordnlp.github.io/stanza/`
[33] `https://github.com/jiesutd/NCRFpp`
[34] Release 11.0, chunked version

- **NLP tool:** The unlexicalized Berkeley parser (Petrov et al. 2006)[35] with a model from Ortmann (2020). It was trained on 80% of the TüBa-D/Z corpus (Telljohann et al. (2017); cf. Chapter 5).[36]
- **Data:** 10% of the TüBa-D/Z corpus with 63.824 fields of 13 different types.

The results for traditional vs. fair evaluation of the three annotation tasks are displayed in Table 4.1. F_1-scores differ between 1–3.7 percentage points. The largest difference is observed for the recognition of named entities, while the difference is smallest for chunks. Except for NER, recall values differ slightly more between evaluation methods than precision. With respect to the labels, the largest differences are found for entities of type ORG, adjective and foreign language chunks, and coordination and post-fields.

Figure 4.1[37] shows the distribution of error types for the three annotation tasks according to traditional and fair evaluation. For NER and chunking, the traditional evaluation identifies 44% of the errors as `FP` and 56% as `FN`, while for topological field parsing, `FPs` are more frequent with 54% compared to 46% `FNs`. However, when the more fine-grained error types are considered, the rate of actual false positives and negatives shrinks substantially. The highest proportion of actual `FNs` is observed for the sparse NER annotation and the highest proportion of actual `FPs` for the hierarchical fields. For chunking, actual `FPs` and `FNs` together make up only 2% of all errors.

On the other hand, boundary errors, which traditionally count as two errors (1 `FP` and 1 `FN`), make up between 14% (NER) and 59% (chunking) of the errors. In most of these cases, the system annotation includes the target span (57%–73%) or vice versa (27%–42%). Errors of type $\texttt{BE}_\circ$ are extremely rare. Interestingly, labeling errors occur especially for the sparse named entities: 30% of the errors are due to entities that were recognized correctly but assigned the wrong label. Labeling-boundary errors are more frequent for chunking (34%).

So, although traditional evaluation suggests similar or even identical error distributions for the three tasks, an analysis of the fine-grained error types reveals that the systems actually make very different kinds of errors. While the NER system should be optimized especially for assigning the correct label and reducing the number of missing entities, the other two systems can gain more by improving the accuracy of span boundaries.

Another advantage of fair evaluation concerns the results for individual labels (cf., e.g., Table 4.2 for NER). While traditional evaluation only counts true positives and (seemingly) missing and superfluous spans without capturing the actual relation between system and target annotation, the fine-grained error structure of fair evaluation also enables the creation of a confusion matrix (cf. Figure 4.2). For system developers and linguists alike, this matrix provides valuable information about which labels are confused most often and which labels contribute to which error types. For example, organizations are the most frequently overlooked named entities, noun chunks are the main source of boundary errors, and middle fields are especially prone to have incorrect boundaries or be false positives.

[35] https://github.com/slavpetrov/berkeleyparser
[36] Release 11.0, CoNLL-U Plus version
[37] The plots in this thesis have been created with the R package *ggplot2*, https://ggplot2.tidyverse.org/.

		Precision	Recall	F_1
NER	*Trad.*	86.66	83.51	85.05
	Fair	90.42	87.23	88.80
Chunks	*Trad.*	97.20	96.39	96.79
	Fair	97.86	97.86	97.86
Topol. Fields	*Trad.*	93.41	94.27	93.84
	Fair	94.78	95.92	95.35

Table 4.1.: Results for traditional vs. fair evaluation of the different annotation tasks in percent.

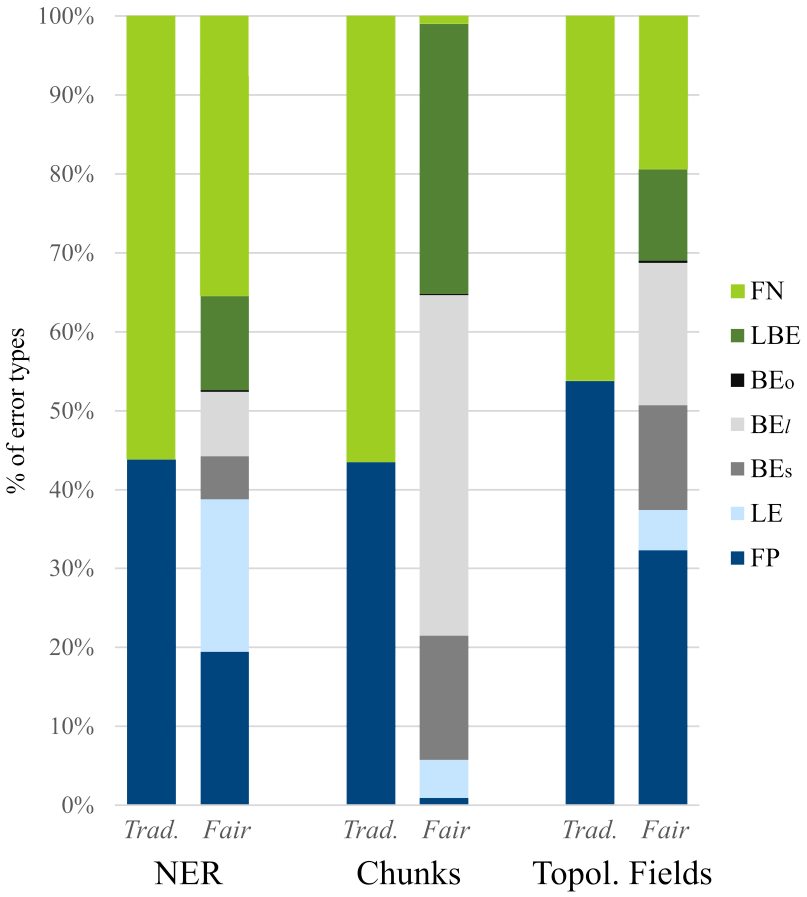


Figure 4.1.: Distribution of error types for the three annotation tasks according to traditional vs. fair evaluation.

Label	TP	FP	LE	BE				LBE	FN	Prec	Rec	F_1
				BE_s	BE_l	BE_o	BE_{all}					
LOC	2132	81	56	29	28	0	57	40	98	93.12	92.43	92.78
ORG	1002	87	76	16	27	0	43	48	167	85.46	80.00	82.64
OTH	473	48	89	15	26	3	44	44	142	77.60	67.24	72.05
PER	1552	37	31	11	25	0	36	23	55	94.98	93.95	94.46
Overall	5159	253	252	71	106	3	180	155	462	90.42	87.23	88.80

Table 4.2.: Raw frequencies of TPs and errors in the NER annotation per label and overall as output by Algorithm 2. In addition, the rightmost columns show fair precision, recall, and F_1 values for individual labels.

4.6. Discussion

Evaluation serves the purpose of comparing and improving NLP systems, but optimizing systems for the traditional metrics can lead to undesirable effects due to double penalties for close-to-correct annotations. In this chapter, I have presented an algorithm for the identification of more fine-grained error types in flat and multi-level annotations of labeled spans to ensure that every annotation counts only once. The algorithm was supplemented by a suggestion on how to calculate meaningful precision, recall, and F_1-scores based on these error types. In combination, the described procedure allows for a more realistic evaluation, which prevents double penalties while, at the same time, providing more information about possible improvements.

The exemplary application to three different annotation tasks has illustrated that annotations that look the same through the lens of traditional evaluation can actually result from very different error distributions. Future studies should consider using the presented algorithm to optimize systems for sensible metrics and gain more insight into their actual weaknesses. In the remainder of this thesis, the new method will be used to evaluate the different annotations and I will report *FairEval* scores unless stated otherwise.

However, the comparison has also shown that F_1-scores are higher according to the new evaluation method because labeling and boundary errors are no longer multiply penalized. Since the objective of the algorithm is not to make systems 'look better', results that are gained in this way should be reported alongside established evaluation metrics to ensure comparability. I will include traditional evaluation results for the different annotations in the appendix.

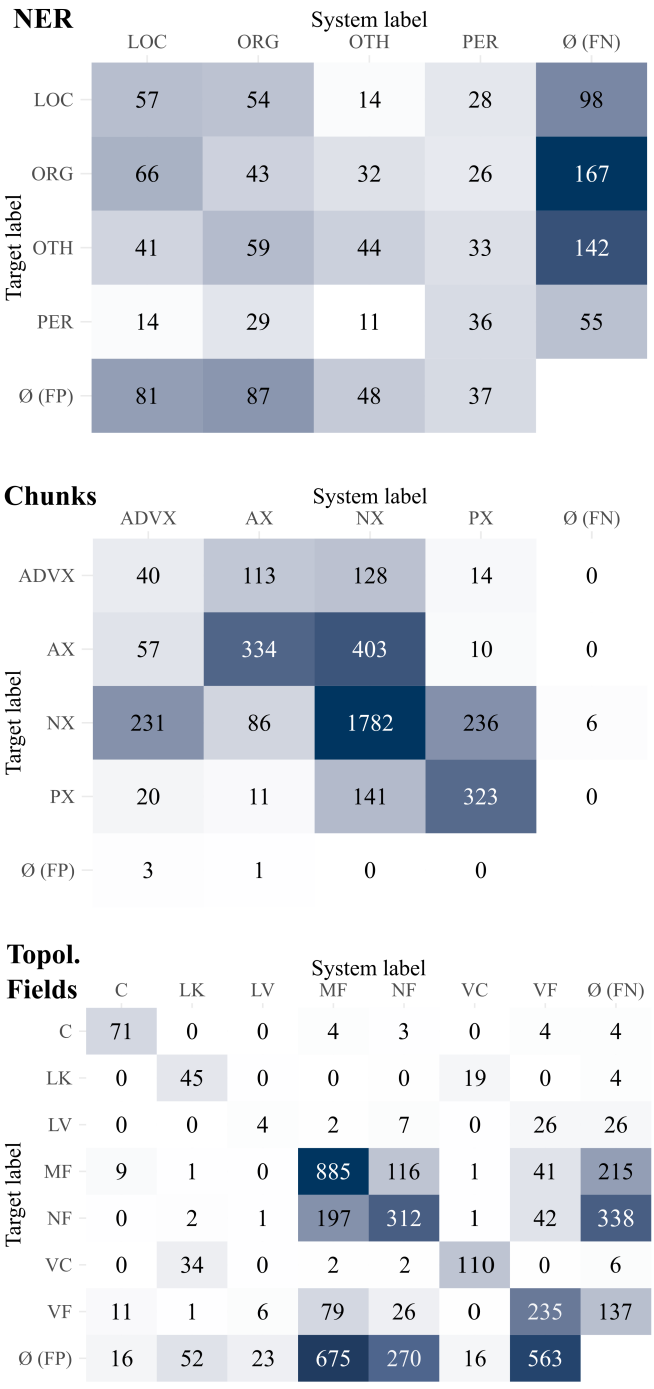


Figure 4.2.: Confusion matrices for the (main) labels of each annotation task. Only errors are included, i.e., the diagonal displays boundary errors. False positives and negatives are shown in the bottom row and the right-most column, respectively. The remaining cells represent labeling and labeling-boundary errors.

CHAPTER 5

Topological Field Analysis[38]

As defined in Chapter 2, a constituent is considered extraposed if it has been 'moved' from its base position in the middle field of the sentence to the post-field. Consequently, identifying topological fields is the first prerequisite for the automatic analysis of extraposition. In this chapter, my exploration of topological field parsing in modern and historical German is presented. Section 5.1 begins with an overview of the topological field model and explains the tagset used in this thesis. Section 5.2 summarizes the related work on automatic topological field identification, and Section 5.3 describes the training and test data for the studies. The implementation is split into two steps, starting with the identification of sentence brackets in Section 5.4 before adding the other topological fields in Section 5.5. The chapter concludes with a discussion in Section 5.6.

5.1. The Topological Field Model

The topological field model (Höhle 2019; Wöllstein 2018) is a widely used theory-neutral framework for the description of syntactic structures in German sentences. While German is considered to have a relatively free word order, the topological fields provide a clear structure on the clause level. In German, there are three different clause types, which are characterized by the position of the finite verb. Figure 5.1 illustrates the linear order of fields for verb-first (V1), verb-second (V2), and verb-last (VL) clauses. In this thesis, a simplified version of the annotation scheme suggested by Telljohann et al. (2017) is used.[39] The following fields are considered:

VF The pre-field (*Vorfeld*) of the sentence is obligatory in V2 clauses and consists of exactly one constituent.[40] Often this is the subject, but it can also be almost any other, possibly complex constituent, e.g. conditional clauses.

[38] The content of this chapter is adapted from my paper Ortmann (2020): *Automatic Topological Field Identification in (Historical) German Texts* and complemented with an unpublished pilot study on sentence bracket identification, additional test data, the `News1` model from Chapter 6.2, and *FairEval* results.

[39] There are different opinions about the boundaries of specific fields, e.g., whether relativizers belong to the left bracket or the middle field and if certain constituents are placed in the middle field or the post-field when the right bracket is empty. For practical reasons, I will follow the guidelines by Telljohann et al. (2017) because the only available training resources are annotated according to their scheme.

[40] For possible exceptions, consider the discussion in Zifonun et al. (1997).

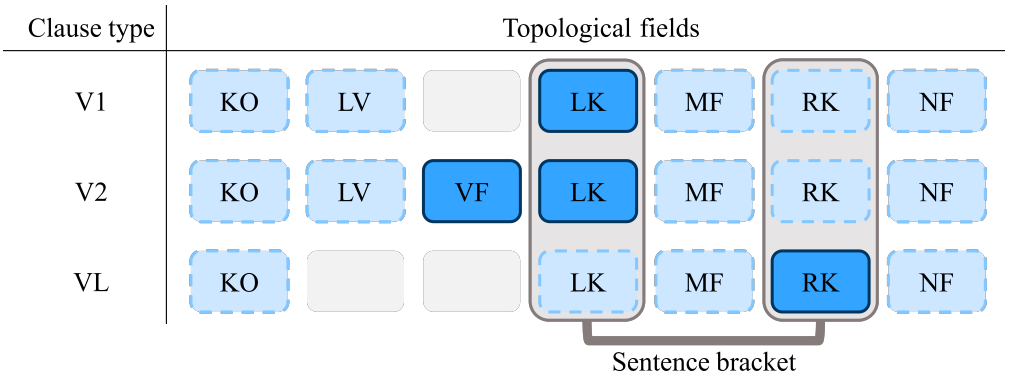


Figure 5.1.: Simplified topological field model for verb-first (V1), verb-second (V2), and verb-last (VL) clauses with mandatory (*blue*) and optional fields (*light blue*, dashed lines). Positions that are never occupied are colored in *light gray*. The coordination field KOORD is abbreviated to KO here.

LK The left sentence bracket (*Linke Klammer*) is obligatory in V1 and V2 clauses and optional in VL clauses. In V1 and V2 clauses, it contains a single finite verb, whereas in VL clauses the position can, instead, be filled with a complementizer and, hence, is often also referred to as C. Following Telljohann et al. (2017), it can be occupied by subordinating conjunctions and relative and interrogative pronouns or phrases.

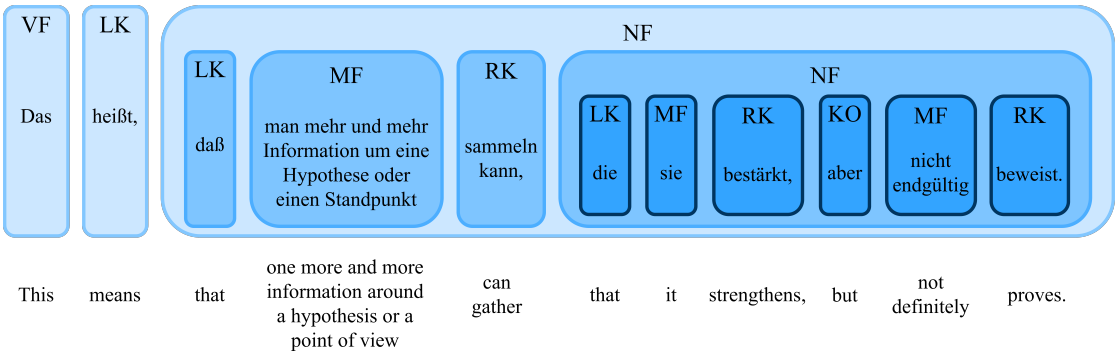
MF The middle field (*Mittelfeld*) is surrounded by the LK to the left and/or the RK to the right and can contain any number of constituents. Here, it subsumes the MF and MFE fields from Telljohann et al. (2017).

RK The right sentence bracket (*Rechte Klammer*) is often referred to as verb complex, as it contains the non-finite verbs, verb particles, and in VL clauses also the finite verb. In this thesis, the label RK is used for the VC and VCE fields from Telljohann et al. (2017).

NF The post-field (*Nachfeld*) is located to the right of the (possibly empty) RK and can contain any number of constituents. While it is the default position for certain types of subclauses, it often also comprises other 'heavy' elements like relative clauses that are extraposed from the middle field.

KOORD The coordination field (*Koordinationsfeld*) subsumes the KOORD and PARORD fields from Telljohann et al. (2017) and contains all conjunctions that coordinate sentences, clauses, or fields. The conjuncts themselves are not evaluated here.

LV Left dislocations (*Linksversetzung*) contain material that is moved in front of the pre-field.



'This means that one can gather more and more information around a hypothesis or a point of view that strengthens it but does not definitely prove it.'

Figure 5.2.: Example sentence from the Modern data set with nested topological fields. The coordination field KOORD is abbreviated to KO here.

Except for the sentence brackets and the coordination field, all fields may contain embedded clauses. Figure 5.2 shows an example annotation with nested topological fields from the Modern data set of this thesis.

When applying the topological field model to data from different time periods, it is important to note that the sentence structure in historical texts was still subject to change and might differ substantially from modern data. As already mentioned in Chapter 2, especially the right sentence bracket only emerged over time. Nevertheless, it seems reasonable to annotate historical data with the positioning of the elements that make up the sentence brackets and surrounding fields today, as this eventually allows studying their development from predecessor structures over time.

5.2. Related Work

There have been a number of different approaches to the automatic identification of topological fields in German. The first studies (Neumann et al. 2000; Müller and Ule 2002; Hinrichs et al. 2002) used rule-based approaches, implemented with finite-state cascades, to identify the sentence brackets and, based on this, the other topological fields. For this rule-based approach, Neumann et al. (2000) report an overall F_1-score of about 87%. Veenstra et al. (2002) show that for sentence brackets, i.e., fields that contain a very restricted set of elements, such rule-based systems can yield competitive results.

For the identification of more complex topological fields and embedded clauses, using (probabilistic) parsers seems more promising: Becker and Frank (2002) train a non-lexicalized chart parser on a probabilistic context-free grammar and achieve labeled recall and precision values of about 93%. The highest values are observed with >99% for left sentence brackets and about 96% for right brackets. Klatt (2004) describes a bi-directional bottom-up parsing approach for non-

recursive topological field recognition, resulting in an overall F_1-score of about 95%. Kok and Hinrichs (2016) treat topological field annotation as a sequence labeling task. They use a bi-directional LSTM and achieve an overall accuracy of 97% for non-recursive topological field identification. For recursive topological field annotation, Cheung and Penn (2009) apply the Berkeley parser (Petrov et al. 2006) and report F_1-scores of 95% on the Tüba-D/Z corpus and 91% on the NEGRA corpus. They observe the best results for sentence brackets with F_1-scores >98%. F_1-scores of about 95% or more are also achieved for coordinations and the pre- and middle field. The post-field is recognized less reliably with about 83%, and left dislocations with only 7%. All of these approaches focus on standard German (newspaper) text.

Prior to my topological field study (Section 5.5; Ortmann 2020), there had been only two attempts to automatically identify topological fields in historical data. Using CoNLL-RDF and SPARQL, Chiarcos et al. (2018) implement a deterministic rule-based parser for topological field identification in Middle High German. It relies on grammars and expert knowledge and makes use of the manual annotations provided in the Reference Corpus of Middle High German (ReM; Klein et al. 2016). However, in the absence of a manual gold standard annotation, the accuracy of the parser is not evaluated and thus remains unclear. Hinrichs and Zastrow (2012) annotate texts from the German Gutenberg project from the 13th to 20th century with the Berkeley parser, trained on the TüBa-D/Z corpus. Since they do not provide evaluation results for the syntactic analysis, the annotation quality of their TüBa-D/DC corpus remains unclear as well. The results that are reported in this chapter thus offer the first evaluation of automatic topological field analysis for historical German.

Applications of topological field annotation range from improving POS tagging (Müller and Ule 2002), chunking (Hinrichs et al. 2002), and HPSG parsing (Frank et al. 2003) to improving anaphora resolution (Becker and Pecourt 2002), machine translation of idiomatic phrases (Anastasiou and Čulo 2007), and dependency parsing (Kok and Hinrichs 2016). In this thesis, the topological field analysis is used to identify extraposition (Chapter 7).

5.3. Data

Although the topological field model is widely used for the description of syntactic structures in German, only few corpora provide topological field annotations. The Tüba-D/Z corpus (Telljohann et al. 2017) is the largest available data set with manually annotated topological fields. Discounting headlines and other fragments, which do not receive a topological field annotation, the training section contains about 74k sentences with 491k fields. Table 5.1 gives an overview of the training data. The test section of the TüBa-D/Z corpus comprises 9k sentences with about 61k fields, including 27k sentence brackets. Most of the studies described in Section 5.2 use previous versions of this corpus for training and/or evaluation.

To investigate how well the automatic identification of topological fields can be transferred to other domains, two additional data sets for modern German are included in my studies. Discounting fragments, the TüBa-D/S corpus (Hinrichs et al. 2000) comprises 19k sentences with 107k fields (45k sentence brackets). The Modern data set (Ortmann et al. 2019) contains 462 sentences that

Model	#Docs	#Sents	#Toks	#Fields
`Punct`	3,075	73,884	1,534,476	491,806
`NoPunct`	3,075	73,884	1,316,329	491,806
`News1`	3,075	83,515	1,566,250	491,806

Table 5.1.: Overview of the TüBa-D/Z training data for each model in this chapter. For models `Punct` and `NoPunct`, only sentences with a topological field annotation are included. Model `News1` is equivalent to the model for constituency parsing in Chapter 6.2 and contains only sentences with a constituency parse. The number of fields refers to fields of the seven types from Section 5.1, even though the original field labels from Telljohann et al. (2017) are used during training and mapped to the smaller tagset of this thesis for evaluation (cf. Section 5.4.1). For the `NoPunct` model, all tokens tagged as punctuation have been removed from the training set.

Corpus	#Docs	#Sents	#Toks	#Words	#Brackets
TüBa-D/Z	364	9,240	189,038	161,893	27,432
Spoken	14	19,522	263,294	218,561	45,245
Modern	78	462	7,224	6,095	1,311
HIPKON	53	342	4,210	3,747	804
DTA	29	417	16,307	14,063	2,245

Table 5.2.: Overview of the test data for sentence bracket identification. Only sentences containing at least one sentence bracket are included in the evaluation.

are not fragments, with a total of 3k fields (1.3k sentence brackets).

Besides the modern data, two historical German corpora are used to assess whether topological fields can be identified automatically in texts from different time periods – without any historical training data available. The HIPKON corpus (Coniglio et al. 2014) contains 342 annotated sentences with 1.8k fields, including 800 sentence brackets. Because HIPKON was created for the investigation of post-fields, only sentences with a post-field are annotated. The second historical data set DTA (BBAW 2021) includes 417 sentences with 4.8k topological fields (2k sentence brackets). Tables 5.2 and 5.3 give an overview of the test data for sentence bracket identification and topological field annotation, respectively.

Figure 5.3 displays the distribution of topological fields in the test data. While the historical data sets contain about equal proportions of left and right brackets, left sentence brackets are more frequent in the modern data sets. Especially the spoken data set contains a lower proportion of right sentence brackets (31% of the brackets), suggesting more empty right brackets or generally less complex sentences. An indication of the latter is the slightly lower proportion of post-fields in the spoken data with less than 6%, compared to newspaper (7%) or other modern data (8%). The historical data sets exhibit more post-fields, with about 10% in the DTA and almost 19% in the

Corpus	#Docs	#Sents	#Toks	#Words	#Fields
TüBa-D/Z	364	9,276	189,352	162,149	61,779
Spoken	14	19,522	263,294	218,561	107,545
Modern	78	462	7,224	6,095	3,052
HIPKON	53	342	4,210	3,747	1,858
DTA	29	417	16,307	14,063	4,838

Table 5.3.: Overview of the test data for topological field parsing. Only sentences containing at least one field of the seven types from Section 5.1 are included in the evaluation.

HIPKON corpus. Due to its special focus on the post-field, HIPKON also has a lower proportion of middle fields than the other data sets, i.e., more cases in which the right bracket immediately follows the left one. Left-dislocations and coordination fields are the least frequent fields, especially in the modern newspaper data.

5.4. Identification of Sentence Brackets

As a first step towards the automatic topological field analysis in (historical) German, I started with the identification of sentence brackets. The schematic depiction of the topological field model in Figure 5.1 illustrates the central role of the sentence brackets. Not only do they determine the sentence structure (V1, V2, or VL). They also delimit the other fields, thus providing the basis for a more in-depth syntactic analysis of a given sentence by reducing ambiguity for further annotation. In addition, the brackets can only contain a limited set of elements, which presumably makes them easier to recognize for automatic algorithms and may ensure the necessary accuracy for subsequent annotations and analyses. If it was possible to automatically identify the sentence brackets in historical texts with high precision, this would be an important first step towards a complete topological field analysis of historical data.

In this section, I first report the results of an unpublished pilot study in which I tested different annotation methods for identifying the left and right sentence bracket (Section 5.4.1). The remainder of the section (5.4.2) then focuses on one annotation method and replicates the pilot study with fair evaluation on the data sets from this thesis.

5.4.1. Pilot study

As described in Section 5.2, various methods have been successfully applied to the recognition of sentence brackets (and other topological fields) in modern German. Since no prior results are available for historical German and no historical training data exists, I conducted an exploratory pilot study to identify suitable approaches for sentence bracket annotation in both modern and historical data.

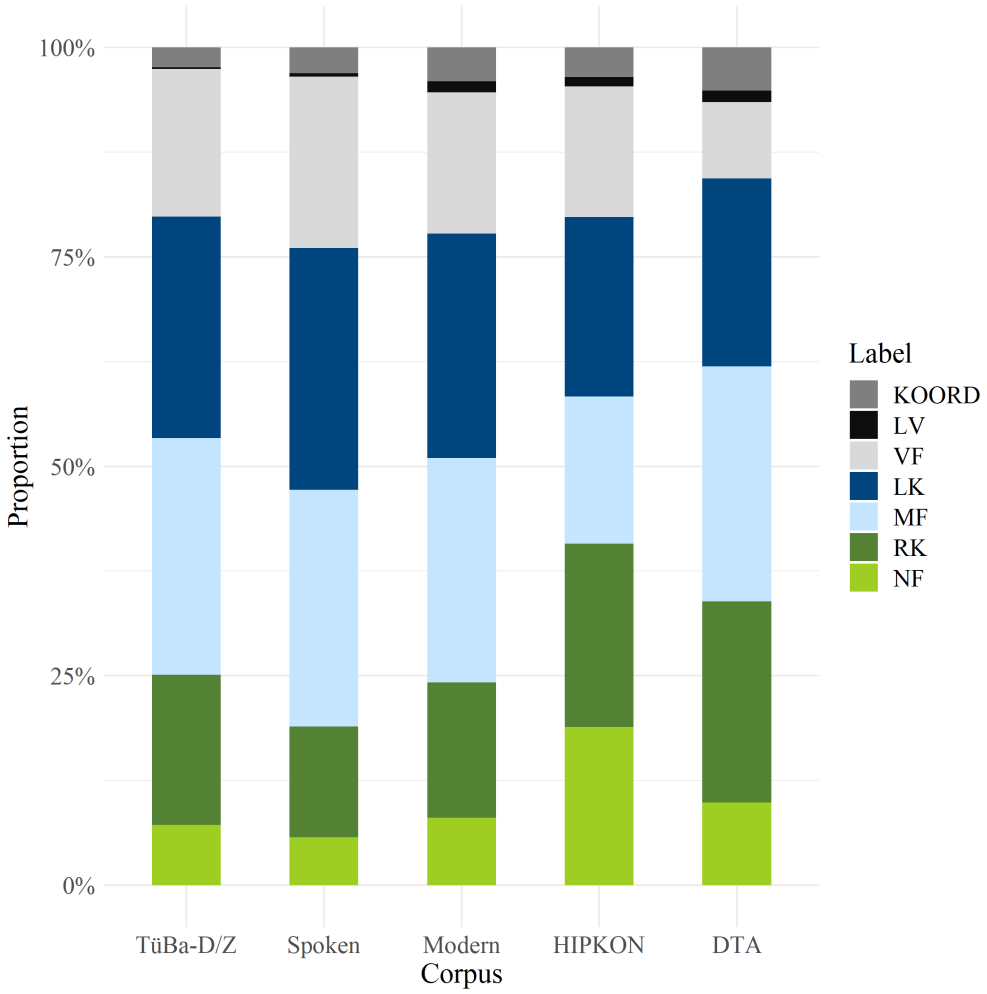


Figure 5.3.: Distribution of topological fields in the test data from Table 5.3.

Table 5.4 gives an overview of the two modern and two historical test sets that were used in the pilot study. The compilation of data sets slightly differs from Table 5.2. In particular, the pilot study did not include the Spoken data, and I used a different, larger sample from the German Text Archive with 4k sentences and 40k tokens that were manually tagged as part of a sentence bracket. For practical reasons, I did not re-run the complete pilot study with other data sets. Instead, I report the original results and focus on selecting a suitable method, followed by updated experiments with the selected method and the data from Table 5.2 in Section 5.4.2.

Corpus	#Docs	#Sents	#Toks	#LK	#RK
TüBa-D/Z	364	10,527	199,691	16,885	14,886
Modern	78	559	7,642	845	622
HIPKON	53	342	4,210	407	464
DTA_{Pilot}	10	3,894	208,489	17,229	23,109

Table 5.4.: Overview of the test data from the pilot study. #LK and #RK are the numbers of tokens that are labeled as (part of a) left sentence bracket and right sentence bracket, respectively.

Methods

Sentence brackets are non-complex topological fields that contain zero or more tokens but no (recursively) embedded fields. The task of identifying sentence brackets can thus be understood as either a tagging or a parsing problem. Consequently, the pilot study includes taggers and parsers and probabilistic methods as well as a few rule-based approaches, most of which require expert knowledge for creating rules. Table 5.5 gives an overview of the different methods. All probabilistic methods are trained on the TüBa-D/Z training set with the original tagset from Telljohann et al. (2017), which is mapped to the tags from Section 5.1 for evaluation. Becker and Frank (2002) observe that this approach of training on more fine-grained and evaluating on coarser categories can improve the results of probabilistic topological field parsers.

As word forms differ substantially between modern and historical texts and also within older writings due to missing standardization, rules and models cannot be based on word forms (see the discussion in Chapter 3). Instead, the methods operate on POS tags to ensure the applicability to different text types and historical data. Besides the lack of orthography, older writings also do not follow modern punctuation rules. Punctuation marks help to determine clause boundaries and, thus, sentence brackets in modern German (Becker and Frank 2002). But it could be problematic if methods heavily rely on the presence of punctuation marks, which do not exist or are used differently in historical texts and, to some degree, in modern non-standard data. Therefore, all probabilistic models are also trained on the training set after removing punctuation from it. The following methods are included in the pilot study:

N-Gram Tagger As a baseline, a unigram tagger as implemented in the NLTK[41] is used, which simply assigns the most frequent tag (left sentence bracket, right sentence bracket or none) to each token. Furthermore, a bigram and trigram tagger with bigram and unigram backoff models are tested.[42]

[41] Version 3.2.1, https://www.nltk.org/

[42] For all n-gram taggers, see http://www.nltk.org/_modules/nltk/tag/sequential.html

	Tagger	Parser	Rule-based	Probabilistic	Expert Knowledge
N-Gram	✓	-	-	✓	-
Brill	✓	-	✓	✓	-
Bayes	✓	-	-	✓	-
Logit	✓	-	-	✓	-
FST	✓	✓	✓	-	✓
Regexp	-	✓	✓	-	✓
CFG	-	✓	✓	-	✓
Berkeley	-	✓	-	✓	-
Stanford	-	✓	-	✓	-
Benepar	-	✓	-	✓	-

Table 5.5.: Overview of the automatic methods that are evaluated in the pilot study.

Brill Tagger The Brill tagger (Brill 1992) combines the results of an initial tagger with an ordered set of learned context-dependent rules to come up with an improved annotation. The implementation is taken from the NLTK[43] using default settings and the pre-defined rule template `fntbl37`. The template consists of 37 different combinations of the current word and tag and the three preceding and following words and tags. For the purpose of this study, POS tags are treated as input words and sentence brackets as tags. The unigram tagger is chosen as initial tagger because, in combination with the Brill tagger, it achieves the best F_1-score of all n-gram taggers on the TüBa-D/Z development data.

Naive Bayes Classifier The NLTK also offers an implementation of the Naive Bayes algorithm,[44] which assigns the most likely label to each token based on an arbitrary set of features. Using the TüBa-D/Z development data, the optimal feature set is determined. When punctuation is retained, it includes the POS tags of the previous, the current, and the following token. If punctuation is removed, the POS tags of the three preceding tokens, the current POS tag, and the POS tag of the next token are used.

Logit Classifier As a second classifier, the logistic regression classifier from scikit-learn[45] is invoked via the NLTK sklearn wrapper,[46] with the options lbfgs solver, multinomial classification, and maximally 500 iterations to converge. The feature set includes the three preceding, the current, and the three following POS tags.

[43] `http://www.nltk.org/_modules/nltk/tag/brill.html`
[44] `http://www.nltk.org/_modules/nltk/classify/naivebayes.html`
[45] Version 0.21.2, `https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html`
[46] `http://www.nltk.org/_modules/nltk/classify/scikitlearn.html`

Finite-State Transducer Finite-state transducers are used in several studies described in Section 5.2, but their exact architecture and transitions remain unclear. Therefore, the FST is implemented from scratch with the python library *fysom*[47] and optimized on the development data set. The transducer is deterministic and non-recursive but allows for embedded clauses. It is robust against ungrammatical sentences or fragments, thus enabling partial analyses. The transition table and a simplified graphical representation of the FST can be found in the appendix (Table A.5, Figure A.1).

Regexp Parser The regular expression parser as implemented in the NLTK chunk package[48] successively applies a set of context-sensitive regular expressions to an input string. For the identification of sentence brackets, six simple rules are used (see Figure A.2 in the appendix). To reduce the amount of false positives, the Regexp parser is only applied to sentences that contain at least one verb.

CFG Parser The CFG parser is a chart parser from the NLTK,[49] which uses a bottom-up left-corner strategy and a basic handwritten context-free grammar optimized on the development data (see Figure A.3 in the appendix). For reasons of computational efficiency, the implementation provided by the NLTK is slightly modified to always return the first possible parse from the chart. Also, only sentences that contain at least one verb get parsed. The impact on the results should be moderate as, in this pilot study, only the sentence brackets are evaluated and not the complete structure of the parse trees.

Berkeley Parser The pilot study also includes two PCFG parsers. The Berkeley parser (Petrov et al. 2006)[50] is an unlexicalized, latent variable-based parser that shows promising results for the identification of sentence brackets in modern newspaper text (Cheung and Penn 2009). To train the parser, the training data is converted to a treebank format and intermediate nodes are introduced to match the required input format with POS tags being the leaves of the tree (for an example, see Figure 5.4).[51] To run the Java-based Berkeley parser, it is invoked in interactive mode via the command line and always returns the single best parse.[52]

[47] `https://github.com/mriehl/fysom`
[48] `http://www.nltk.org/api/nltk.chunk.html`
[49] `https://www.nltk.org/_modules/nltk/parse/chart.html`
[50] `https://github.com/slavpetrov/berkeleyparser`
[51] Training options: `java -Xmx1024m -cp BerkeleyParser-1.7.jar edu.berkeley.nlp.PCFGLA.GrammarTrainer -treebank SINGLEFILE -out grammar.gr -path treebank.txt`
[52] `java -Xmx10g -jar BerkeleyParser-1.7.jar -gr grammar.gr -maxLength 350`

TüBa-D/Z

```
(VF NE)
(LK VMFIN)
(MF ADV PIS PAV)
(VC VVINF)
COMMA
(NF
  (C KOUS)
  (MF APPR PPOSAT NN PPER PIS)
  (VC VVPP VAFIN))
PUNCT
```

→

Modified training data

```
(S
  (VF (OTH NE))
  (LK VMFIN)
  (MF (OTH ADV) (OTH PIS) (OTH PAV))
  (VC VVINF)
  (OTH COMMA)
  (NF
    (C KOUS)
    (MF
       (OTH APPR)
       (OTH PPOSAT)
       (OTH NN)
       (OTH PPER)
       (OTH PIS))
    (VC (VC VVPP) (VC VAFIN)))
  (OTH PUNCT))
```

Freudenthal wollte gestern nichts dazu sagen, ob bei ihren Prüfungen ihr etwas aufgefallen sei.

'Freudenthal did not want to say anything yesterday about whether she had noticed anything during her examinations.'

Figure 5.4.: To match the required treebank input format of probabilistic parsers without supplying word forms, the topological field annotations from the TüBa-D/Z training data (left) have to be modified (right). A sentence node `S` that also spans over unannotated tokens like punctuation or fragments is added at the top-level of the tree. Furthermore, intermediate pre-terminal nodes are inserted: For sentence brackets, the bracket label (here: `LK`, `C` or `VC`) is repeated if necessary. For the other fields, the artificial label `OTH` is introduced. This way, each pre-terminal corresponds to exactly one terminal symbol, as it would be the case with word forms and POS tags. Originally, there is no internal structure in most fields. So, as a side effect, the intermediate level also reduces the amount of rules the parsers have to learn, which can be expected to improve grammar coverage (Becker and Frank 2002).

Stanford Parser The Stanford parser (Rafferty and Manning 2008)[53] is another commonly used statistical constituency parser. For this study, it is trained as a vanilla PCFG parser on the same treebank as the Berkeley parser using default options.[54]

Benepar Nowadays, many state-of-the-art parsers are neural network-based, especially dependency parsers. There are also some neural constituency parsers, but their adaptation to unlexicalized topological field parsing is often difficult or impossible. One notable exception is the neural Berkeley parser (Kitaev and Klein 2018).[55] For this study, Benepar is trained on the same treebank data as the PCFG parsers using default options.[56] During training, the parser is also provided with the TüBa-D/Z development data in treebank format.

Evaluation and Results

The evaluation procedure in the pilot study deviates from the other evaluations in this thesis with respect to the evaluated units and metrics. Since the annotation of sentence brackets can be interpreted as a tagging task, I performed a token-wise evaluation instead of comparing spans as in subsequent evaluations. The output of the different methods is compared to the gold standard annotation, and tokens correctly recognized as (part of a) left or right bracket are counted as true positives. If a system identifies a sentence bracket where there is none in the gold standard, this counts as false positive. If a system misses a bracket or labels it with the wrong type, e.g., `LK` instead of `RK`, this is considered a false negative. All other tokens are counted as true negatives.

While the TüBa-D/Z training data distinguishes between verbal and non-verbal left brackets, this is not the case for all other corpora. During evaluation, both categories are unanimously mapped to the same tag `LK`. All methods are evaluated on the original corpora from Table 5.4 and on the same data after all punctuation has been removed from it. Table 5.6 gives an overview of the results.

For the newspaper corpus, which also serves as training and development data, many methods achieve very good results. Except for the CFG parser, all automatic approaches reach overall precision values between 98.8% and 99.8% (recall: 83.5%–99.7%). The F_1-score for all methods is >90%. The best results are achieved by the neural and the conventional Berkeley parser, followed by the Logit classifier and the Brill tagger, which all reach F_1-scores >99%. The accuracy lies above 97% for all systems. On the newspaper data, all systems (except CFG) perform slightly

[53] https://nlp.stanford.edu/software/lex-parser.shtml

[54] Training options: `java -mx1600m -cp stanford-parser.jar edu.stanford.nlp.parser.lexparser.LexicalizedParser -PCFG -vMarkov 1 -uwm 0 -headFinder edu.stanford.nlp.trees.LeftHeadFinder -saveToSerializedFile grammar.ser.gz -train treebank.txt`

[55] https://github.com/nikitakit/self-attentive-parser

[56] The training took 99 epochs to complete for the model with punctuation and 121 epochs for the model without. Training options: `python3 src/main.py train --model-path-base ./models/topf --train-path treebank.txt --dev-path treebank_dev.txt --use-words --predict-tags --batch-size 100`

Method	TüBa-D/Z				Modern				Historical			
	Prec	Rec	F_1	Acc	Prec	Rec	F_1	Acc	Prec	Rec	F_1	Acc
With punctuation												
Unigram	98.96	83.49	90.57	97.23	98.33	84.53	90.91	96.75	90.03	69.23	78.27	92.55
Bigram	99.25	88.34	93.48	98.04	98.55	88.28	93.13	97.50	90.90	76.12	82.86	93.90
Trigram	99.40	89.03	93.93	98.17	99.02	89.64	94.10	97.84	90.94	76.74	83.24	94.01
Brill	99.71	98.52	99.11	99.72	99.51	97.00	98.24	99.33	90.41	88.39	89.39	95.93
Bayes	99.66	97.26	98.45	99.51	99.37	96.52	97.93	99.21	**92.41**	87.70	90.00	96.22
Logit	99.65	98.81	99.23	99.76	99.38	97.75	98.56	99.45	90.53	88.55	89.53	95.99
FST	99.12	95.40	97.22	99.13	98.59	95.36	96.95	98.85	91.78	78.99	84.91	94.56
Regexp	99.66	96.09	97.84	99.33	99.44	96.11	97.75	99.15	91.80	81.47	86.33	95.00
CFG	95.72	85.69	90.43	97.12	94.74	86.03	90.18	96.40	78.59	60.64	68.46	89.19
Berkeley	99.70	99.23	99.46	99.83	**99.73**	**99.18**	**99.45**	**99.79**	91.37	**89.64**	**90.50**	**96.35**
Stanford	98.82	95.79	97.29	99.15	99.02	96.11	97.54	99.07	90.90	77.38	83.60	94.12
Benepar	**99.76**	**99.74**	**99.75**	**99.92**	99.45	98.64	99.04	99.63	89.77	88.18	88.97	95.76
Without punctuation												
Unigram	98.96	83.64	90.66	96.80	98.33	84.70	91.01	96.15	90.03	69.43	78.40	91.44
Bigram	99.36	87.67	93.15	97.61	98.70	88.18	93.15	97.02	90.74	74.12	81.59	92.52
Trigram	99.37	88.47	93.60	97.76	98.79	89.07	93.68	97.24	90.75	75.09	82.18	92.72
Brill	99.65	94.56	97.04	98.93	98.92	93.65	96.21	98.30	90.13	83.93	86.92	94.35
Bayes	99.05	95.77	97.38	99.04	98.26	96.38	97.31	98.78	90.75	83.33	86.88	94.37
Logit	99.57	97.40	98.48	99.44	99.17	97.54	98.35	99.25	90.03	85.78	87.85	94.70
FST	99.12	94.48	96.74	98.82	98.56	93.51	95.97	98.19	91.69	76.54	83.43	93.20
Regexp	99.66	93.67	96.58	98.77	99.50	95.15	97.28	98.78	**91.72**	80.75	85.89	94.06
CFG	96.44	88.13	92.10	97.20	95.85	88.46	92.01	96.47	79.73	62.30	69.95	88.06
Berkeley	99.78	99.14	99.46	99.80	**99.59**	**98.77**	**99.18**	**99.62**	91.49	**87.70**	**89.55**	**95.42**
Stanford	99.14	95.08	97.06	98.93	99.02	96.17	97.57	98.90	90.80	83.24	86.86	94.37
Benepar	**99.79**	**99.64**	**99.71**	**99.89**	99.38	98.63	99.01	99.54	89.98	86.71	88.31	94.87

Table 5.6.: Overall evaluation results for sentence bracket recognition with the different methods (with and without punctuation) in the pilot study. The historical data includes the DTA_{Pilot} and HIPKON sets from Table 5.4. The numbers for precision, recall, F_1-score, and accuracy are given in percent, and the highest values are marked in bold.

better when punctuation marks are available, which confirms the observation of Becker and Frank (2002) that punctuation marks help in identifying topological fields.

On the Modern data set, most systems perform slightly worse than on newspaper text. The precision values lie between 98.3% and 99.7% (except CFG). The recall values range between 84.5% and 99.2%. Again, all systems achieve F_1-scores >90% and accuracies >96%, with the Berkeley parser and the neural Berkeley parser showing the best results. Although non-standard data may not always contain correct punctuation, the presence of punctuation marks still improves the results of most systems.

For the historical data, results are considerably worse. Especially the recall values are much lower with only 60% to 90%, while the precision values lie mostly around 90% to 92%. The Berkeley parser achieves the highest F_1-score with 90.5%, followed by the two classifiers and the Brill tagger. Interestingly, the performance differs significantly between the two historical corpora. While all methods reach F_1-scores between 94% and almost 98% on the HIPKON corpus, the results for the DTA_{Pilot} texts range only between 67% and 90%. Punctuation seems to be helpful for the DTA_{Pilot} data, whereas several systems achieve better results if punctuation is removed from the HIPKON corpus beforehand, supporting the hypothesis that deviations from modern punctuation rules can be detrimental to the recognition of sentence brackets.

To understand the large differences between the two historical data sets, I exemplarily inspected the annotations of the Berkeley parser for the DTA_{Pilot} sample. The manual analysis reveals that many errors for both bracket types in the DTA_{Pilot} seem to be caused by POS tagging errors, which are quite prevalent in the automatic annotation provided by the German Text Archive (cf. Chapter 3.2). Unfortunately, several potential bracket elements contribute to these high POS error rates and lead to false positive and negative sentence brackets. For example, false negative `LKs` are caused by relative pronouns that are incorrectly tagged as articles as in example (17), or conjunctions that are mistaken for adverbs or prepositions as in example (18). On the other hand, false positive `LKs` can be triggered, e.g., by demonstratives that are incorrectly tagged as relative pronouns or nouns, and adjectives that are tagged as finite verbs as in example (19).

(17) denen / **die/*ART** ſolches thun
'those who do such things'

(18) **bis/*APPR** dieſer [...] eine Ausnahme verdienet
'until this one deserves an exemption'

(19) dieſer **verhinderte/*VVFIN** Umlauff des Blutes
'this prevented circulation of the blood'

False negative `RKs` can result, e.g., from verb particles that are annotated as adverbs/prepositions as well as adjectives or nouns that are in fact verbs as in example (20), whereas false positive `RKs` get triggered by supposed verbs and verb particles that are really nouns or adjectives.

(20) daß euch jemand **lehre/*NN**
'that someone teaches you'

The confusion of LK and RK can be caused by confusions between verb tags, e.g., finite verbs and infinitives. Also, since the parsers (and several other methods) take into account the whole sentence, a missing left sentence bracket in many cases triggers the confusion of the next right bracket with a left bracket.

The strong influence of incorrect POS tags on the system results makes it difficult to draw reliable conclusions from the evaluation on this data set. Of course, it is a realistic scenario that annotations are imperfect because many (historical) corpora are not manually annotated, and even gold-standard annotations may contain errors (cf., e.g., Manning (2011) for POS tagging). However, evaluating on such incorrect data makes it difficult to judge whether an error is actually caused by the evaluated system or by errors in the data. For a conclusive impression of system performance with respect to a specific task, clean and correct gold data is indispensable. As a consequence of the pilot study, I decided to exchange the DTA_{Pilot} data set for a smaller but manually corrected sample that is used in all subsequent experiments (Table 5.2; see also Chapter 3.2).

Discussion

Despite the problems with POS errors in one of the historical data sets, the pilot study gives a good impression of the suitability of different methods for the automatic recognition of sentence brackets. The results show that several of the tested methods can reliably identify sentence brackets in German text, including taggers (Brill), classifiers (Bayes, Logit), and parsers (Berkeley, Benepar). The Regexp parser as one of the systems that do not require training data also achieves surprisingly good results, despite its simplicity.

Since the end goal of this chapter is a complete topological field analysis, it seems most promising to use one of the probabilistic parsers. The neural and conventional Berkeley parsers achieve the best results with almost perfect F_1-scores for modern data and also reach high accuracy values for historical German. The conventional Berkeley parser appears to be slightly better at adapting to non-standard and historical language and is much faster to train, which qualifies it as the preferred method for my follow-up studies. Future work has to show whether the results obtained in this thesis could be improved by using Benepar (or another neural parser), e.g., in combination with pre-trained word embeddings and normalized/modernized historical word forms or historical training data in general.

5.4.2. Evaluation and Results

To resolve the methodological problems, the annotation experiment from the pilot study is repeated with the Berkeley parser and the data sets from Section 5.3. Again, two models are tested: the Punct model from the pilot study, trained on the TüBa-D/Z training set with punctuation, and the parser model News1, trained on TüBa-D/Z constituency trees, i.e., it includes a constituency and topological field analysis (cf. Chapter 6.2).

Instead of the token-wise evaluation from the pilot study, a span-based evaluation is conducted. Each bracket span that was identified by the parser is compared to each target span from the gold

	Punct			News1		
Corpus	Prec	Rec	F_1	Prec	Rec	F_1
TüBa-D/Z	99.06	99.36	99.21	**99.58**	**99.39**	**99.48**
Spoken	97.60	99.29	98.44	**98.59**	**99.40**	**98.99**
Modern	98.56	**99.39**	**98.97**	**98.81**	98.81	98.81
HIPKON	**96.88**	**94.82**	**95.84**	90.91	83.39	86.98
DTA	92.25	**92.46**	**92.36**	**92.35**	91.18	91.77

Table 5.7.: Overall precision, recall, and F_1-scores (in percent) for sentence bracket recognition according to *FairEva*l for the different models on each data set. The highest scores for each corpus are highlighted in bold. Traditional evaluation results can be found in Table A.6 in the appendix.

data sets, as described in Chapter 4. Only target sentences that contain at least one sentence bracket are evaluated. Table 5.7 shows fair precision, recall, and F_1-scores.

Compared to the token-based evaluation in the pilot study, the span-based evaluation leads to slightly lower overall scores, which means that errors are especially caused by multi-token brackets. For modern German, fair F_1-scores lie above 98.9%. For the historical corpora, there is again a difference of about 3.5 percentage points between HIPKON and the clean DTA sample, but both data sets are analyzed with fair F_1-scores above 92%. F_1-scores decrease with the age of the text, ranging from 77%–100% for the DTA and 67%–100% for HIPKON. While the constituency model `News1` reaches higher scores for the TüBa-D/Z and Spoken test data, the pure topological field model with punctuation performs better on the other data sets, especially on older historical data.

For the modern newspaper data and HIPKON, precision values are higher than recall, reflecting low proportions of false positives (cf. Figure 5.5). For the other data sets, the opposite is observed, i.e., recall values are higher than precision due to more false positives. Except for the spoken data, about half of the errors are boundary errors. In most of these cases, the system annotates a shorter bracket. A manual inspection of the annotations shows that this concerns, e.g., subordinating conjunctions consisting of more than one word like *so dass* ('so that') or relative and interrogative phrases in the non-verbal left bracket of which the parser sometimes only recognizes the first word(s). Another common cause of boundary errors are coordinated verbs in the right sentence bracket.

Table 5.8 shows that the left bracket is recognized more reliably than the right bracket in historical data. The opposite is true for the modern spoken and non-newspaper data, where left brackets are prone to be false positives (cf. Figure 5.6). Confusions between the two bracket types are rare for written modern German. In the DTA, confusions are, for example, caused by an unusual order of elements in the right sentence bracket, as in example (21), where the initial finite verb is incorrectly recognized as a left bracket.

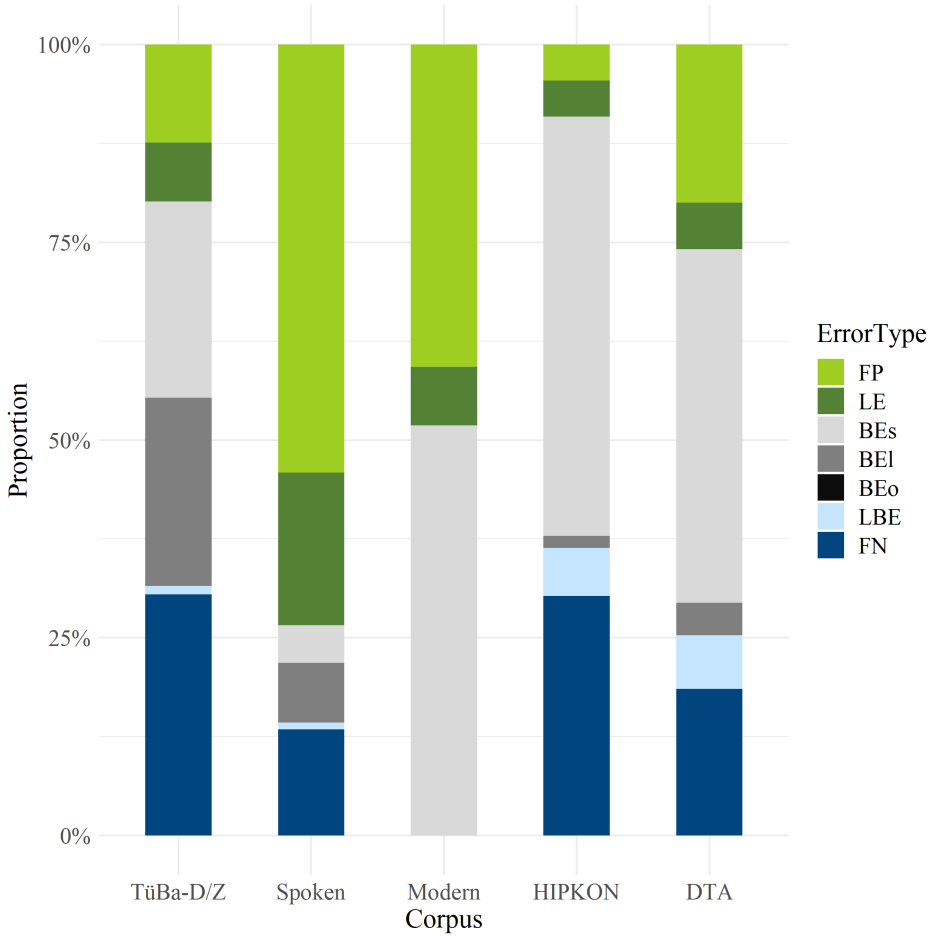


Figure 5.5.: Proportion of the different error types for sentence bracket identification: false positives (FP), labeling errors (LE), shorter, longer, and overlapping boundary errors (BE_s, BE_l, BE_o), labeling-boundary errors (LBE), and false negatives (FN). Numbers are shown for the best model on each data set.

(21) welche [...] dahero leichtlich **kȯnnen** verſtopffet werden
'which therefore can easily become clogged'

In older texts, especially in the HIPKON corpus, confusions between the bracket types are mostly triggered by false negative left brackets, which in turn lead to RKs labeled as LKs. Often, the missing LKs are relative adverbs or particles like in example (22), which no longer exist in modern German but are frequently used in older historical writings.

Corpus	LK	RK
TüBa-D/Z	99.69	99.17
Spoken	98.82	99.38
Modern	98.72	99.39
HIPKON	97.72	93.97
DTA	94.25	90.63

Table 5.8.: Overall F_1-scores for each label (in percent) according to *FairEval* for the best performing model on each data set.

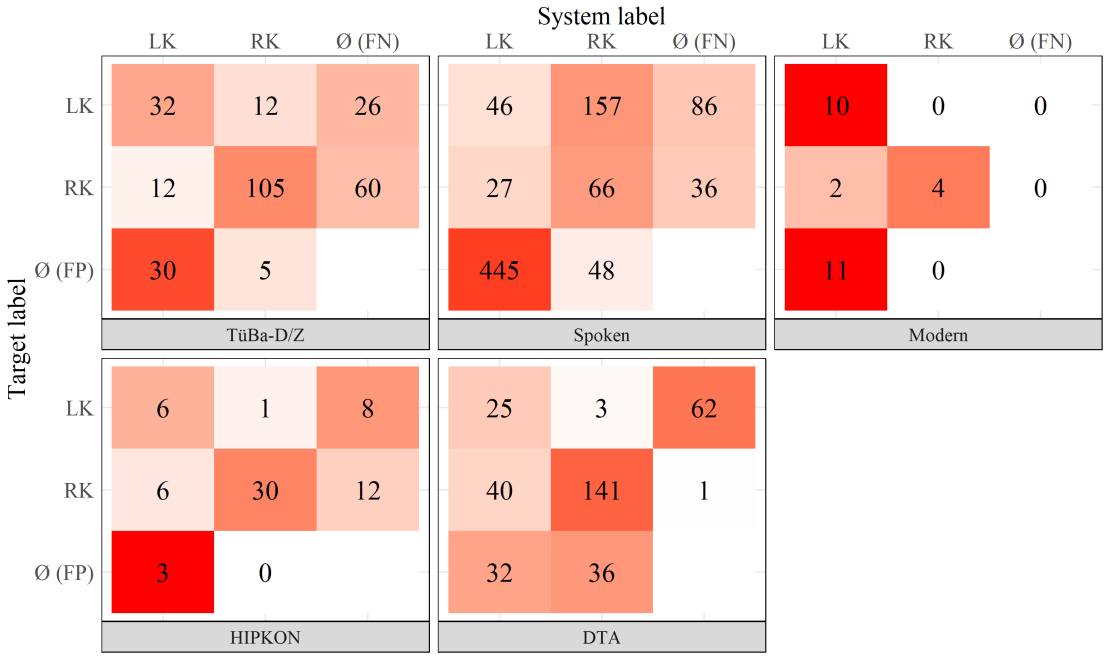


Figure 5.6.: Confusion matrix for the identification of sentence brackets. Only errors are displayed, so the diagonal displays boundary errors.

(22) nach mittē tage **do** er hat geſclâfen
'after the middle of the day where he had slept'

These relativizers were tagged as such with the original custom tagset of the HIPKON corpus, but the information about their relative function was lost during mapping to the STTS (Schiller et al. 1999). As the STTS is intended for the annotation of modern German, it does not provide a special category for these cases, making them undetectable for the parser. In the pilot study, I experimented with explicitly marking relativizers as such, which improved the Berkeley parser's F_1-score on the HIPKON data by 0.35 percentage points.

	Punct			News1		
Corpus	Prec	Rec	F_1	Prec	Rec	F_1
TüBa-D/Z	95.25	96.67	95.96	**97.31**	**97.44**	**97.37**
Spoken	88.72	93.58	91.08	**91.30**	**94.14**	**92.70**
Modern	96.10	**94.38**	**95.23**	**96.78**	93.63	95.18
HIPKON	**93.99**	**92.69**	**93.34**	86.42	83.36	84.86
DTA	85.22	**85.61**	**85.42**	**85.61**	82.80	84.18

Table 5.9.: Overall precision, recall, and F_1-scores (in percent) for topological field parsing according to *FairEval* for the different models on each data set. The highest scores for each corpus are highlighted in bold. Traditional evaluation results can be found in Table A.7 in the appendix.

5.5. Topological Field Parsing

The previous section has demonstrated that sentence brackets can be reliably identified in modern and historical data without any historical training data available. The results are in accordance with the reported literature (Section 5.2), which always finds the highest scores for the bracket elements. The focus of this section lies on the identification of the other, more complex topological fields. Again, the Berkeley parser with the `Punct` and `News1` models is used and applied to the data sets from Table 5.3.

For evaluation, the parser output is compared to the target annotation. As explained in Chapter 4, only the token span covered by a field is considered, independently of possible intermediate embedded fields. Punctuation is ignored during evaluation, and only sentences for which there is a gold analysis are included. Table 5.9 shows fair precision, recall, and F_1-scores for each data set.

As expected, the parser achieves the best results on the TüBa-D/Z test data, i.e., the type of data it was trained on, with an F_1-score of 97.4% with the constituency model and about 96% with the pure topological field model. This is comparable to the results of Cheung and Penn (2009), who report a (traditional) F_1-score of 95.2% for a (much smaller) part of the same corpus. For the two other modern data sets, the parser reaches an overall F_1-score of 95.2% (written) and 92.7% (spoken). For the historical data, accuracies differ between data sets. While the results for the HIPKON corpus are comparable to the modern spoken data, the overall F_1-score for the DTA is much lower with about 85.4%.

Like in previous studies, the sentence brackets are annotated with the highest accuracy in all data sets, followed by the pre- and middle fields, while the results for post-fields are worse for all data sets (cf. Table 5.10). Left dislocations are recognized even more rarely by the parser. The results for the coordination field vary between data sets, as well as the proportion of sentences the parser can analyze without errors (31%–79%, Ortmann 2020). In general, correctly analyzed sentences are on average shorter and contain fewer fields. For some fields, it makes a difference if they are embedded in other fields or contain embedded fields themselves. For example, post-fields and left dislocations are recognized less often and less accurately if they do not contain other fields. This

Corpus	KOORD	LV	VF	LK	MF	RK	NF
TüBa-D/Z	94.05	75.09	97.75	99.73	96.66	99.15	87.45
Spoken	64.71	25.96	91.71	98.98	92.96	99.42	64.80
Modern	72.54	13.33	96.65	98.84	96.70	99.39	81.84
HIPKON	96.88	0.00	94.89	97.97	96.25	93.62	85.21
DTA	71.98	39.53	88.45	94.79	82.14	90.67	63.89

Table 5.10.: Overall F_1-scores for each label (in percent) according to *FairEval* for the best performing model on each data set.

can be explained by the characteristics of the training data: Post-fields and left dislocations are rare in newspaper text and mostly contain 'heavy', complex elements, i.e., longer clauses.

The confusion matrices (Figure 5.7) reveal that errors for all data sets mainly occur on the diagonal (boundary errors) and in the right-most column and/or bottom row (false negatives and positives). False positives are most frequent in the modern spoken data, while the modern non-newspaper data exhibits more false negatives (cf. Figure 5.8). Labeling-boundary errors are most prevalent in the historical corpora.

Besides those general observations, every corpus poses different challenges to the parser. To better understand the differences between data sets and the causes of errors, in the following, the results for the different corpora are analyzed in more detail.

TüBa-D/Z Except for post-fields and left dislocations, all fields in the Tüba-D/Z test data are recognized with F_1-scores between 94% and 99.7%. The sentence brackets are identified with the highest accuracy, followed by pre- and middle fields and coordinations. For all fields (except KOORD and LV), more than 24% of the errors are boundary errors. This value is highest for the middle field, where 63% of the fields only have incorrect boundaries. That can, for example, be the case if the right sentence bracket is empty and the parser regards the middle and post-field as a single field.

In Ortmann (2020), I found that four out of five sentences from this data set are analyzed without any error. On average, those sentences are ten words shorter than sentences containing errors. A qualitative error analysis reveals that errors mostly occur with elliptical constructions, fragments, parenthetical phrases, and sentence structures that are uncommon in standard German and, therefore, rare in the training data. This observation is in accordance with Cheung and Penn (2009), who also identify parentheticals as the main error cause in their study. Additional error sources are quotes and reported (direct) speech, as well as left dislocations and post-fields without internal structure.

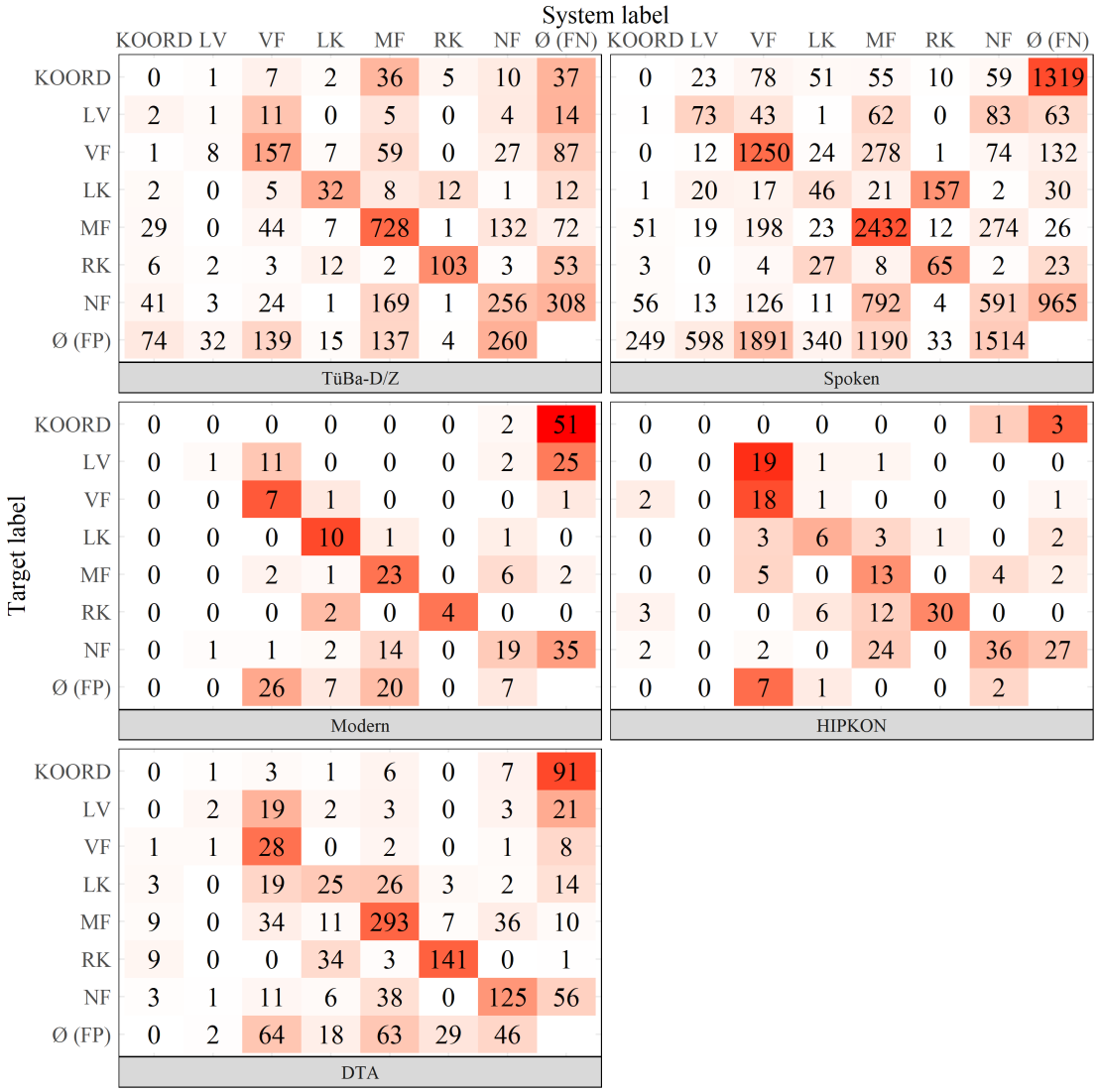


Figure 5.7.: Confusion matrix for the identification of topological fields. Only errors are displayed, so the diagonal displays boundary errors.

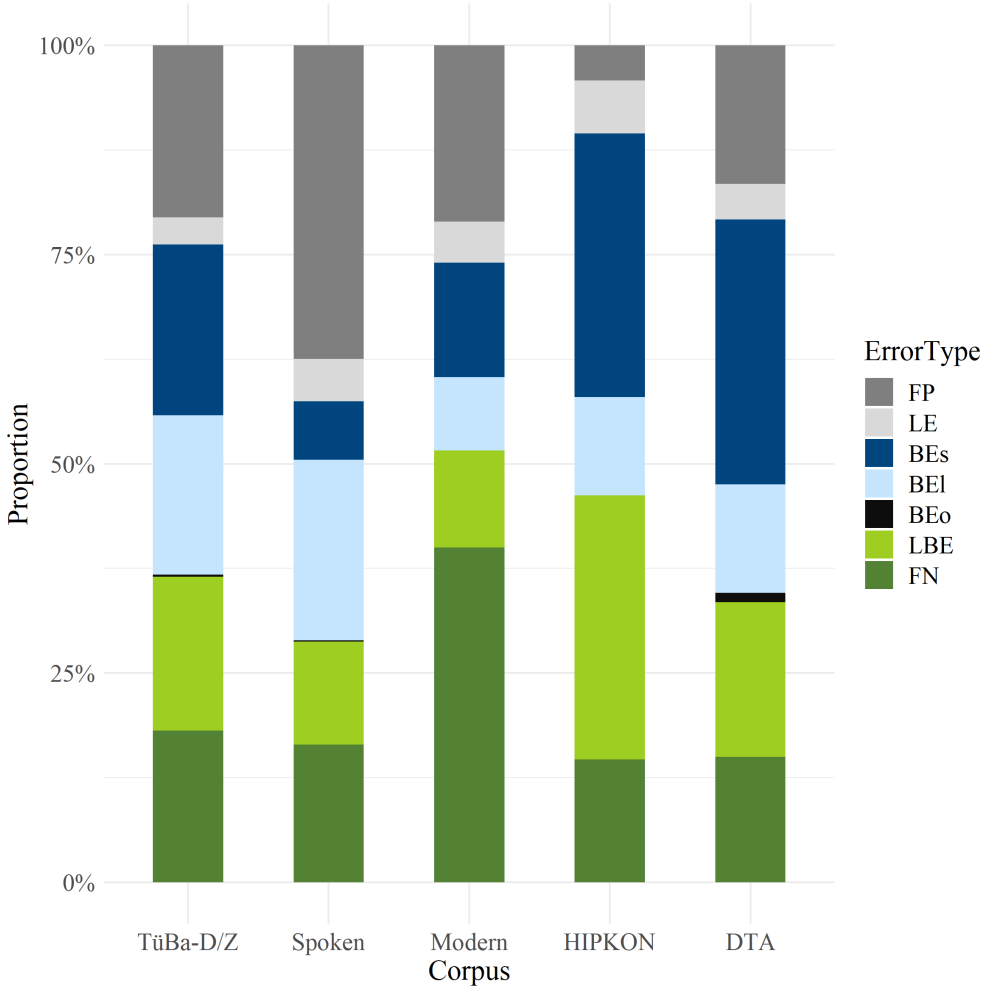


Figure 5.8.: Proportion of the different error types for topological field analysis: false positives (FP), labeling errors (LE), shorter, longer, and overlapping boundary errors (BE_s, BE_l, BE_o), labeling-boundary errors (LBE), and false negatives (FN). Numbers are shown for the best model on each data set.

Spoken While the sentence brackets and pre- and middle fields are recognized with F_1-scores >91%, only about two thirds of the coordinations and post-fields are identified correctly in the spoken data. Left dislocations are recognized with an F_1-score of only 26%. Again, many system spans overlap with the gold standard annotation, especially in the case of middle fields (58%), which often erroneously stretch across post-fields.

In Ortmann (2020), I found that almost two thirds of all sentences in the spoken data set are analyzed without errors. On average, these sentences contain nine fewer words than incorrect sentences. Errors mostly result from the divergence between spoken and written language structures, for instance, incomplete utterances, repeated words, or unrelated clauses and fragments in a single sentence. Still, it can be stated that, despite the differences between written training and spoken test data, the majority of the fields are recognized with fairly high accuracy and, if similar data should be processed automatically, using part of the spoken data as additional training resource could further improve the results for this text type.

Modern For the modern written data set, the evaluation shows that texts from different registers can be analyzed with comparable accuracy as newspaper data. The parser performs best on the Wikipedia articles (F_1: 96.7%), while F_1-scores range between 94% and 96% for the other registers. Although the data shows a slightly different distribution of fields with more left dislocations, post-fields, and coordinations, the parser still recognizes most fields with high F_1-scores. Except for `KOORD` and `LV` fields, between 20% and 67% of the errors are boundary errors.

In Ortmann (2020), I found that 58% of the sentences are analyzed completely correctly. For many sentences, missing coordination fields are the only error. Since coordinating conjunctions are not always annotated in the training data, the parser often does not recognize them in the test data, leading to low recall for the `KOORD` field (96% `FNs`). In my study (Ortmann 2020), I showed that the recall of the `KOORD` field could be increased from 56% to 97% with simple rules while keeping the precision at 100%, thus improving the F_1-score of this field to 98%. Other common causes of errors are direct and reported speech, especially in sermons and fiction texts, and the higher proportion of left dislocations and post-fields in informal, spoken-like language.

HIPKON The overall scores for the first historical corpus are comparable to those of modern spoken data (although with quite different error distributions, cf. Figure 5.8). For most fields, the F_1-score is >93%. Despite the higher proportion of post-fields resulting from the corpus design, post-fields are analyzed with a higher F_1-score in this historical text sample than in most of the other corpora. For left dislocations, the opposite is true: Although they are more frequent in the data set, no `LV` field is recognized in the HIPKON sample. Either the corresponding tokens are not analyzed at all, or they are analyzed as part of the pre-field, which is also reflected in the high percentage of pre-fields with incorrect boundaries (62%). In general, boundary errors account for 37%–62% of the incorrect fields (without `LV` and `KOORD`).

In Ortmann (2020), I found that about two thirds of the sentences from this data set are analyzed without errors by the parser. There is no clear tendency concerning to the age of the text, but recall

values tend to be slightly lower than precision. Common error causes for this data set include empty middle fields like in example (23), which are relatively frequent in the HIPKON corpus due to its specific focus on the post-field.

(23) vñ [$_{\texttt{LK}}$ wǒlte] [$_{\texttt{RK}}$ gan] zů ſínem vatt' vnd ſprechē.
'And wanted to go to his father and speak.'

Adding historical training data or implementing simple rules, in these cases, could prevent the wrong identification of a middle field if, for example, it is preceded by a right bracket or starting with verbal elements. Additional rules could also improve the identification of post-fields, which are often not recognized by the parser (29% `FNs`). By simply labeling unanalyzed tokens following a post-field or right bracket as post-field, in my study (Ortmann 2020), I was able to increase the recall for this field by six percentage points to over 90%.

Another common cause for errors in this historical data set are left brackets like relative adverbs and particles that no longer exist in modern German (cf. Section 5.4.2 on sentence bracket identification). The information about their relative function was lost during conversion to the modern STTS tagset, preventing the parser from identifying them. Since one missing bracket can change the complete analysis of a sentence, the explicit marking of these tokens as left brackets results in improvements of all fields from pre- to post-field. For a reliable analysis of older historical data, available information about the relative function of tokens should somehow be transferred to the modern tagset, e.g., by adding a special tag and corresponding training data or by (mis-)using an existing tag for relativizers.

Overall, the evaluation of the HIPKON data shows that, by using the POS tags as input, it is generally possible to transfer a model from modern to historical data, even though some special adjustments and/or historical training data would be beneficial to improve the reliability of the automatic analysis.

DTA The results for the second historical corpus are substantially worse than for the other data sets. Only the sentence brackets are identified with F_1-scores >90%, while the other fields range only between 39% and 88%. Like for the other corpora, the results are worst for left dislocations: Only a quarter of them is recognized correctly, while the rest is mostly skipped by the parser, especially if they do not contain embedded fields. Coordination fields are also often not recognized, but adding the same simple rules as for the modern written data could increase the recall for the `KOORD` field from 56.8% to 90.1% in Ortmann (2020), improving the F_1-score of this field by 20 percentage points.

Again, between 22% and 65% of the errors result from incorrect boundaries, especially in the case of middle fields, right sentence brackets, and post-fields. But only 30% of all sentences in the study (Ortmann 2020) were analyzed without errors. On average, those sentences are 26.5 words shorter and contain 6 fewer fields than sentences with one or more errors. That already indicates that the sentences in the DTA are very long and complex. The average sentence length in the sample is 39 words, compared to 19 words in the modern newspaper texts (spoken: 10, written: 14,

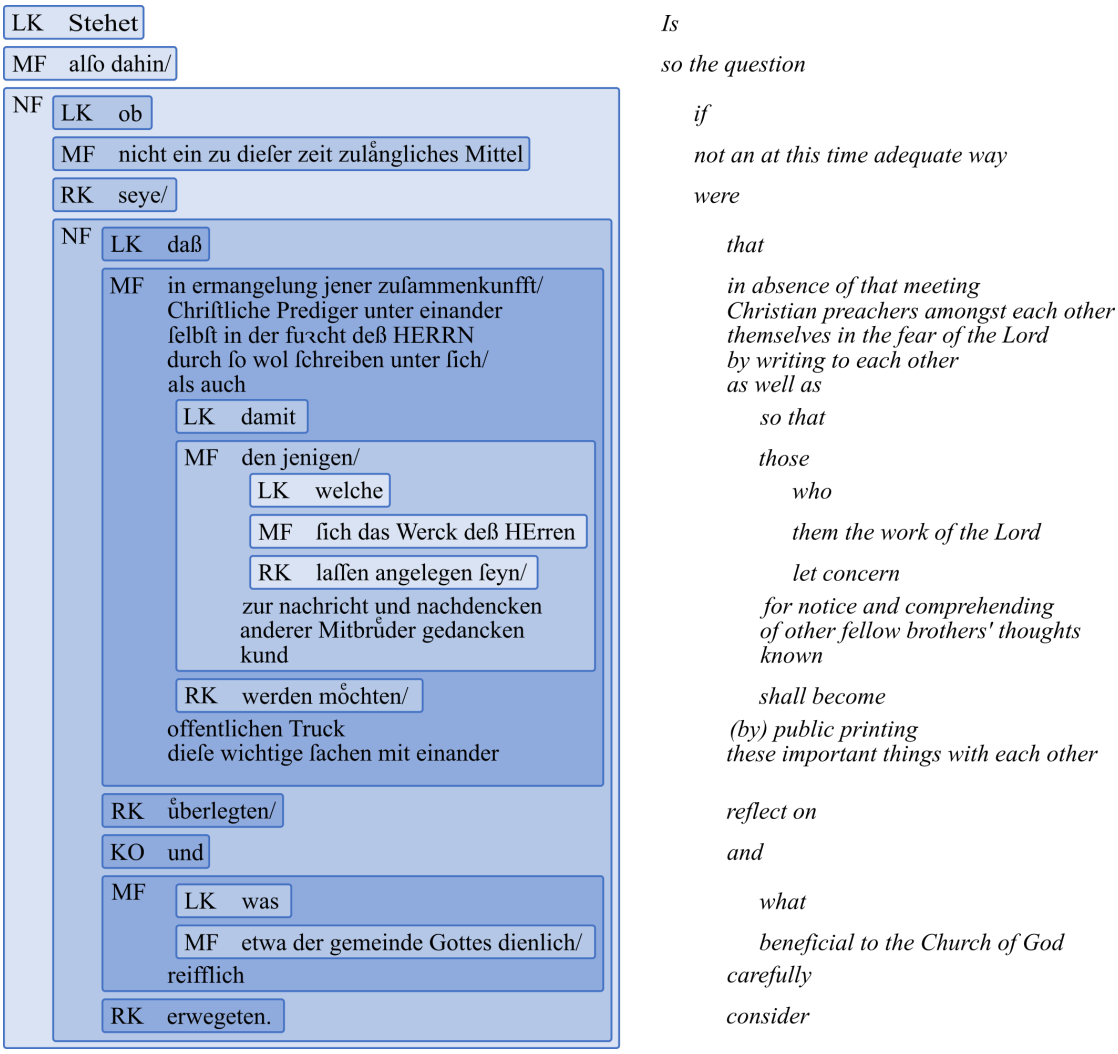
HIPKON: 12), with a maximum embedding depth of 10 fields, i.e., one field containing nine other nested topological fields, compared to a maximum depth of 6 fields in the newspaper data (spoken: 5, written: 4, HIPKON: 3). Long and complex left dislocations and deeply embedded post-fields are very common in the data set, as well as embedded structures within the middle field, which are infrequent in modern German. Furthermore, the data contains many parenthetical constructions that are hard to process and understand, even for human annotators. An example sentence from the data set can be found in Figure 5.9.

The often extreme sentence length and complexity and the deep embedding of fields is a typical characteristic of the Early New High German data and not covered by the modern training data, which explains the high number of errors. While the parser is mostly able to recognize local, clause-internal structures, e.g., left and right brackets surrounding a middle field, it often fails to identify larger structures, especially in complex constructions, e.g., with several embedded post-fields. The different historical use of punctuation further exacerbates the problems, for example, with reported speech and parenthetical constructions. The same can be said about the fact that writers during this time period commonly left out right sentence brackets, which makes embedded clauses even harder to recognize and analyze correctly, for example in (24). Also, similar to the HIPKON corpus, the DTA sample contains adverbial left brackets that the parser cannot recognize, leading not only to missing left brackets but also incorrect surrounding fields.

(24) Ob dieſes wol eine lôbliche Sache / wodurch vielmal ſolche Seuche abzuhalten [...]: So bezeuget doch die tägliche Erfahrung / daß [...]
'Although this (is) a laudable thing, whereby often such an epidemic can be prevented [...], daily experience shows that [...]'

Since these error sources become less frequent over time, there is a clear relationship between the age of the text and how well the parser performs: F_1-scores decrease with increasing age of the text (cf. Figure 5.10). This observation holds for all genres in the sample except the youngest gardening text and the funeral sermons, which are only available for the earliest time periods. The highest F_1-scores are reached for the most recent newspaper and chemistry texts and the lowest for the oldest texts from the genres of law and language science.

It has to be kept in mind, though, that the texts in this sample are already corrected for sentence boundaries and POS tags. Using the original annotations, the results would be worse, especially for older texts where POS error rates are high. In my study (Ortmann 2020), I supplied the parser with the original POS tags (and gold sentence boundaries for evaluation purposes). As a result, the overall F_1-score decreased by almost 10 percentage points to 75.6%. For many older texts, there was an even larger reduction in F_1-score of 20 or more percentage points. Using the original sentence segmentation can be expected to further reduce the accuracy. While missing sentence boundaries do not necessarily cause problems, the low precision values (avg: 83%) would lead to many incomplete fields crossing sentence boundaries. This highlights the importance of reliable basic annotations like sentence boundaries and POS tags.



'So it is the question whether it would not be adequate at this time that - in absence of that meeting - Christian preachers themselves faithfully reflect on these important things and about what would be beneficial to the Church of God by writing to each other as well as through public printing, so that these thoughts become known to those who are concerned.'

Figure 5.9.: Topological field analysis of a sentence from the theological text *Pia Desideria* from the DTA data set (Philipp Jacob Spener, 1676; BBAW 2021)

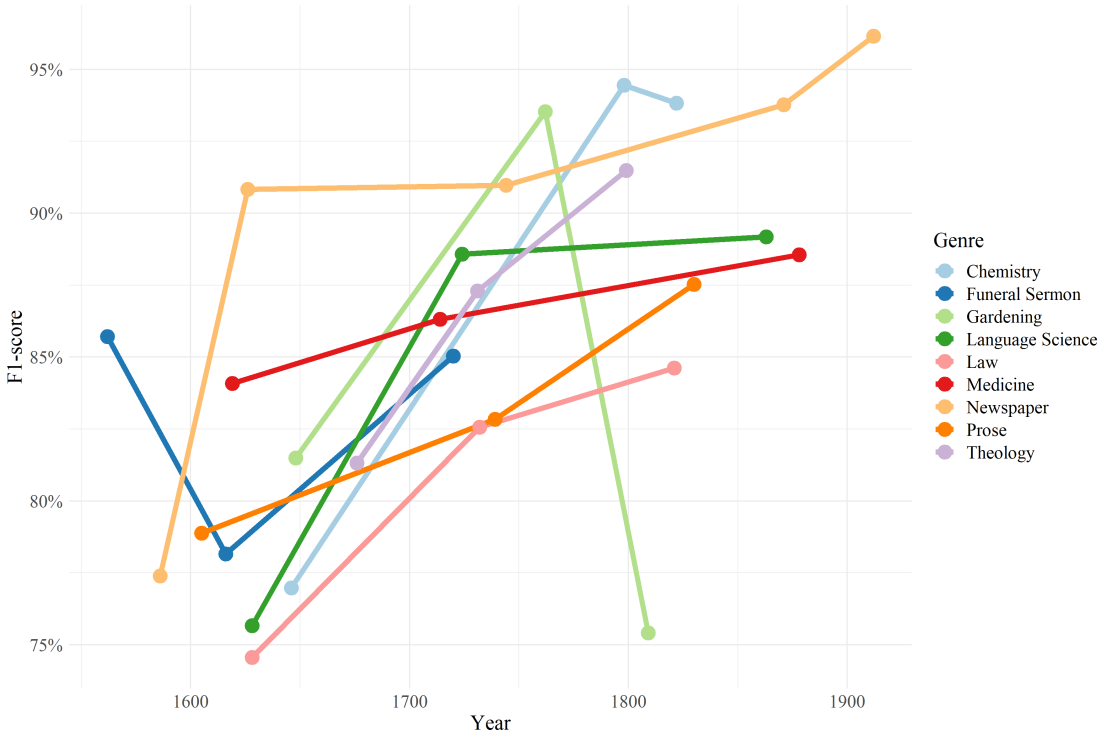


Figure 5.10.: *FairEval* F_1-scores for the different genres from the DTA sample over time.

Overall, the evaluation of this data set shows that texts from the Early New High German period that were written by skilled writers or scientists, like it is the case for the DTA sample, can only unsatisfactorily be analyzed with models purely trained on modern German. While additional rules could certainly improve the automatic field identification to a certain extent, it is unlikely that a parser will be able to reliably analyze such complex sentences without sufficient similar training data.

5.6. Discussion

In this chapter, I have explored the automatic identification of topological fields in modern and historical German without any historical training data. Based on my pilot study, I selected the Berkeley parser as the preferred tool and trained models on modified topological field trees with POS tags as input tokens. The evaluation has shown that these probabilistic, POS-based models achieve very good results for the annotation of sentence brackets in modern and historical language, with overall F_1-scores between 92% and 99.5%.

Identifying the other topological fields is more difficult, with overall F_1-scores of 92%–97% for modern German and 85%–93% for the historical data sets. For the HIPKON corpus, which covers

sermons from the 12th to 18th century, the accuracy is comparable to modern spoken German, whereas the high complexity of Early New High German sentences in the DTA poses more problems to the parser.

Independently of the data set, the different fields are analyzed with different levels of accuracy. While sentence brackets, pre-fields, and middle fields are recognized quite reliably with F_1-scores mostly >90%, the scores for post-fields and left dislocations are significantly lower. These results are in accordance with the literature (cf. Section 5.2) but especially problematic in the context of this thesis, since post-fields are highly relevant to the identification of extraposition.

However, the specific effects of an imperfect topological field analysis on the end result cannot be deduced from the resulting scores alone because errors from the two annotation steps (Chapters 5 and 6) can reinforce or mitigate each other. Also, post-fields with incorrect boundaries (14%–44% in the different data sets) do not necessarily prevent the recognition of extraposed elements if the spans still overlap. And different types of extraposed elements are also affected to different degrees (see Chapter 7) because not all post-fields are affected by errors in the same way, either (e.g., regarding complexity and embedded fields). So, despite the uncovered weaknesses, the developed methods are a practical basis for the subsequent annotation experiments in this thesis.

CHAPTER 6

Identification of Extraposition Candidates

The second prerequisite for the automatic analysis of extraposition is to identify constituents that could be extraposed. I will refer to these elements as candidates for extraposition. In this chapter, I do not (yet) distinguish where in the sentence the candidates are located (pre-field, middle field, or post-field). Instead, the studies focus on recognizing all potential candidates that *could* be moved to the post-field. According to Zifonun et al. (1997), the following constituents can be placed in the post-field (ordered by likelihood of extraposition; p. 1651):

(25) sentences > adjunct phrases > prepositional phrases > nominal phrases > adjective/adverb phrases

Sentences include content clauses, complement clauses, and relative clauses. For some of these clause types, the post-field is the default position, e.g., argument clauses as in example (26a). For these clauses, the pre-field is the only unmarked alternative to a post-field placement, cf. example (26b). An actual variation of positions between middle field and post-field is only observed for attributive clauses like relative clauses, which can be placed adjacent to their antecedent in the pre-, middle, or post-field or be separated from it through extraposition. Antecedents are treated in Chapter 7.1, which deals with determining the base position.

(26) a. Ihr **ist** bewusst **gewesen**, [$_{NF}$ dass das nicht einfach wird].

b. [$_{VF}$ Dass das nicht einfach wird], **ist** ihr bewusst **gewesen**.
'She was aware that this will not be easy.'

Besides clauses, several phrase types can be placed in the post-field. Adjunct phrases are phrases starting with *als* or *wie* ('as/like'). According to Zifonun et al. (1997), the post-field is considered their de-facto default position if they are used as comparative elements. Although they could be placed in the middle field, as demonstrated in example (27) from the Modern data set, this rarely happens neither in modern nor in historical data (but see Sahel (2015) for opposing observations in the 17th century). In our manually annotated data set of Early New High German, over 92% of the comparative elements are unambiguously extraposed. The remaining ones are placed adjacent to their antecedent at the edge of middle field and post-field with an empty right bracket, or they

are moved to the post-field together with their antecedent. The Modern data set contains no *in situ* comparative phrases at all.

(27) a. In einer Studie fanden wir heraus, **dass** [$_{MF}$ Achtsamkeitstraining Menschen doppelt so gut] **hilft** [$_{NF}$ das Rauchen aufzugeben wie die beste Standardtherapie].

b. In einer Studie fanden wir heraus, **dass** [$_{MF}$ Achtsamkeitstraining Menschen doppelt so gut wie die beste Standardtherapie] **hilft** [$_{NF}$ das Rauchen aufzugeben].

'In one study, we found that mindfulness training helps people quit smoking twice as well as the best standard therapy.'

The remaining phrase types are extraposed more rarely. As already mentioned in Chapter 2, prepositional phrases are the most frequently extraposed phrase type in modern standard German, while other phrase types are extraposed mainly in oral language, with higher amounts of extraposition being reported for older stages of German.

In this thesis, I will only consider elements for which there is relevant variation between *in situ* placement and extraposition. That excludes comparative elements, which are almost exclusively found in the post-field, as well as content clauses, which can only be placed in the pre- or post-field. Complements of nouns and adjectives are also found either adjacent to the antecedent in the pre-field or extraposed if the antecedent is located in the middle field (Zifonun et al. 1997). Therefore, they are not considered here. What remains are the following constituents: noun phrases (NP), prepositional phrases (PP), adjective phrases (AP), adverb phrases (ADVP), and (attributive) relative clauses (RelC).

Similar to the previous chapter, where I started with the simpler task of sentence bracket recognition before proceeding to the complete topological field analysis, I also split the task of identifying candidates for extraposition into three incremental steps. Section 6.1 starts with an experiment on chunking modern and historical German. In Section 6.2, I report a study on the automatic recognition of phrases, followed by the identification of relative clauses in Section 6.3. The chapter concludes with a discussion in Section 6.4.

6.1. Chunking (of German)[57]

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 (28) shows the annotation of an English sentence following Abney's chunk definition.

[57] The content of Section 6.1 is taken from my paper Ortmann (2021a): *Chunking Historical German* and was updated with *FairEval* results.

(28) [$_{S}$ [$_{NP}$ 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. It 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 (28), this results in the annotation in example (29).

(29) [$_{NP}$ 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 (30) 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 (31).

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

(31) **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.[58]

Of the eleven original chunk types from the CoNLL-2000 shared task, four main types are considered in this chapter: noun chunks (`NC`), prepositional chunks (`PC`), adjective chunks (`AC`), and adverb chunks (`ADVC`), and, in addition, stranded noun (`sNC`) and prepositional chunks (`sPC`).

[58] Similar to the decision for a topological field scheme in Chapter 5, my selection of the chunking scheme has a very practical reason because the available training data from the TüBa-D/Z corpus is annotated according to the definition by Kübler et al. (2010).

Example (32) shows the annotation of a sentence from an 1871 newspaper taken from the DTA data set. For better readability, the relation of stranded articles to their respective noun chunks is indicated by subscripts.

(32) [$_{\text{sNC}_1}$ die] [$_{\text{sNC}_2}$ den] [$_{\text{PC}}$ an Deutschland] [$_{\text{NC}_2}$ abgetretenen Landestheilen]
the *the* *to Germany* *transferred territories*

[$_{\text{NC}_1}$ angehörenden Kriegsgefangenen] [...] werden [$_{\text{ADVC}}$ sofort] [$_{\text{PC}}$ in Freiheit]
belonging prisoners of war *will be* *immediately* *to freedom*

gesetzt;
set

'Prisoners of war belonging to the territories transferred to Germany will be released immediately.'

Allgemeine Zeitung, no. 72, 1871 (DTA; BBAW 2021)

6.1.1. 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 several 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 bi-directional 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, 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 (cf. the discussion in Chapter 2.3). In the absence of training data, some studies develop rule-based approaches, e.g., Chiarcos et al. (2018) for topological field identification in Middle High German (cf. Chapter 5). 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%. 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 the previous chapter has shown, models trained on modern newspaper text can successfully be transferred to historical German with F_1-scores $\geq$90% when POS tags are used as input – unless the historical language structures differ too much from modern German (Ortmann 2020). In this section, rule-based and statistical approaches will be tested for chunking historical (and modern) German.

6.1.2. Data

Most German data sets 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 contains over 743k instances of the six chunk types considered in this thesis.

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, I included the Tiger corpus (Brants et al. 2004) as a second German treebank in my chunking study. The Tiger-style 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 before chunks of the six different types can be extracted from the constituency trees similar to the procedure described by Kübler et al. (2010). Here, a combination of the raising and splitting approaches described by Hsu (2010) is applied to

Model	#Docs	#Sents	#Toks	#Chunks
Training				
`News1`	3,075	83,225	1,564,840	593,735
`News2`	1,863	39,976	726,811	255,077
`Hist`	28	23,470	566,288	217,269
`Mix`	1,891	63,446	1,293,099	472,346
Development				
`News1`	377	10,702	196,308	74,780
`News2`	200	4,567	81,593	28,615
`Hist`	28	2,932	72,123	27,815
`Mix`	228	7,499	153,716	56,430

Table 6.1.: Overview of the training and development data for each of the models. Only sentences containing at least one chunk of the relevant types are included. `News1` corresponds to the TüBa-D/Z and `News2` to the Tiger corpus. `Hist` includes the historical treebanks Mercurius and ReF.UP. The `Mix` set is a combination of the `News2` and `Hist` sets.

the trees until no crossing branches remain (cf. Ortmann 2021b).[59]

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 treebanks for historical German, Mercurius and ReF.UP, which are annotated according to the Tiger scheme and thus can also be used for chunk extraction. 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 and over 205k chunks of the six relevant types, respectively. Table 6.1 gives an overview of the four training and development sets.

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 that are unrelated to the training data are used for evaluation. 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, originally, includes only a partial annotation of chunks, which was completed for my chunking study, yielding a total amount of 1.5k chunks. The second historical data set, DTA, contains about 6.6k chunks. Table 6.2 gives an overview of the test data.

In Figure 6.1, the distribution of the six chunk types in the test data is shown. As could be

[59] Basically, discontinuous nodes are split and re-inserted into the tree based on the linear order of tokens in the sentence. The same holds for punctuation, which is appended to the same parent node as the next token to the left (or to the right for sentence-initial punctuation).

Corpus	#Docs	#Sents	#Toks	#Words	#Chunks
TüBa-D/Z	364	10,491	196,636	167,847	74,981
Tiger	200	4,445	78,018	67,685	27,253
Modern	78	547	7,605	6,354	2,829
Mercurius	2	818	18,740	16,401	6,691
ReF.UP	26	2,173	61,399	48,820	21,120
HIPKON	53	342	4,210	3,747	1,529
DTA	29	609	18,515	15,822	6,651

Table 6.2.: Overview of the test data. Only sentences containing at least one of the relevant chunk types are included in the evaluation.

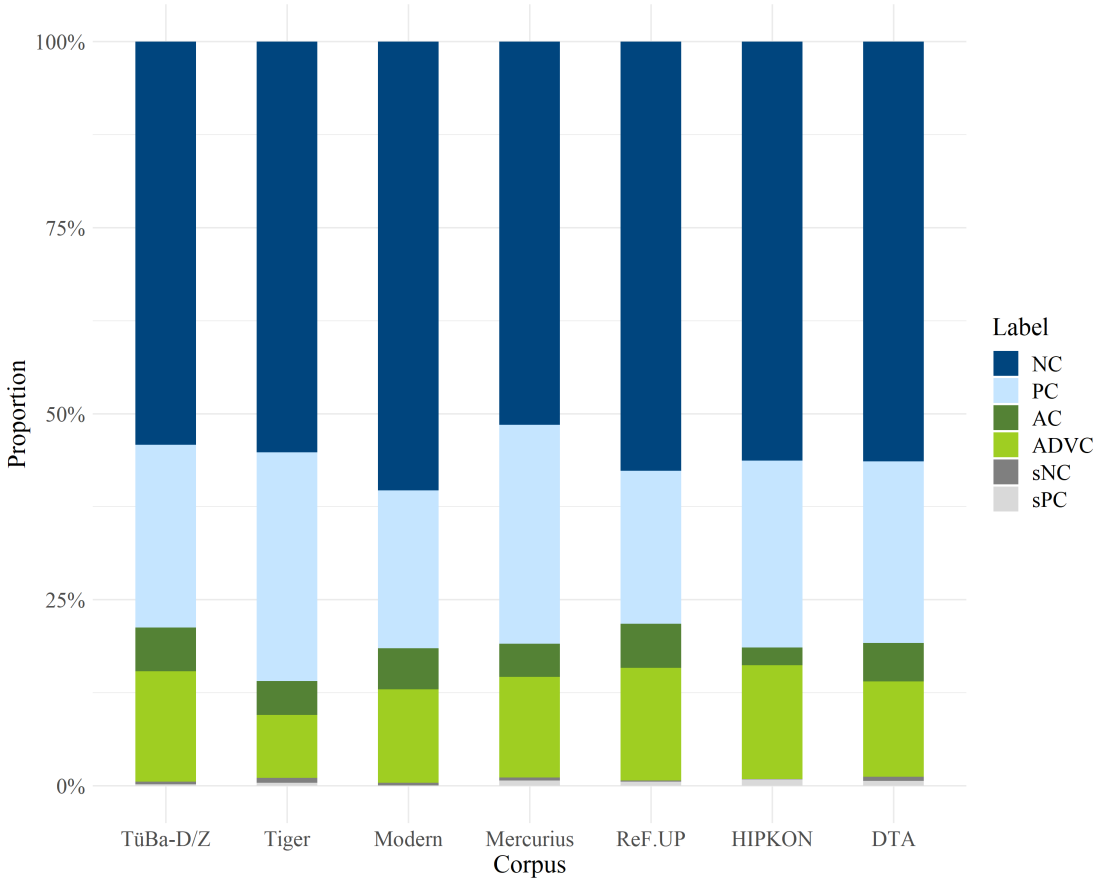


Figure 6.1.: Distribution of chunk types in the test data.

expected, noun chunks (NC) are the most frequent chunk type (51%–60%), followed by prepositional chunks (PC; 20%–30%) and adverb chunks (ADVC; 8%–15%). 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 stranded prepositional chunks (sPC), as in example (33) from the Mercurius corpus, are more common.

(33) [$_{sPC}$ von] [$_{NC}$ der Frantzosen] [$_{PC}$ Vorhaben]
of *the French's* *plan*
'of the plan of the French'

6.1.3. Methods

As detailed in Section 6.1.1, various methods have been applied to the automatic recognition of chunks in modern text. In my experiment, 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 6.1.1. For this study, a simple Regexp chunker as implemented in the NLTK[60] 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 (for the exact rules, see Figure A.4 in the appendix).

The neural sequence labeling tool NCRF++ (Yang and Zhang 2018)[61] 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 joint 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 Mix is trained on a combination of the modern and historical treebanks that follow the same annotation scheme (Tiger, Mercurius, ReF.UP). During training, the tool is provided with the corresponding development data, and each of the models is trained with and without POS tags as 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 pre-trained on Ger-

[60] http://www.nltk.org/api/nltk.chunk.html
[61] https://github.com/jiesutd/NCRFpp

man Wikipedia.[62] To ensure comparability, all models are trained with the same default settings.[63]

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, even though only the six types described at the beginning of Section 6.1 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.

6.1.4. Evaluation and Results

To assess the performance of the automatic methods from the previous section, their output is compared chunk-wise to the gold standard annotation. In my chunking study (Ortmann 2021a), for the first time, I performed an evaluation with fine-grained error types. However, the new error types were counted exclusively as false positives, which makes precision and recall values hard to interpret. In this section, I report the original results, complemented by updated *FairEval* results (according to Chapter 4).

Table 6.3 from Ortmann (2021a) 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% on 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%. As already observed in other studies, models that include POS tags 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

[62] GloVe embeddings trained on German Wikipedia and provided by deepset, https://deepset.ai/german-word-embeddings; Download: December 15, 2020.

[63] 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, this study (Ortmann 2021a) was only a first investigation of chunking historical German and further experiments should be conducted to test for model stability and 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**
`Mix`	+	-	-	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 6.3.: Overall F_1-scores for the Regexp chunker and all NCRF++ models for the seven corpora (Ortmann 2021a). 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.

(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 tags and word embeddings achieves the best results with F_1-scores of about 93%, followed by the `Mix` model. For the HIPKON corpus, the `Mix` model with POS tags 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% by the `Hist` model with POS tags and word embeddings, followed by the `Mix` and the `News1` models with F_1-scores of about 90% as well.

These results are in line with the observations from the previous chapter that models trained on modern newspaper data can successfully be transferred to historical German with overall F_1-scores $\geq$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. For completeness, I reproduced the annotation experiment from Ortmann (2021a) with the four POS-based models and word embeddings. Table 6.4 shows the results for fair evaluation, also including precision and recall values.

	News1			News2			Hist			Mix		
Corpus	Prec	Rec	F_1	Prec	Rec	F_1	Prec	Rec	F_1	Prec	Rec	F_1
TüBa-D/Z	**96.85**	**96.77**	**96.81**	91.45	91.02	91.24	-	-	-	-	-	-
Tiger	90.38	90.99	90.68	**94.99**	**95.04**	**95.01**	-	-	-	-	-	-
Modern	93.56	93.63	93.59	**94.12**	**93.81**	**93.96**	-	-	-	-	-	-
Mercurius	90.61	90.14	90.38	88.56	88.52	88.54	**92.83**	**92.78**	**92.81**	92.24	92.13	92.18
ReF.UP	90.44	89.05	89.74	87.94	87.80	87.87	**93.12**	**92.24**	**92.67**	92.95	91.95	92.45
HIPKON	92.80	92.74	92.77	92.45	92.39	92.42	**93.94**	**93.75**	**93.84**	93.93	93.74	93.84
DTA	90.16	88.65	89.40	87.38	86.70	87.04	90.50	**88.72**	**89.60**	**90.63**	88.57	89.59

Table 6.4.: Overall precision, recall, and F_1-scores (in percent) according to *FairEval* for chunking with the different POS-based models with pre-trained word embeddings. Models trained on historical data are only applied to the historical test sets, and the highest scores for each corpus are highlighted in bold. Traditional evaluation results can be found in Table A.8 in the appendix.

Corpus	NC	PC	AC	ADVC	sNC	sPC
TüBa-D/Z	97.00	97.36	89.81	98.52	76.37	71.24
Tiger	96.04	94.92	89.60	92.80	87.50	73.22
Modern	95.57	93.37	85.71	91.31	80.00	0.00
Mercurius	93.43	92.97	89.32	94.48	0.00	38.60
ReF.UP	94.40	92.96	83.32	90.90	5.41	40.82
HIPKON	96.34	91.47	83.58	92.34	0.00	26.67
DTA	90.96	90.42	82.89	88.58	10.26	16.67

Table 6.5.: Overall F_1-scores for each label (in percent) according to *FairEval* for the best performing model on each data set.

It is important to note that, once again, the experiments in this chapter were conducted with gold standard POS tags. Using automatically assigned POS tags 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 tags. In Ortmann (2021a), I applied the Stanza tagger (Qi et al. 2020, German `hdt` model) to the modern data sets, which resulted in POS error rates of 4% (TüBa-D/Z) to 6% (Modern) and reduced 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 tags as feature decreased 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.

In Table 6.5, the results per chunk type are displayed for the best performing model (with POS and embeddings) on each data set. The best results are observed for noun and prepositional chunks with F_1-scores above 90%, while the results for adjective and adverb chunks range mostly between 83% and 94%. The stranded chunk types are recognized much less reliably, especially in the histor-

ical 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 (33) 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 joint PC, followed by an NC as in example (34).

(34) **Target:** [$_{\text{sPC}}$ von] [$_{\text{NC}}$ der Frantzosen] [$_{\text{PC}}$ Vorhaben]

System: [$_{\text{PC}}$ von der Frantzosen] [$_{\text{NC}}$ Vorhaben]

Since, in these cases, the embedded noun chunk cannot be recognized based on STTS POS tags, a morphological analysis would be 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 likely not enough training data to recognize them reliably.

Finally, Figures 6.3 and 6.2 show the distribution of error types and confusions of labels in the data sets. For all corpora, boundary errors constitute the majority of errors with 50% to 62%, which means that, even in the case of errors, the models often identified the chunks but did not achieve an exact match of the boundaries. This could be considered less severe than completely missing (FN) or made-up chunks (FP), which are infrequent for most data sets. As discussed in Chapter 4, this observation once again highlights the advantages of fair evaluation for a realistic impression of model performance.

6.2. Phrase Identification[64]

The previous section has explored the automatic identification of chunks in modern and historical German. However, chunks are (often) only partial constituents. The identification of complete candidates for extraposition, thus, requires the recognition of larger, more complex units. This section explores the automatic identification of phrases before Section 6.3 targets the identification of relative clauses.

In the context of this thesis, phrases are understood as continuous, non-overlapping constituents from a sentence's parse tree. This section focuses on four main phrase types: noun phrases (NP), prepositional phrases (PP), adjective phrases (AP), and adverb phrases (ADVP). For each sentence, only the highest non-terminal nodes of the given types are considered, ignoring the internal structure of phrases. This means that phrases may dominate other phrases of the same or different types, but the dominated phrases are not evaluated here. Example (35) shows an annotated sentence from a 1731 theological text from the DTA sample.

[64]The content of Section 6.2 is taken from my paper Ortmann (2021b): *Automatic Phrase Recognition in Historical German* and complemented with the Spoken data set and *FairEval* results.

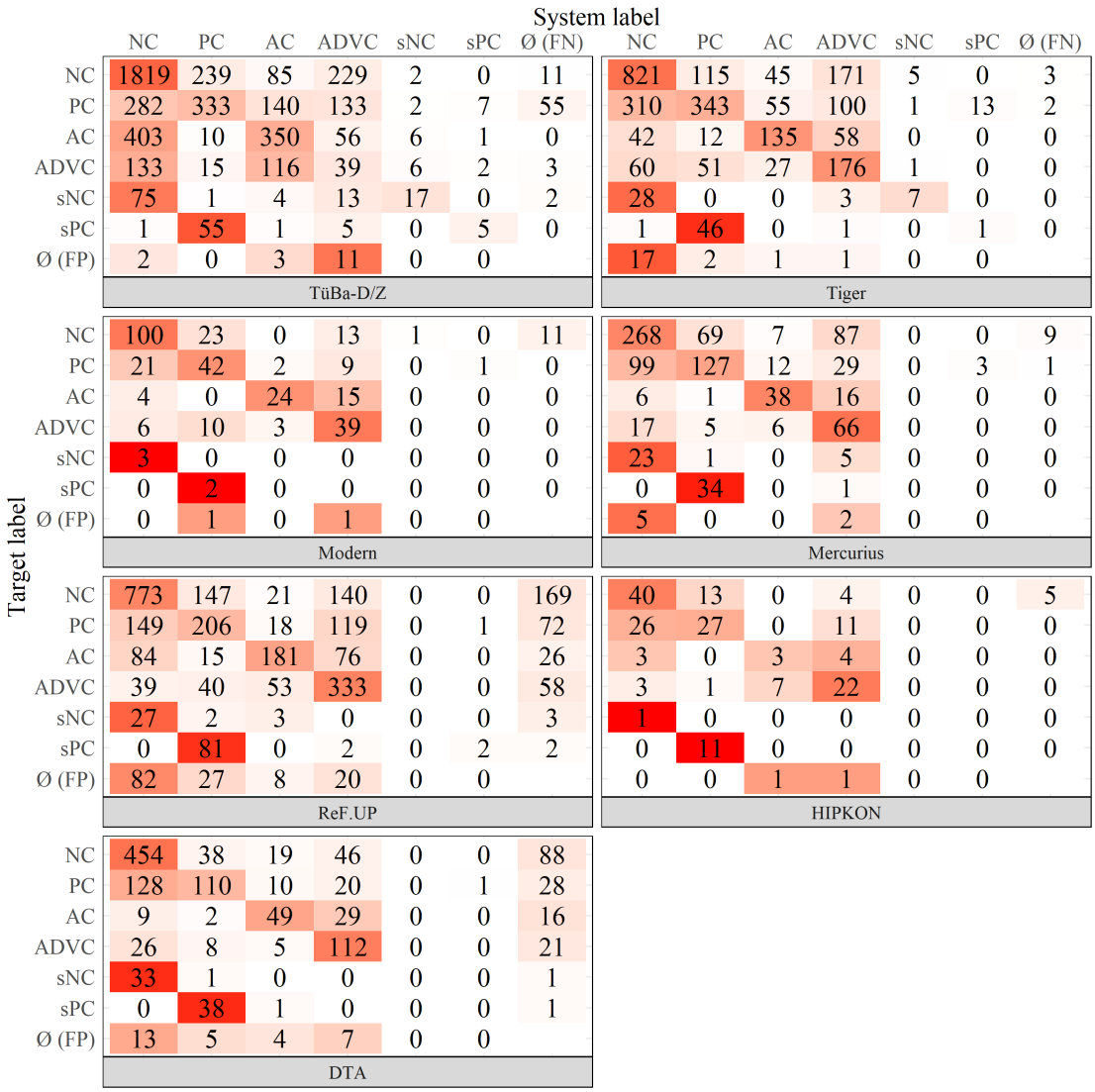


Figure 6.2.: Confusion matrix for the identification of chunks. Only errors are displayed, so the diagonal displays boundary errors.

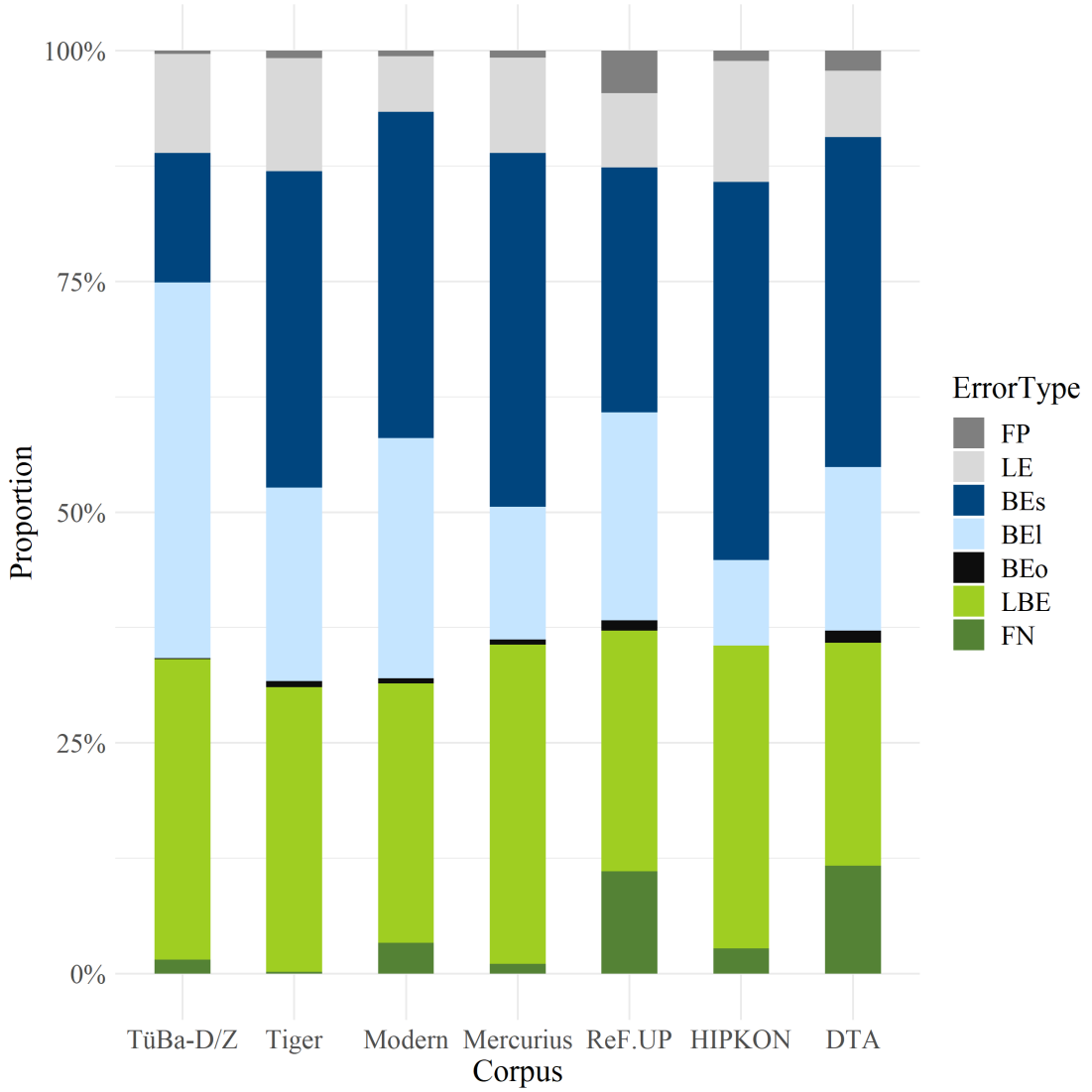


Figure 6.3.: Proportion of the different error types for chunking: false positives (FP), labeling errors (LE), shorter, longer, and overlapping boundary errors (BE_S, BE_I, BE_O), labeling-boundary errors (LBE), and false negatives (FN). Numbers are shown for the best model on each data set.

(35) [NP Der kräftigſte Bewegungs-Grund] nimmt [NP seinen Urſprung] [PP aus einer zärtlichen Leydenſchaft meines Gemůhts].

'The most powerful motive takes its origin from a tender passion of my heart.'

To enable research on extraposition, phrases may not cross topological field boundaries. For example, a prepositional phrase in the middle field is considered separate from an adjacent modifying relative clause in the post-field, as shown in example (36) from a chemistry essay from the DTA data set (field boundaries are indicated by vertical pipes).[65] Also, discontinuous structures as they exist in some German corpora are not allowed here.

(36) Erhebt | [NP es] [NP ſich] [PP mit dem Waſſerſtoffgas], | [NP welches] | [NP die Moråſte] [PP in Ueberfluß] | ausdunſten?

'Does it rise with the hydrogen gas that the swamps evaporate in abundance?'

6.2.1. Related Work

The recognition of phrases, as defined here, is related to chunking as well as (constituency) parsing and can be located somewhere in between the two tasks regarding its complexity. As explained in Section 6.1, chunking refers to the identification of non-overlapping, non-recursive phrases from a sentence's parse tree, ending with the head token (Sang and Buchholz 2000). As a consequence, chunks are often shorter than phrases because post-modifying elements form separate chunks. For simple cases without pre- or post-modifying elements, the definitions of chunks and phrases overlap. Thus, methods that are successful at chunking may also be useful for phrase recognition.

Parsing, on the other hand, aims at a complete syntactic analysis of the sentence. Hence, the resulting constituency tree includes more information than just the phrase annotation, e.g., dominance relations, which are not considered in this study. As a result, phrase annotations can be derived from the more complex parse output, but the complexity of the task may also reduce overall accuracy.

While studies on chunking observe F_1-scores >95% for modern German (cf. Müller 2005; Ortmann 2021a), the highest F_1-scores for constituency parsing of German are reported with approx. 90%, compared to 95% for English (Kitaev et al. 2019). In general, parsing results heavily depend on the selected treebank and the inclusion of grammatical functions (Dakota and Kübler 2017) and discontinuous structures (cf. Vilares and Gómez-Rodríguez 2020). Also, all of these results are obtained for standard language like newspaper text. For non-standard data, performance drops must be expected (Pinto et al. 2016; Jamshid Lou et al. 2019).

[65] Actually, the position of the relative clause is ambiguous since the right sentence bracket is empty and the RelC could be placed either in the middle field or in the post-field, cf. the discussion in Chapter 7. I will follow the guidelines by Telljohann et al. (2017) in considering the relative clause as part of the post-field and, thus, separate from its antecedent. In Chapter 7, this decision will allow to distinguish between unambiguously embedded RelCs and ambiguous or extraposed RelCs.

For historical German, so far, there have been experiments on chunking (Petran 2012; Ortmann 2021a; cf. Section 6.1), topological field parsing (Chiarcos et al. 2018; Ortmann 2020; cf. Chapter 5), and statistical constituency parsing (Hinrichs and Zastrow 2012), but no evaluation results exist for the latter. For chunking, the best results are observed for CRF-based sequence labeling, with overall F_1-scores between 90% and 94% (Ortmann 2021a). For topological field identification, the application of a probabilistic parser yields overall F_1-scores $\geq$90% (Ortmann 2020). In this section, both of these approaches will be explored for the purpose of phrase recognition in historical German.

6.2.2. Data

The training data for the experiment is the same as in Section 6.1, consisting of two modern (TüBa-D/Z, Tiger) and two historical (Mercurius, ReF.UP) treebanks. All four data sets are annotated with constituency trees, but before they can be used to train a parser or extract phrase annotations for sequence labeling, a few modifications are necessary.

(i) The underlying annotation scheme of the Tiger corpus and the two historical treebanks allows for discontinuous annotations, which must be removed to enable the use of standard chunking and parsing methods (see the remarks in Section 6.1.2).

(ii) Since German exhibits a relatively free word order, grammatical functions like subject and object play an important role in the syntactic analysis of sentences, especially for the reduction of ambiguity (Fraser et al. 2013). For the purpose of phrase recognition, however, they are not relevant and, therefore, mostly excluded from the trees to reduce the size of the tagset and improve parsing performance (Rafferty and Manning 2008; Dakota and Kübler 2017).[66]

The modified trees can serve as training input for a parser, or they can be used to extract phrase annotations. Contrary to chunking, where the lowest non-terminal nodes are converted to chunks (Kübler et al. 2010; Ortmann 2021a), here, the highest non-terminal nodes of the relevant types correspond to the desired phrases.[67] Before the extracted phrases can be used for evaluation or to train a sequence labeling tool, another difference between the annotation schemes of the treebanks regarding topological fields must be taken into account.

[66] The only exception are GFs that are needed to extract correct phrases (and relative clauses) from the trees. For the Tiger scheme, these are `S:RC` and `S:OC`. For TüBa-D/Z, the following GFs are preserved: `KONJ`, `OS`, `R-SIMPX`, `NX:HD` dominated by `PX`, and `NX:APP` dominated by `NX`. Also, one-word children of sentence nodes that only receive a grammatical function according to the Tiger scheme are assigned a phrase type `NP`, `PP`, `AP`, `AVP`, `VP`, or `SVP` based on their POS tag.

[67] Again, phrases of the four types are added for one-word constituents from Tiger-style trees based on the POS tag of the word.

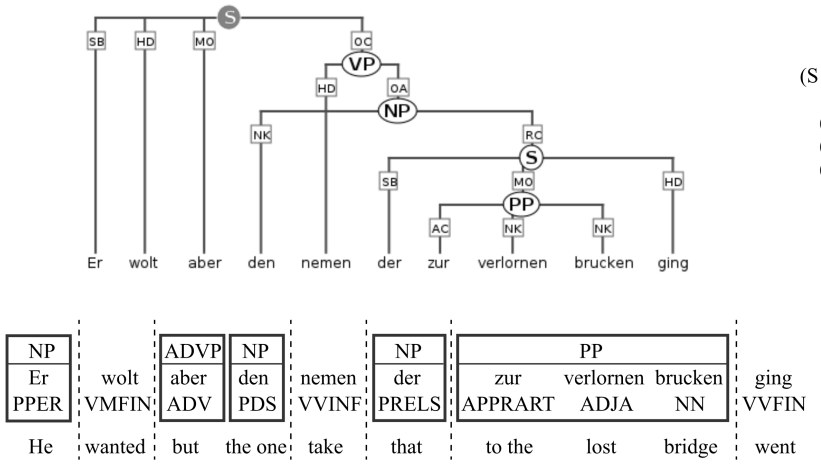


Figure 6.4.: Example modification of a sentence from the ReF.UP corpus. At the top, the original constituency tree with discontinuous annotations according to the Tiger scheme is displayed (image taken from https://annis.linguistics.rub.de/?id=23b13a12-16cb-4258-919e-2f31a53e24f7). The bracket structure to the right represents the linearized version of the tree without crossing branches and grammatical functions. This format can be used to train a standard parser. At the bottom, the phrase annotation for the sentence is shown. The phrases have been extracted from the tree structure to the right and checked with a topological field parser to ensure that phrases do not cross field boundaries (indicated by dashed lines). The phrase annotations serve as training data for a sequence labeling tool and are also used for evaluation.

Model	#Docs	#Sents	#Toks	#Phrases	#RelCs
News1	3,075	83,515	1,566,250	388,531	12,480
News2	1,863	40,037	727,011	162,336	5,177
Hist	28	23,747	569,854	152,866	5,431
Mix	1,891	63,784	152,866	315,202	10,608

Table 6.6.: Overview of the training data for each of the models. Only sentences with a gold parse are included, and the number of phrases refers to phrases of the four relevant types. #RelCs is the number of relative clauses, which are relevant for Section 6.3. News1 corresponds to the TüBa-D/Z and News2 to the Tiger corpus. Hist includes the historical treebanks Mercurius and ReF.UP. The Mix set is a combination of the News2 and Hist sets.

Corpus	#Docs	#Sents	#Toks	#Words	#Phrases
TüBa-D/Z	364	10,488	196,630	167,844	49,329
Tiger	200	4,445	78,018	67,685	17,622
Spoken	14	23,937	285,594	234,094	106,945
Modern	78	547	7,605	6,354	2,240
Mercurius	2	818	18,740	16,401	4,400
ReF.UP	26	2,173	61,399	48,820	15,355
HIPKON	53	342	4,210	3,747	1,146
DTA	29	609	18,515	15,822	4,400

Table 6.7.: Overview of the test data. Only sentences containing at least one phrase of the four types are included in the evaluation.

(iii) While TüBa-style trees represent a combination of constituency and topological field annotations, the other three corpora that follow the Tiger scheme do not include topological fields. This means that constituents in the TüBa-D/Z data are already bound to the corresponding fields as required by the phrase definition in this study, whereas constituents in the other data sets may cross field boundaries. Therefore, phrases that are extracted from these data sets or identified by a parser that is trained on them are corrected with the help of the topological field parser (`Punct` model) from Chapter 5. Phrases that cross fields are split at the field boundary and replaced by the dominated sub-phrases to ensure that no phrase is located in more than one field.[68]

An example of the different modifications of the trees and extracted phrases can be found in Figure 6.4. The resulting data sets are used to build four distinct training sets: `News1` corresponds to the TüBa-D/Z data, `News2` is based on the Tiger treebank, `Hist` contains the historical data, and a joint set `Mix` includes all data sets that follow the Tiger annotation scheme. Table 6.6 gives a summary of the four training sets.

For evaluation, the test sections of the four treebanks are processed in the same way as the training data, and phrases of the four types are extracted and split at topological field boundaries if necessary. The Spoken data set was not included in the original study (Ortmann 2021b) but is added here, too. In addition, the other three test sets from the previous section (Modern, HIPKON, DTA) with manually annotated phrases are included in the study. Table 6.7 gives an overview of the test data. In Figure 6.5, the distribution of phrase types in the test data is displayed. The most frequent phrase type are `NPs` with 49% to over 60% in the Modern data set, followed by `PPs` with 13% to 27%. `ADVPs` make up for 11% to 32%, while `APs` that are not dominated by other phrases are rare with 6% or less.

[68] Theoretically, it would also be possible to merge the constituency trees with automatically created topological field annotations before training a parser on the merged trees. However, experiments indicate that this creates too many inconsistencies in the training data, e.g., due to errors in the automatic field annotation, and therefore leads to worse results than splitting the extracted phrase output at the field boundaries afterwards.

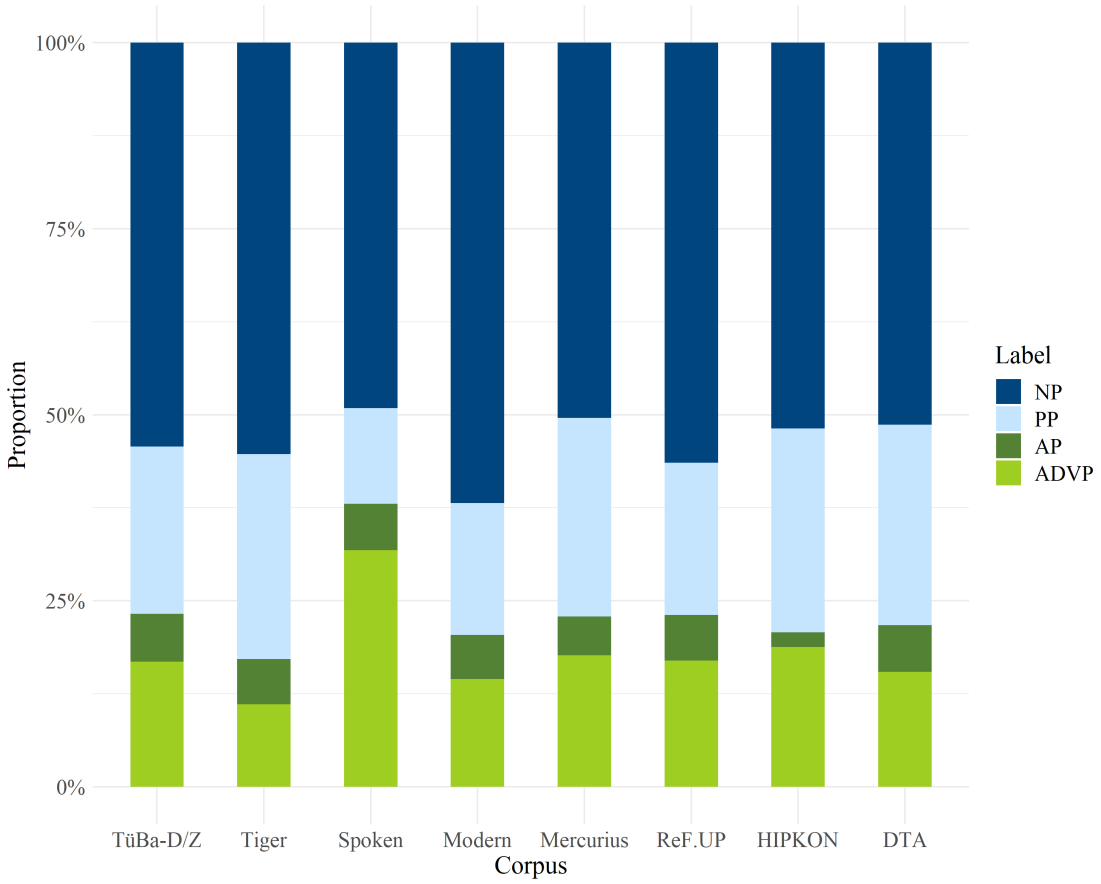


Figure 6.5.: Distribution of phrase types in the test data.

6.2.3. Methods

So far, the automatic syntactic analysis of historical German has been focused on the identification of chunks and topological fields. As described in the previous sections, the best results for these tasks are reported for sequence labeling and statistical parsing. In the following, both approaches are applied to the recognition of phrases.

For sequence labeling, the neural CRF-based sequence labeling tool NCRF++ (Yang and Zhang 2018) is selected. It achieves state-of-the-art performance for several tasks, including tagging, chunking, and named entity recognition in English (Yang et al. 2018). When POS tags are used as features, it also proves successful at identifying chunks in historical German with F_1-scores >90% (Ortmann 2021a; cf. Section 6.1). The default configuration consists of a three-layer architecture with a character and a word sequence layer plus a CRF-based inference layer. For this study, the toolkit is trained on the extracted phrases from the four training sets, where phrases are represented

as BIO tags. POS tags are included as additional feature and, during training, the tool is also provided with the development sections of the training corpora. For every word, NCRF++ outputs the single most likely BIO tag, i.e., `B-XP` (beginning of phrase), `I-XP` (inside of phrase), or `O` (outside of phrase). For evaluation, the labels are converted to phrases, and the best result over five runs with different random seeds is reported.

For parsing, the unlexicalized Berkeley parser (Petrov et al. 2006) is used.[69] It achieves a parsing F_1-score of 91.8% on the TüBa-D/Z corpus and 72% on the Tiger corpus (Dakota and Kübler 2017) and has also been successfully applied to topological field parsing of historical German with overall F_1-scores $\geq$90% (Ortmann 2020; cf. Chapter 5). In this study, it is trained with default settings[70] on the four training sets, where the modified constituency trees are used as training input. For annotation, the parser is invoked in interactive mode.[71] Given a sentence annotated with POS tags, it returns the single best parse. For evaluation, the constituency trees are then converted to phrases, as described in the previous section.

6.2.4. Evaluation and Results

To evaluate the performance of the selected approaches on the task of phrase recognition, the output of the trained systems is compared phrase-wise to the gold standard annotation. In the original phrase recognition study (Ortmann 2021b), I already performed an evaluation with fine-grained error types. But the new error types were counted exclusively as false positives, which made precision and recall values hard to interpret. In this thesis, instead of repeating the complete experiment, I report the original results for NCRF++ from Ortmann (2021b), and updated *FairEval* results for the Berkeley parser, also including the Spoken data set. In all experiments, only sentences containing at least one of the four phrase types are evaluated, and punctuation at phrase boundaries is ignored.

Sequence Labeling

As already mentioned, the neural sequence labeling tool NCRF++ has been applied successfully to the identification of chunks in German, reaching F_1-scores between 90% and 94% for different historical data sets (Ortmann 2021a, cf. Section 6.1). As could be expected from previous studies (e.g., Petran 2012), the accuracy for the recognition of phrases, i.e., longer units, with CRF-based sequence labeling is considerably lower. Table 6.8 (from Ortmann 2021b) gives a summary of the results for each of the four models.

[69] Even though the parser is unlexicalized and called with the `-useGoldPOS` flag, it can de-facto take the word forms into account. When no possible parse for the input POS sequence is found, the (undocumented) behavior is to generate new tags for the words. Contrary to the topological field model that is directly based on POS tags as terminal nodes, the constituency models thus may actually consider the word forms during parsing (albeit to an unknown extent).

[70] `java -cp BerkeleyParser-1.7.jar edu.berkeley.nlp.PCFGLA.GrammarTrainer -treebank SINGLEFILE -out grammar.gr -path treebank.txt`

[71] `java -jar BerkeleyParser-1.7.jar -gr grammar.gr -maxLength 350 -useGoldPOS`

Corpus	News1	News2	Hist	Mix
TüBa-D/Z	**85.18**	76.82	-	-
Tiger	78.93	**79.69**	-	-
Modern	**86.80**	83.10	-	-
Mercurius	**70.25**	67.83	9.05	8.93
ReF.UP	**70.62**	67.91	8.80	9.90
HIPKON	80.13	**81.18**	8.17	7.99
DTA	**72.02**	68.89	6.93	7.78

Table 6.8.: Overall F_1-scores for phrase recognition with the sequence labeling approach (Ortmann 2021b). Models trained on historical data are only applied to the historical test sets. The table reports the highest F_1-score over five runs and the best result for each corpus is highlighted in bold.

Using gold POS tags as feature, the two newspaper-based models still perform relatively well. Model News1 achieves the best results with F_1-scores between 70% and 87%. The results for the second modern model News2 also lie above 67% for all data sets. Contrary to the results for chunking (Ortmann 2021a), using historical training data does not improve the results on the historical test sets. Instead, the historical and mixed models do not reach F_1-scores >10% for phrase recognition, indicating that the tool was not successful at learning to identify the different phrase types based on the historical corpora. Possible reasons could be the high syntactic complexity of Early New High German sentences or too much variation in the training data, e.g., caused by the unstandardized spelling in historical German. Perhaps, using automatically generated chunks (Chapter 6.1) as additional features could improve the results of the sequence labeling approach.

Parsing

So far, the parsing approach has been evaluated only for topological field parsing of historical German, with overall F_1-scores $\geq$90% (Ortmann 2020). In Table 6.9, the results of the Berkeley parser for the recognition of phrases are given. On the modern data sets, the parser achieves F_1-scores of 86% to 91%, with visible differences between the two modern models. While, unsurprisingly, each of them performs best on the test section of the corpus it was trained on, the News1 model also achieves the best results on the Modern and Spoken data sets and the DTA corpus, while the News2 model performs slightly better on the other historical data sets.

In contrast to the sequence labeling results, here, including historical training data improves the syntactic analysis of historical language – probably because the unlexicalized parser is unaffected by the unstandardized spelling or can better handle the complex sentence structures. For three of the four historical data sets, the Hist and Mix models outperform the modern models by ten percentage points or more. F_1-scores lie between 81% and almost 85% for the Mercurius, ReF.UP, and HIPKON data, while the DTA is only analyzed with an F_1-score of 71.5%.

	News1			News2			Hist			Mix		
Corpus	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
TüBa-D/Z	**91.35**	**91.35**	**91.35**	82.04	81.10	81.57	-	-	-	-	-	-
Tiger	83.49	84.75	84.11	**85.66**	**87.14**	**86.39**	-	-	-	-	-	-
Spoken	**88.98**	**89.84**	**89.41**	80.26	82.30	81.27	-	-	-	-	-	-
Modern	**88.27**	**88.23**	**88.25**	84.77	84.11	84.44	-	-	-	-	-	-
Mercurius	61.81	64.34	63.05	66.00	65.77	65.88	**81.25**	**81.82**	**81.53**	81.04	81.29	81.16
ReF.UP	58.55	58.93	58.74	59.07	59.03	59.05	**84.02**	**84.30**	**84.16**	83.98	84.15	84.07
HIPKON	74.54	74.75	74.64	75.10	75.45	75.27	**84.88**	**84.96**	**84.92**	84.69	84.77	84.73
DTA	**73.03**	**70.07**	**71.52**	69.61	64.90	67.17	69.10	64.80	66.88	70.45	66.75	68.55

Table 6.9.: Overall precision, recall, and F_1-scores (in percent) according to *FairEval* for phrase recognition with the different parser models on each data set. Models trained on historical data are only applied to the historical test sets, and the highest scores for each corpus are highlighted in bold. Traditional evaluation results can be found in Table A.9 in the appendix.

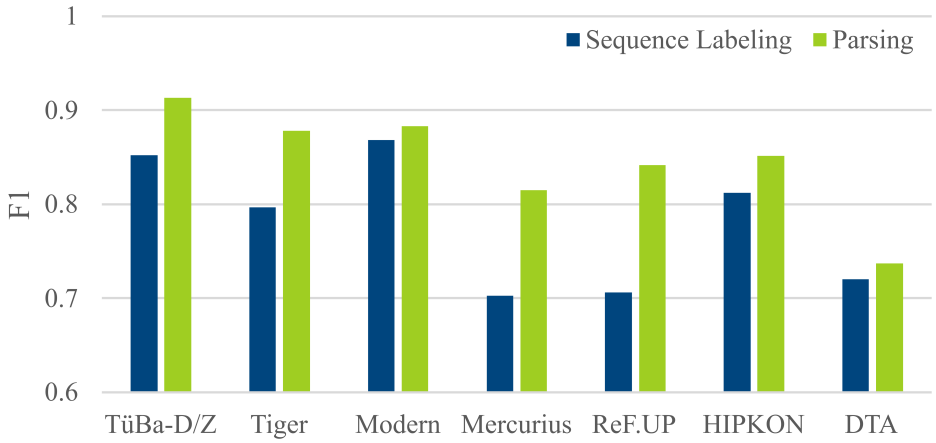


Figure 6.6.: Comparison of the best F_1-scores for sequence labeling and parsing on the different test sets (Ortmann 2021b).

When compared to the sequence labeling tool, the parsing approach consistently yields better results for the recognition of phrases. Figure 6.6 from Ortmann (2021b) confirms that the best parser model outperforms the best sequence labeling model by up to 13.5 percentage points on each data set. Only for the Modern data set and the DTA, the results of the methods are similar. For the Modern data, this could be due to the fact that the data set contains many non-complex phrases that are similar to chunks, e.g., simple noun phrases. 54% of the phrases in this data set consist of only one token, compared to 35%–50% in the other data sets, which makes it easier for the sequence labeling approach to identify them.

Corpus	NP	PP	AP	ADVP
TüBa-D/Z	91.32	88.73	90.99	95.03
Tiger	87.69	84.82	79.86	87.18
Spoken	90.89	77.55	84.81	92.68
Modern	90.30	81.98	83.92	88.82
Mercurius	82.07	77.63	71.81	88.48
ReF.UP	86.40	80.72	69.25	85.81
HIPKON	85.84	81.32	66.67	89.50
DTA	71.10	70.14	73.98	74.14

Table 6.10.: Overall F_1-scores for each label (in percent) according to *FairEval* for the best performing parser model on each data set.

However, parser accuracy also declines for longer units (cf. Bastings and Sima'an 2014). While the Berkeley parser reaches overall parsing F_1-scores of 92% and 86% for the modern data and 78%–79% for the historical data (cf. Table A.10 from Ortmann 2021b in the appendix), F_1-scores heavily decline for longer constituents and phrases (see Figure 6.7 from Ortmann 2021b). For constituents with more than five words, the average F_1-score of the four models is only about 70%. For phrases, the reduction is even larger with F_1-scores below 40% for phrases of twenty or more words. This observation may, in part, explain the lower results for the DTA because, proportionally, this data set contains about twice as many phrases of twelve or more words than the other corpora, due to many dedications and very long phrases with coordinations and dominated sentences, e.g., in legal texts. A parser that performs better on longer constituents might be better equipped to analyze this data set.

Table 6.10 reports the parser results broken down by phrase types. For most data sets, the highest F_1-scores are reached for adverb and noun phrases. While the former are usually very short and therefore easier to identify, noun phrases and prepositional phrases often contain pre- and/or post-nominal modifiers including longer constituents like relative clauses that lead to errors in the parser output. Adjective phrases are the least frequent phrase type and, although they tend to be short, also show the least accurate results for more than half of the data sets. Often they get mixed up with neighboring adverbs (cf. Figure 6.8) because a lexicalized model would be necessary to distinguish between pre-modifying adverbs as in example (37) and a separate adverb phrase in (38).

(37) Sie war [$_{\texttt{AP}}$ sehr/ADV glücklich/ADJD].
'She was very happy.'

(38) Sie war [$_{\texttt{ADVP}}$ gestern/ADV] [$_{\texttt{AP}}$ glücklich/ADJD].
'Yesterday, she was happy.'

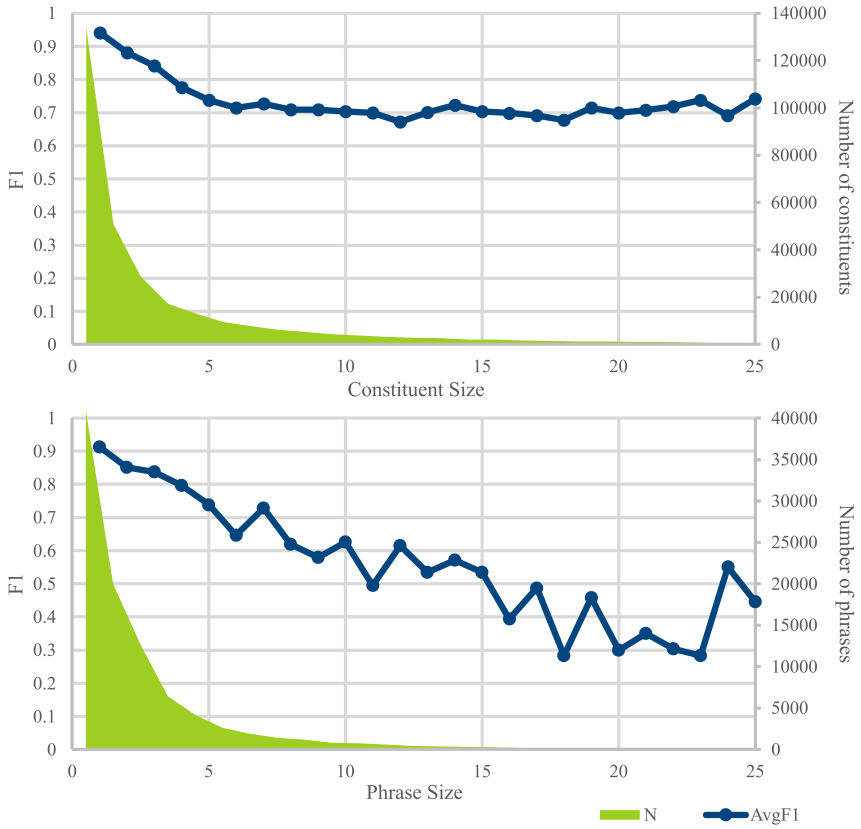


Figure 6.7.: Average F_1-score of the four parser models for the recognition of constituents and phrases of sizes 1–25. The number of constituents includes all constituents of the given sizes in the test sections of the four training corpora. The number of phrases refers to phrases of the four types in the seven test sets from Ortmann (2021b).

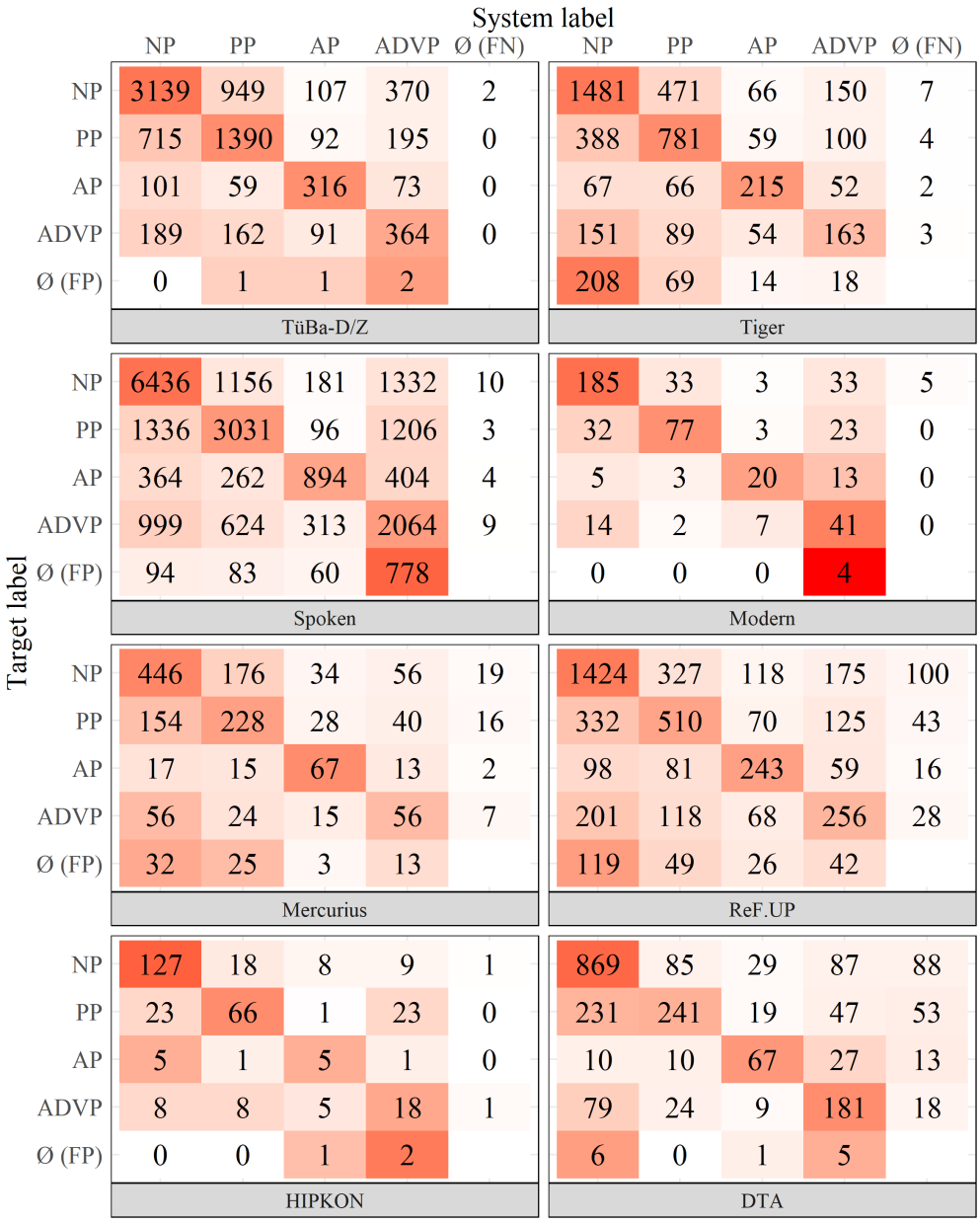


Figure 6.8.: Confusion matrix for the identification of phrases. Only errors are displayed, so the diagonal displays boundary errors.

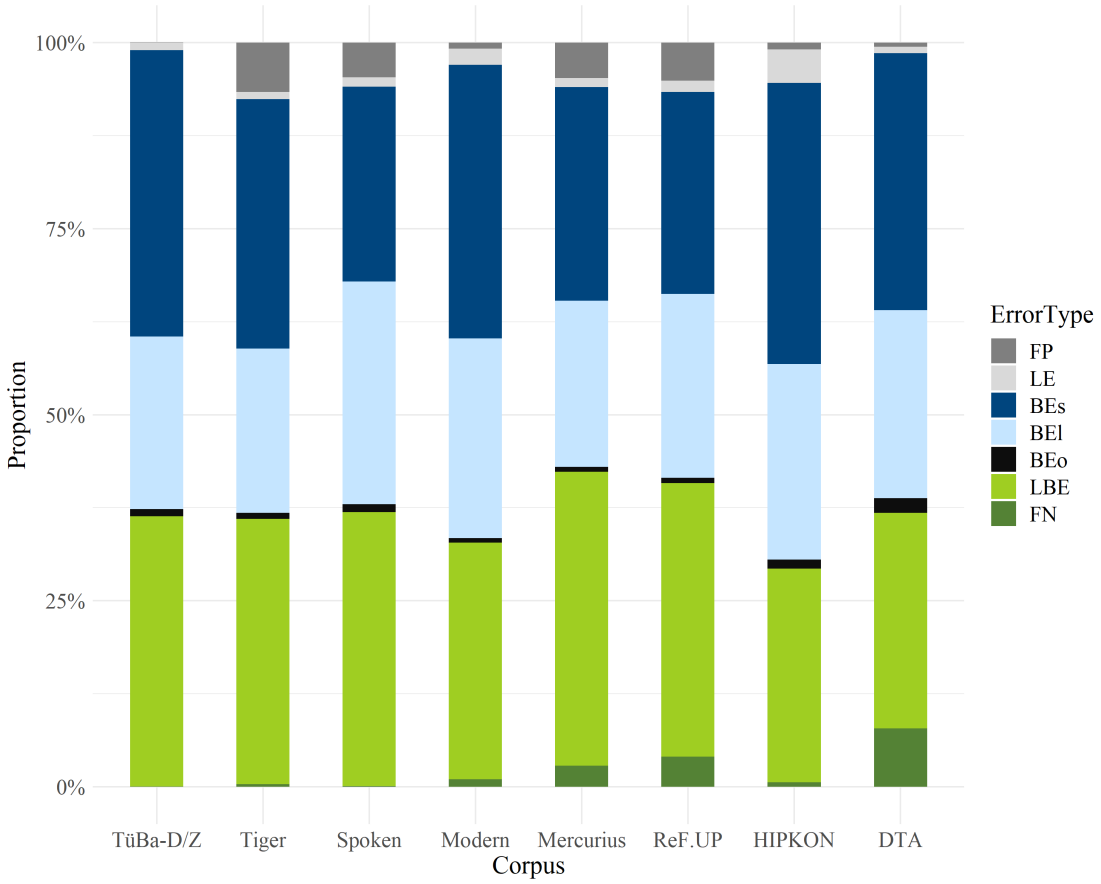


Figure 6.9.: Proportion of the different error types for phrase recognition: false positives (FP), labeling errors (LE), shorter, longer, and overlapping boundary errors (BE_s, BE_l, BE_o), labeling-boundary errors (LBE), and false negatives (FN). Numbers are shown for the best parser model on each data set.

Finally, Figure 6.9 shows the distribution of error types for the best parser models. For all test sets, boundary errors are the most frequent error types with a proportion of 51% to 65%. The remaining errors are mostly labeling-boundary errors, while traditional false positives and false negatives are infrequent. Considering that the identification of phrases with almost correct boundaries may still satisfy the requirements of certain tasks, this can thus be assumed for a large proportion of the errors. Furthermore, the results suggest potential for improvement because the high percentage of boundary errors means that the parser already identified these phrases, and correcting boundaries could potentially lead to significantly higher accuracy.

Corpus	#Docs	#Sents	#Toks	#Words	#RelCs
TüBa-D/Z	364	10,488	196,630	167,844	1,620
Tiger	200	4,445	78,018	67,685	566
Spoken	14	23,937	285,594	234,094	333
Modern	78	547	7,605	6,354	65
HIPKON	53	342	4,210	3,747	46
DTA	29	609	18,515	15,822	171

Table 6.11.: Overview of the test data. Only sentences containing at least one candidate for extraposition (phrase or relative clause) are included in the evaluation.

6.3. Relative Clause Identification

The previous section has explored the identification of phrases. In this section, the experiments are extended to relative clauses (RelCs). Although the focus of this thesis is on attributive RelCs, in this section, I will start with the identification of relative clauses in general.

Conveniently, the RelC annotation is already included in the constituency trees output by the Berkeley parser (cf. Section 6.2) and can simply be extracted from them. In addition to the tree nodes that are explicitly labeled as relative clauses,[72] I also consider sentences whose first constituent is a relative pronoun or dominates one. Based on experiments with the development data, this can be expected to increase annotation recall. For (historical) corpora with custom POS tagsets, both the original and STTS tags are consulted for identifying the relativizers (cf. the discussion in Chapter 5).

Since the extraction of relative clauses from the two historical treebanks is a bit complicated (and no information about the antecedent or topological fields is provided), only the other six data sets from the previous section are used for the evaluation of RelCs. Table 6.11 gives an overview of the test data.

In Table 6.12, the results for RelC identification with the Berkeley parser and the models from Section 6.2 are shown. For the modern data sets, F_1-scores lie between 91% and 96%, which is 2–10 percentage points higher than for phrase recognition. For the historical data sets, scores are 4–5 percentage points higher, with about 77% and 88%. It can be assumed that this difference is due to the distinctive structure of RelCs, which always start with some relativizer and usually end with a verb, whereas phrase boundaries are not as clear. In general, precision is always higher than recall, and models that performed best on a data set in Section 6.2 also achieve the highest scores for RelC identification (except for the DTA).

[72] Labels for relative clauses are `R`, `R-SIMPX`, `R-SIMPX:KONJ` (TüBa-style) or `S:RC` (Tiger-style).

	News1			News2			Hist			Mix		
Corpus	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
TüBa-D/Z	**96.47**	**95.85**	**96.16**	91.74	90.45	91.09	-	-	-	-	-	-
Tiger	94.64	94.64	94.64	**96.52**	**96.17**	**96.34**	-	-	-	-	-	-
Spoken	**92.13**	**90.35**	**91.23**	90.94	88.27	89.59	-	-	-	-	-	-
Modern	**93.44**	**91.94**	**92.68**	88.89	87.39	88.14	-	-	-	-	-	-
HIPKON	85.71	68.18	75.95	86.67	59.09	70.27	**89.66**	87.64	**88.64**	88.64	**88.64**	**88.64**
DTA	**81.95**	72.19	76.76	76.92	69.93	73.26	77.09	72.35	74.65	77.89	**75.77**	**76.82**

Table 6.12.: Overall precision, recall, and F_1-scores (in percent) according to *FairEval* for RelC recognition with the different models on each data set. Models trained on historical data are only applied to the historical test sets, and the highest scores for each corpus are highlighted in bold. Traditional evaluation results can be found in Table A.11 in the appendix.

Figure 6.10 shows the distribution of error types. For most data sets, boundary errors are by far the most frequent error type (30%–90%). Except for the Modern and DTA data sets, the parser tends to identify RelCs as too long rather than too short. Relevant proportions of false negatives and false positives are only observed for the historical data sets. However, it has to be kept in mind that there is only a small absolute number of errors (TüBa-D/Z: 120, Tiger: 40, Spoken: 54, Modern: 9, HIPKON: 10, DTA: 67), so the error distribution may not be representative for some of the data sets.

6.4. Discussion

In this chapter, I have explored the automatic identification of candidates for extraposition, starting with chunks before proceeding to more complex constituents. As expected from the literature, evaluation results are better for shorter units (chunks) compared to longer ones (phrases), while relative clauses are recognized with high accuracy in most data sets, despite their length. The remaining errors are often caused by incorrect boundaries, which leaves room for further improvement.

Contrary to the previous chapter on topological field parsing, historical training data was available for the syntactic annotation studies in this chapter. Interestingly, the inclusion of the historical data improved the results for chunking and phrase recognition with a constituency parser but not for phrase recognition with the sequence labeling tool. One possible explanation I found in (Ortmann 2021b) could be the more complex task in combination with too much variation in the data due to the unstandardized spelling in historical German. The variation of word forms does not affect the unlexicalized parser but may prevent more accurate analyses with lexicalized (neural) models. Future studies could experiment with spelling normalization, which was observed to improve the annotation results of modern NLP tools for dependency parsing of Middle English (Schneider et al. 2015) or tagging historical German (Bollmann 2013) and Dutch (Tjong Kim Sang et al. 2017).

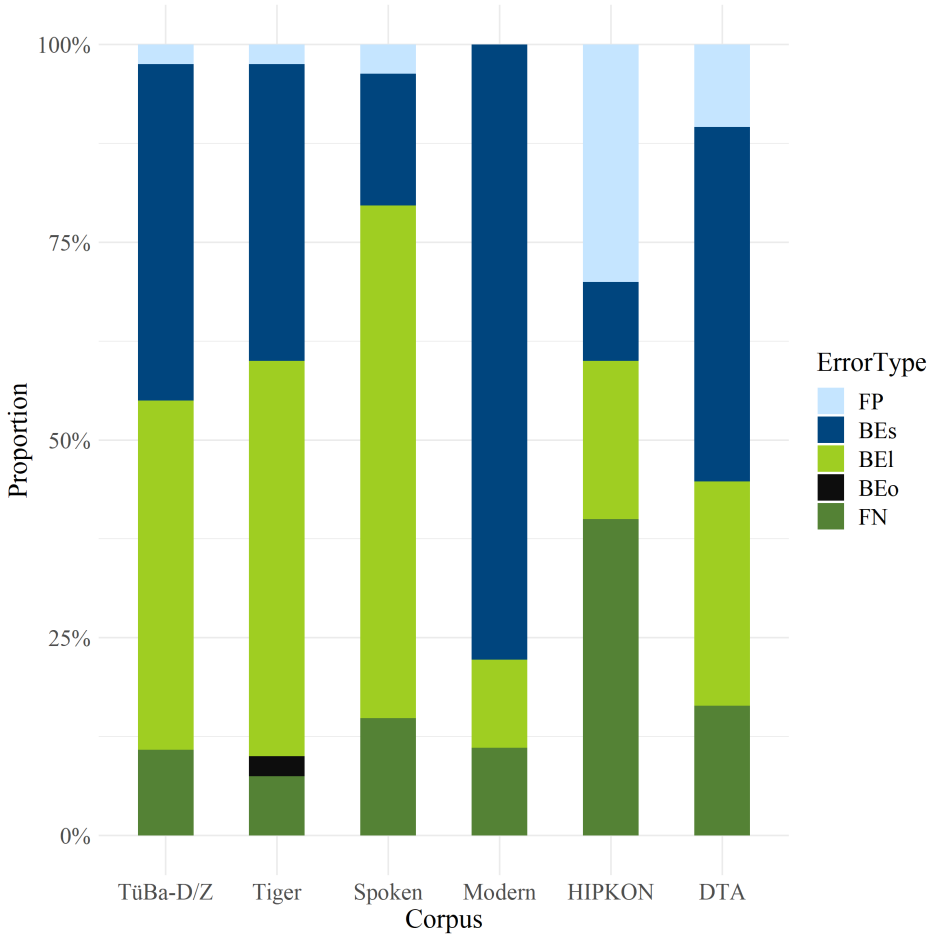


Figure 6.10.: Proportion of the different error types for RelC recognition: false positives (FP), shorter, longer, and overlapping boundary errors (BE_s, BE_l, BE_o), and false negatives (FN). Only one label (RELC) is evaluated, so labeling and labeling-boundary errors cannot occur. Numbers are shown for the best model on each data set.

The normalized data could then also be used to explore lexicalized parsing, e.g., with the neural Berkeley parser (Kitaev and Klein 2018). Although parsers do not necessarily need lexical information for good performance (Coavoux et al. 2019), studies on modern English show that the application of neural parsing methods in combination with pre-trained word embeddings can further improve the results (cf., e.g., Vilares and Gómez-Rodríguez 2020). For morphologically more complex languages like German, this should be even more relevant (Fraser et al. 2013) and could also help in cases where lexical information is necessary to decide about the correct phrase boundaries.

CHAPTER 7

Automatic Analysis of Extraposition

The preceding Chapters 5 and 6 have built the foundation for the automatic recognition of extraposition. First, I explored the topological field analysis and trained models to automatically find the post-field in modern and historical German (Chapter 5). Then, I developed methods to identify selected candidates for extraposition that could be moved from the middle field to the post-field (Chapter 6). I focused on elements that are expected to show at least some variability concerning their position in the middle field and post-field, namely noun phrases, prepositional phrases, adjective and adverb phrases, and (attributive) relative clauses. In this chapter, the puzzle pieces are put together for a completely automatic analysis of extraposition.

The chapter consists of three parts. Section 7.1 deals with the base position of extraposed elements, i.e., the original or unmarked position in the middle field, and explores the automatic identification of antecedents for attributive constituents. Given the annotations from Chapters 5–6 and information about the base position, Section 7.2 then describes how to decide whether a constituent is extraposed, left *in situ*, or if the position is ambiguous. Finally, Section 7.3 explains how these results can be used to automatically inspect the effects of extraposition with a corpus of variants, in which the extraposed constituents have been artificially moved back to their base position. The chapter concludes with a discussion in Section 7.4.

7.1. Base Position

The definition of extraposition from Chapter 2 assumes that extraposed constituents have been 'moved' from the middle field (or sometimes the pre-field) to the post-field of the sentence. That entails that the original, unmarked position of the constituents is somewhere in the middle field (or pre-field). I will refer to this original position as the 'base position' of the constituent. For example, the unmarked base position of the `PP` in example (39a) from the Modern data set would be in the middle field as in (39b).

(39) a. Das **ist** mir ganz klar **geworden**, schon bei dieser kurzen Trennung.

b. Das **ist** mir schon bei dieser kurzen Trennung ganz klar **geworden**.
'That has become very clear to me, even from this short separation.'

However, due to the relatively free word order in German, especially in the middle field, the base position depends on several interrelated factors (e.g., grammatical aspects and information struc-

ture) and is not always easy to determine for all extraposed elements, especially for adjuncts and complements (Frey and Pittner 1998; Lenerz 1977, among others). For instance, two base positions (a vs. b) are equally possible for the extraposed PP in example (40) from the TüBa-D/Z corpus, and even more alternatives exist with an increasing number of constituents in the middle field.

(40) Würdest du den Mönchen, **die** jeden Tag **meditieren** hinter ihren Mauern, auch sagen, sie hätten einen Knastkoller?

a. ... den Mönchen, **die** jeden Tag hinter ihren Mauern **meditieren**, ...

b. ... den Mönchen, **die** hinter ihren Mauern jeden Tag **meditieren**, ...

'Would you also tell the monks who meditate every day behind their walls that they suffer from prison-induced madness?'

As a consequence, currently, only the base position of attributive constituents like relative clauses can be unambiguously identified automatically. Their designation as 'attributive' or 'relative' already indicates that they are used *relative* to something else, e.g., providing additional information about a modified noun, as in example (41) from the DTA sample. I will refer to the modified element as the antecedent. For attributive constituents, the base position is directly to the right of the antecedent, even though both positions (extraposed and adjacent to the antecedent) are equally grammatical, at least for attributive clauses.

(41) Erhebt es ſich [ANTEC mit dem Waſſerſtoffgas], [RELC welches die Moråſte in Ueberfluß ausdunſten?]

'Does it rise with the hydrogen gas that the swamps evaporate in abundance?'

As Zifonun et al. (1997) note, the relation to the antecedent is also the reason why attributive clauses are the clause type that is least prone to be located in the post-field. Since they can always be placed adjacent to their antecedent, they are the only clausal elements for which significant variability of their positioning in the middle field vs. post-field is observed.

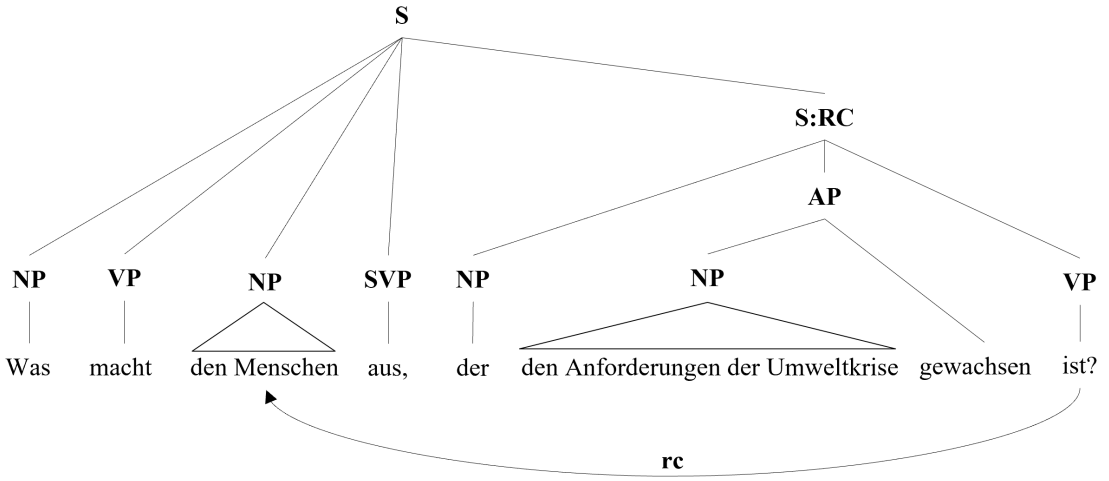
Similarly, attributive phrases like PPs may show interesting differences from independent phrases regarding their likelihood of extraposition (cf., Voigtmann and Speyer forthcoming). However, the overall frequency of extraposed attributive phrases is much too low for a reliable automatic identification or meaningful statistical analysis. Therefore, I will treat them like independent phrases and focus on attributive relative clauses for the remainder of this section.

In particular, I will explore how the antecedent and, hence, the base position of attributive relative clauses can be identified automatically in modern and historical German. I will focus on attributive relative clauses with a (pro-)nominal antecedent, assuming that the antecedent is a phrase of type NP or PP, as defined in Chapter 6.2. Other types of relative clauses (independent RelCs, continuous RelCs) are not considered here because they do not have an antecedent in the sentence, or their antecedent is the whole sentence, leaving only the end of the sentence (i.e., the post-field) as possible base position.

7.1.1. Data

To evaluate the recognition of antecedents, the same test sets as in Chapter 6.3 are used. For the Modern, HIPKON, and DTA samples, manually annotated relative clauses and antecedents are available. For the other three data sets, relative clauses are extracted from the constituency trees (cf. Chapter 6.3) and automatically linked to their antecedents. As already mentioned, only (pro-)nominal antecedents of attributive relative clauses are considered.

For the newspaper corpora, target antecedents are identified via dependency relations. For each RelC, the token linked to its (verbal) head via the respective relation `rc` or `relc` is selected.[73] The antecedent then corresponds to the token's parent in the constituency tree. Consider the following example of a simplified Tiger-style tree from the Tiger corpus:



'What makes the human who can meet the demands of the environmental crisis?'

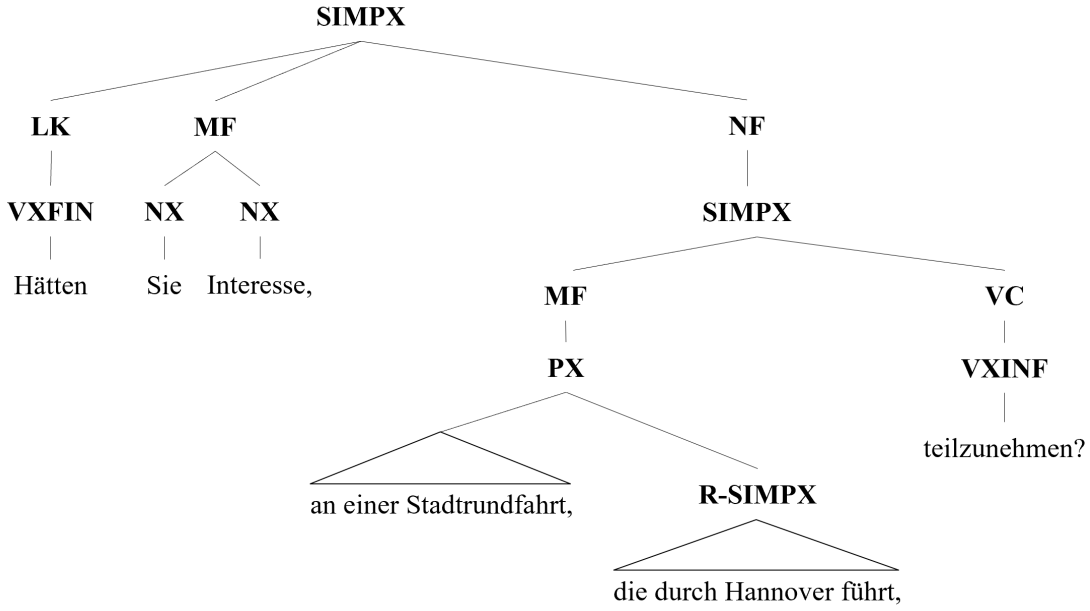
The relative clause `S:RC` from the constituency analysis is linked to the word *Menschen* via the `rc` dependency relation. The `NP` that dominates the token *Menschen* is thus selected as the antecedent of the RelC, with *Menschen* being the head of the antecedent.

For the Tiger corpus, the officially provided dependency annotations are used, whereas more accurate results are achieved for the TüBa-D/Z data with a modern neural parser.[74] The official (automatically generated) dependencies are only consulted if the parser does not find a suitable dependency head for the RelC.

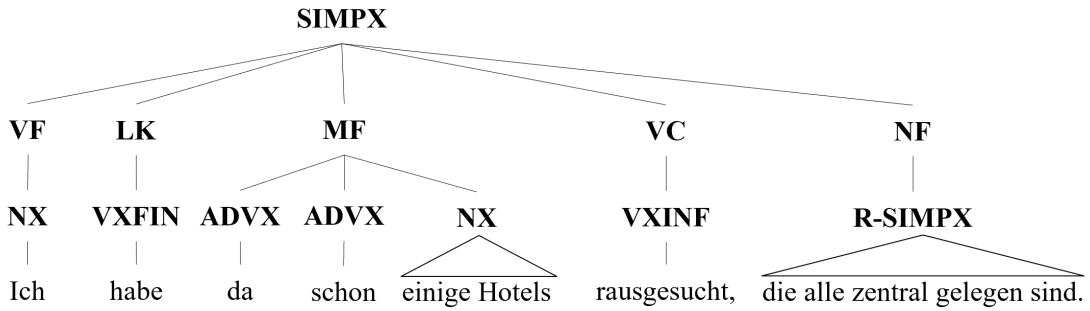
For the Spoken data, automatic dependency annotations are often not accurate enough. Instead, antecedents are identified solely based on the constituency analysis. Either the antecedent con-

[73] The dependency link can be established directly or indirectly in the case of coordinated relative clauses.

[74] I selected the fast and accurate spaCy parser (version 3.2.4, `https://spacy.io/`) with the German transformer model (`de_dep_news_trf-3.2.0`; labeled attachment score is stated as 95%) because it seems to be more accurate at identifying the dependency head of relative clauses than the officially provided dependencies in the development data.



'Would you be interested in participating in a city tour of Hannover?'



'I have already picked out a few hotels, which are located in the city center.'

Figure 7.1.: Simplified TüBa-style tree of a sentence with *in situ* RelC (top) and a sentence with extraposed RelC (bottom) from the Spoken data set.

Corpus	#Docs	#Sents	#Toks	#Words	#ExtrapCandidates	#Antecs
TüBa-D/Z	364	10,488	196,630	167,844	54,498	1,558
Tiger	149	522	15,075	12,986	540	540
Spoken	14	23,937	285,594	234,094	108,141	324
Modern	78	547	7,605	6,354	2,314	65
HIPKON	53	342	4,210	3,747	1,303	46
DTA	29	609	18,515	15,822	4,563	163

Table 7.1.: Overview of the test data. Only sentences containing at least one candidate for extraposition (phrase or attributive relative clause) are included in the evaluation. For the Tiger corpus, #ExtrapCandidates include only RelCs, because no gold standard topological fields are available. #Antecs refers to antecedents of attributive relative clauses.

sists of the preceding tokens dominated by the parent node in the constituency tree (e.g., *an einer Stadtrundfahrt* in Figure 7.1, top). Or, if the relative clause is not directly dominated by a phrase, the antecedent is considered to be the preceding `NP` or `PP` (*einige Hotels* in Figure 7.1, bottom). The head token of the constituent is also the head of the antecedent. Even though this is only an approximation, it seems accurate enough for the simple sentence structures of spoken German. Table 7.1 gives an overview of the test data that is used for the experiments.

7.1.2. Method

The automatic extraction of target antecedents from the gold data sets in the previous section already illustrates two possible ways to identify antecedents automatically. Using dependency relations as for the newspaper corpora appears to be the most natural way, given that antecedents can be located directly adjacent to the relative clause as well as (almost arbitrarily) far away from it.

However, while this might be a practical solution for modern (standard) German, models for dependency parsing of historical German do not exist yet. It may be possible to transfer a dependency parser from modern to historical data, similar to my experiments on other syntactic annotations in Chapters 5 and 6 (cf. Krielke et al. 2022). But this approach likely requires POS-based models to compensate for the different, unstandardized word forms in historical language. Or for dependency parsers like the spaCy parser[75] that operate on word forms and not POS tags, the historical data would have to be modernized before applying the parser. And even then, it remains an open question of how much the different sentence structures in historical data would affect parser accuracy (cf. the observations in the previous chapters and Krielke et al. 2022).

As a consequence, the exploitation of dependency relations seems impractical for identifying antecedents, at least within the scope of this thesis. Instead, I decided to use very simple heuristics that require only the existing annotations from the previous chapters. Assuming that the antecedent and relative clause are either directly adjacent to each other or separated only by the right sentence

[75] `https://spacy.io/`

bracket in most cases (cf., e.g., Zifonun et al. 1997; Uszkoreit et al. 1998), it may be sufficient to select the phrase closest to the left of the relative clause that could qualify as antecedent.

Given an identified RelC, its parent node in the constituency tree is determined (e.g., the NP in example (42) from the DTA, or the prepositional phrase in Figure 7.1, top). If the parent node is a noun or prepositional phrase, the dominated tokens that precede the RelC are taken to form the antecedent (*Die Flasche* in (42), and *an einer Stadtrundfahrt* in Figure 7.1, top).

(42) [NP Die Flaſche, [RELC welche dieſe Miſchung enthielt]], war klar und durchſichtig;
[Antec Die Flaſche], [RELC welche dieſe Miſchung enthielt], war klar und durchſichtig;
'The bottle that contained this mixture was clear and transparent.'

If the relative clause is not dominated by a phrase, i.e., RelC and antecedent are not adjacent, the NP or PP closest to the left in the constituency tree is considered to be the antecedent (e.g., *den Steten* in example (43) from the HIPKON data, or *einige Hotels* in Figure 7.1, bottom).

(43) Es gehe dem ſelben menſchen / wie es [Antec den Steten] ergangen iſt / [RELC die der HERR one barmhertzigkeit vmbgekert hat.]
'May that person be treated like the cities that the Lord converted without mercy.'

If this antecedent is dominated by another relative clause, the RelCs are considered coordinated and share the antecedent of the left-most RelC, as in example (44) from the HIPKON data set.

(44) vñ ſu̇len v̇ns ſchamē [Antec$_{1+2}$ etlicher zimlich' díngē] [RELC$_1$ du̇ nít verbottē ſínt] /
vñ [RELC$_2$ dc mā wol tete].
'And we should be ashamed of some decent things that are not forbidden and that one would do.'

For each antecedent, in addition to the span, the head token is determined based on the hierarchical structure of the phrase and POS tags. The algorithm first checks the tokens that are directly dominated by the phrase. If there are no such tokens, e.g., due to coordination, the token children of the last dominated child phrase are checked. Possible head tokens are then filtered by their POS tags. The following POS tags are considered (in the given order):

1. Nouns (NN, TRUNC)
2. Names (NE)
3. Pronouns (substituting pronouns only)
4. Numbers (CARD)
5. Adjectives (ADJA)[76]

[76] Nominalized adjectives are tagged as ADJA like normal attributive adjectives according to the STTS (Schiller et al. 1999). So adjectives are allowed as heads if the antecedent does not include a noun.

6. Foreign words (FM)

The first category with a matching token is selected. In example (44), the antecedent phrase contains two adjectives (*etlicher* and *zimlich'*) and one noun (*díngē*). So the first match would return the head noun *díngē*. If there is more than one matching token, e.g., several names or a noun with post-nominal modification, the last head candidate is chosen because it is closest to the relative clause.

In the case of coordinations, the RelC could refer to one or all of the conjuncts, cf. example (45a) vs. (45b). This difference cannot be distinguished without a morphological analysis, and even then may remain ambiguous because relative pronouns are identical for plural and feminine singular referents (cf. example 46). While it depends on the constituency analysis whether one or all of the conjuncts are analyzed as antecedent, I always select the last possible candidate as the head token (underlined in the examples).

(45) a. [$_{\text{Antec}}$ Die Frau und der Mann], die ...

b. Die Frau und [$_{\text{Antec}}$ der Mann], der ...
'The woman and the man who ...'

(46) a. [$_{\text{Antec}}$ Die Frau und die Kinder], die ...

b. Die Frau und [$_{\text{Antec}}$ die Kinder], die ...
'The woman and the children who ...'

7.1.3. Evaluation and Results

For evaluation, the automatically identified antecedents are compared span-wise to the target annotation. Only sentences that contain a candidate for extraposition are included in the evaluation, and punctuation is ignored. Despite these similarities, the exact evaluation procedure for antecedents differs from other evaluations in this thesis. In particular:

(i) Antecedents are not labeled, so there are no `LE` and `LBE` errors (comparable to the RelC annotation in Chapter 6.3).

(ii) Antecedents must always be evaluated with respect to the corresponding relative clause – if the RelC is missing, the antecedent will also be missing. And if the antecedent is linked to the wrong RelC, it is incorrect independent of its position.

(iii) The right boundary of the antecedent is much more relevant than the span itself (at least in the context of this thesis) because a fuzzy match with correct right boundary is sufficient for determining the base position of the RelC and the distance between both.

(iv) Annotation accuracy likely depends on the distance to the relative clause.

To account for these differences, the error types from Chapter 4 are re-defined for the evaluation of antecedents and complemented with two newly created categories:

TP	Antecedent with correct boundaries, linked to correct RelC
$\texttt{BE}_\text{s}$, $\texttt{BE}_l$, $\texttt{BE}_\text{o}$	Boundary error of the respective type, linked to correct RelC
$\texttt{BE}_{right}$	Error of type $\texttt{BE}_\text{s}$, $\texttt{BE}_l$, or $\texttt{BE}_\text{o}$ with correct right boundary
IL	Antecedent of correct RelC but in an incorrect location (not overlapping with the target antecedent)
FP	Antecedent only in system annotation, e.g., caused by false positive RelC
FN	Antecedent only in target annotation, e.g., caused by false negative RelC

Error types are counted for each distance and overall, with distance being measured as the number of intervening tokens between antecedent and RelC (ignoring punctuation). For the calculation of precision, recall, and F_1-score, the new error categories are treated like boundary errors (Eq. 7.1).

$$1BE_\text{s} = 1BE_l = 1BE_\text{o} = 1BE_{right} = 1IL = 0.5FP + 0.5FN \tag{7.1}$$

In addition, a weighted score F_{1right} is calculated, for which errors of type $\texttt{BE}_{right}$ are counted as true positives (Eq. 7.2).

$$1BE_{right} = 1TP \tag{7.2}$$

Table 7.2 shows the results for antecedent identification with the four parser models from Chapter 6.2. For the modern newspaper data, the simple heuristics reach F_1-scores of 78% and 79%, respectively. The modern spoken and non-newspaper data sets are analyzed with even higher F_1-scores >83%. The models that achieved the best results for RelC recognition in modern data in Chapter 6.3 also perform best for the identification of antecedents in the same data set. For the historical data, the HIPKON corpus, once again, shows higher results than the DTA, with an F_1-score of 83% vs. about 69%.

Interestingly, most of the errors are boundary errors (36%–79%, cf. Table 7.3). Often, the system antecedents are longer than the target annotation. But in many of these cases, the errors only concern the left boundary, while the (more important) right boundary is correct (28%–74%, see also Figure 7.2). That can happen, e.g., when the antecedent is the embedded post-modifier and not the complete NP, as in example (47) from the DTA. Even though the system annotation is technically incorrect in this case, the analysis is accurate enough for the identification of extraposition and the RelC's base position. When the (irrelevant) $\texttt{BE}_{right}$ errors are counted as true positives, F_{1right} lies between 82% and 96%, which is stunningly high considering the simplicity of the identification approach.

Model		TüBa-D/Z	Tiger	Spoken	Modern	HIPKON	DTA
News1	Prec	**77.64**	71.21	**86.01**	**83.19**	75.00	**72.29**
	Rec	**80.65**	75.46	**86.01**	**83.19**	58.54	64.98
	F_1	**79.12**	73.27	**86.01**	**83.19**	65.75	68.44
	$F_{1\,right}$	**92.61**	**88.10**	**95.13**	**96.06**	77.50	81.91
News2	Prec	75.92	**76.39**	80.95	79.63	69.23	69.88
	Rec	78.08	**80.09**	79.21	78.18	45.00	64.68
	F_1	76.98	**78.20**	80.07	78.90	54.55	67.18
	$F_{1\,right}$	91.43	88.06	94.42	91.80	73.68	**82.59**
Hist	Prec	-	-	-	-	82.93	67.69
	Rec	-	-	-	-	80.95	65.19
	F_1	-	-	-	-	81.93	66.42
	$F_{1\,right}$	-	-	-	-	**89.89**	81.61
Mix	Prec	-	-	-	-	**83.33**	69.14
	Rec	-	-	-	-	**83.33**	**68.63**
	F_1	-	-	-	-	**83.33**	**68.89**
	$F_{1\,right}$	-	-	-	-	**89.89**	82.00

Table 7.2.: Overall precision, recall, F_1-score, and F_1 with correct right boundary (in percent) according to *FairEval* for antecedent recognition with the different models on each data set. Models trained on historical data are only applied to the historical test sets, and the highest scores for each corpus are highlighted in bold.

Corpus	FP	BE					IL	FN
		BE_s	BE_l	BE_o	BE_{right}	BE_{all}		
TüBa-D/Z	11.25	25.54	43.57	1.07	60.18	70.18	16.43	2.14
Tiger	12.08	18.75	42.50	1.67	49.58	62.92	23.75	1.25
Spoken	8.54	28.05	43.90	2.44	62.20	74.39	8.54	8.54
Modern	5.26	26.32	47.37	5.26	73.68	78.95	10.53	5.26
HIPKON	12.00	12.00	24.00	0.00	28.00	36.00	4.00	48.00
DTA	9.64	15.66	30.12	1.20	36.14	46.99	16.87	26.51

Table 7.3.: Proportion of the different error types: false positives (FP), shorter, longer, and overlapping boundary errors (BE_s, BE_l, BE_o), boundary errors with correct right boundary and boundary errors in general (BE_{right}, BE_{all}), antecedents in an incorrect location (IL), and false negatives (FN). BE_{right} errors include errors of types BE_s and BE_l. Numbers are given in percent for the best model on each data set.

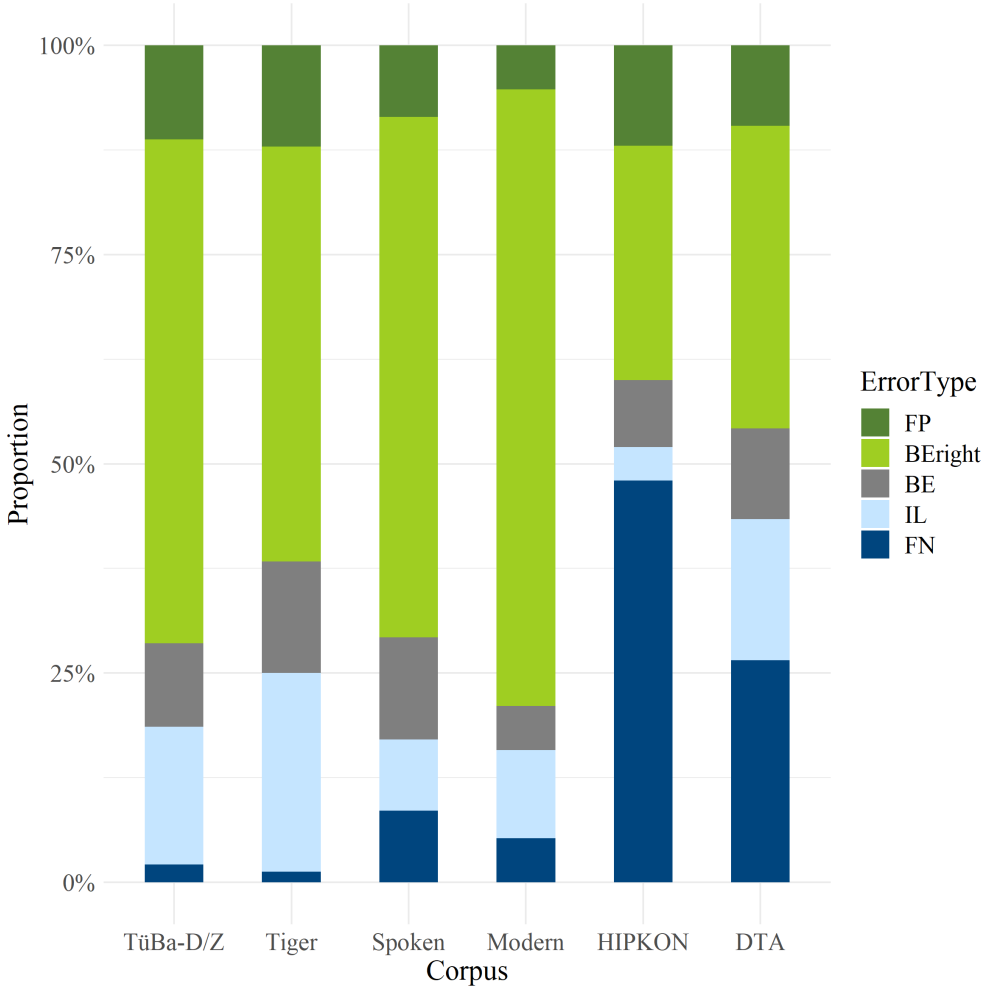


Figure 7.2.: Proportion of the different error types for antecedent identification: false positives (FP), boundary errors with correct right boundary (BE_{right}) and without correct right boundary (BE), antecedents in an incorrect location (IL), and false negatives (FN). Numbers are shown for the best model on each data set.

(47) Euer Königliche Majeſtät genießen das göttliche Vergnügen, [$_{\texttt{NP}}$ eine Freundin und Kennerin [$_{\texttt{NP}}$ der ſchönen Natur]] zu ſeyn, [$_{\texttt{RELC}}$ in deren Tempel Allerhöchſtdieſelben ſo gerne des Glanzes Allerhöchſtdero Thrones vergeſſen] [...]

Target: ... eine Freundin und Kennerin [$_{\texttt{Antec}}$ der ſchönen Natur] ...

System: ... [$_{\texttt{Antec}}$ eine Freundin und Kennerin der ſchönen Natur] ...

'Your Royal Majesty enjoys the divine pleasure of being a friend and connoisseur of beautiful nature in whose temple you so gladly forget the splendor of your throne.'

False negatives, which mainly result from errors in the RelC annotation, are rare in the modern data sets. In the historical corpora, which also show lower accuracies for RelC identification in Chapter 6.3, `FNs` are more frequent. The main reason for false positives in the modern data are non-attributive relative clauses, which are analyzed as attributive RelCs by the simple heuristics.

`IL` errors, i.e., antecedents linked to the correct RelC but located in the wrong place, are most frequent in the news corpora and the complex DTA texts and occur especially for longer distances, which is a logical consequence of the applied heuristics. If another constituent is placed before the relative clause that could qualify as antecedent according to the simple heuristics (e.g., the `PP` in example (48) from the Tiger corpus), more elaborate analyses (e.g., including agreement checking) would be necessary to identify the correct antecedent.

(48) Bündnis 90/Die Grünen wollen im kommenden Frühjahr [$_{\texttt{Antec}}$ ein Einwanderungsgesetz] [$_{\texttt{PP}}$ in den Bundestag] einbringen, [$_{\texttt{RELC}}$ das die Aufnahme von Einwanderern mit Quoten regelt.]

'Bündnis 90/Die Grünen want to introduce an immigration law in the federal parliament next spring that would regulate the admission of immigrants with quotas.'

Cases like this are infrequent, though. The evaluation confirms that almost all relative clauses are placed within a distance of one or two words from the antecedent, in accordance with the observations by Uszkoreit et al. (1998). Greater distances mainly result from coordinated RelCs, which are explicitly covered by the heuristics. So, in general, the developed method seems sufficiently accurate for the identification of antecedents in the context of this thesis.

For completeness, Table 7.4 shows traditional precision, recall, and F_1-scores for antecedent heads, even though they are not relevant for the purpose of identifying the base position. Since the head cannot be correct if the antecedent is missing or annotated at the wrong position, the table includes the results for all antecedents and for antecedents with correct (right) boundaries only. The results show that the head token is identified with high accuracy if at least the right boundary of the antecedent is recognized correctly, speaking in favor of always selecting the right-most possible head. F_1-scores range between 80% and 96.5% for antecedents with correct right boundary and over 95% to 100% for completely correct antecedent spans.

	All			Right			Correct		
Corpus	Prec	Rec	F_1	Prec	Rec	F_1	Prec	Rec	F_1
TüBa-D/Z	72.81	73.30	72.32	80.55	79.83	81.28	96.71	95.66	97.78
Tiger	66.97	67.78	66.18	79.01	78.12	79.91	97.55	96.96	98.15
Spoken	87.99	86.24	89.81	92.69	92.08	93.31	98.81	98.81	98.81
Modern	80.62	80.00	81.25	84.30	83.61	85.00	95.74	95.74	95.74
HIPKON	74.07	65.22	85.71	93.33	90.32	96.55	95.65	91.67	100.00
DTA	70.36	65.45	76.06	89.36	87.50	91.30	96.09	95.56	96.63

Table 7.4.: Precision, recall, and F_1-scores (in percent) according to traditional evaluation for antecedent head identification on each data set with the model with the highest $F_{1_{right}}$-score from Table 7.2. Results are given for all antecedents (`All`), for antecedents that are linked to the correct RelC and have (at least) a correct right boundary (`Right`), and for antecedents that are linked to the correct RelC and have two correct boundaries (`Correct`).

7.2. Identification of Extraposition

Given the developed methods from Chapters 5–6 and the identification of antecedents from Section 7.1, the position of extraposition candidates can be determined. Two basic positions are distinguished:

`insitu` The constituent is placed in its base position.

`extrap` The constituent is extraposed, i.e., it has been 'moved' to the post-field.

In addition to these options, a third case can occur when the right sentence bracket is empty. As explained in Chapter 5, the boundary between the middle field and post-field must not always be marked explicitly. The respective elements that are located at this boundary could thus belong to the middle field (`insitu`) or the post-field (`extrap`). In this thesis, I adopt a conservative perspective similar to Telljohann et al. (2017) and only consider phrases as part of the post-field if they are placed behind an explicit right bracket. If the right bracket is empty, phrases are analyzed as part of the middle field and labeled as `insitu`, although this may not necessarily be true, e.g., in historical or spoken data.

For attributive constituents like relative clauses, the position depends not only on the topological field but also on the location of their antecedent. Here, RelCs are labeled as `extrap` if they are unambiguously separated from their antecedent, either by the sentence bracket or other words, even if they are not placed in the post-field. For RelCs that are adjacent to their antecedent, the label depends on the position relative to the topological fields. If the antecedent and RelC are both placed in the pre-field, in the middle field before other constituents and/or a right bracket, or in the post-field behind a right bracket, the RelC is considered as `insitu`. If the antecedent and RelC are located at the end of the middle field with an empty right bracket, the position cannot be determined

unambiguously. While Telljohann et al. (2017) would annotate the RelC as part of the post-field, I will use a third category `ambig` to distinguish them from clearly extraposed cases. In studies on relative clause extraposition, ambiguous cases are usually discarded (e.g., Sahel 2015; Uszkoreit et al. 1998).

`ambig` Only for RelCs: The right sentence bracket is empty, and the RelC is located adjacent to its antecedent at the end of the middle field or beginning of the post-field.

7.2.1. Data

For evaluation, the same test data as in the previous section (Table 7.1) is used, with five data sets providing gold annotations of all extraposition candidates. The Tiger corpus does not contain a topological field analysis, so only the position of relative clauses can be reliably extracted from the existing annotations (complemented by an automatic topological field analysis in ambiguous cases). Figure 7.3 shows the distribution of positions by extraposition candidate in the different data sets.

For phrases, extraposition is the exception rather than the rule. Except for the HIPKON data, which only includes sentences with at least one extraposed element, almost all phrases are left *in situ*. In modern German, `NPs` and `APs` are extraposed least often, whereas `ADVPs` are the least frequently extraposed phrase type in historical German. `PPs` are the phrase type with the highest proportion of extraposition in all data sets (3%–10%, HIPKON: 70%). As expected, phrasal extraposition is more frequent in non-standard language than in newspaper text and even more so in spoken than written German.

For attributive relative clauses, extraposition is considered grammatical in German, which is reflected in 23%–31% extraposed RelCs in the modern data sets. In the historical corpora, RelCs occur adjacent to their antecedent more often (56%–95%). Ambiguous cases with empty right brackets are less frequent in the historical samples ($\leq$15%), while they account for 30%–55% of the cases in modern German.

7.2.2. Method

To determine the position of constituents as `insitu`, `extrap`, or `ambig`, phrases and relative clauses are identified as described in Chapter 6 and located within the topological field analysis from Chapter 5. The exact procedure differs slightly for phrases and RelCs, especially regarding the number of positions (2 vs. 3) and the role of antecedents (Section 7.1).

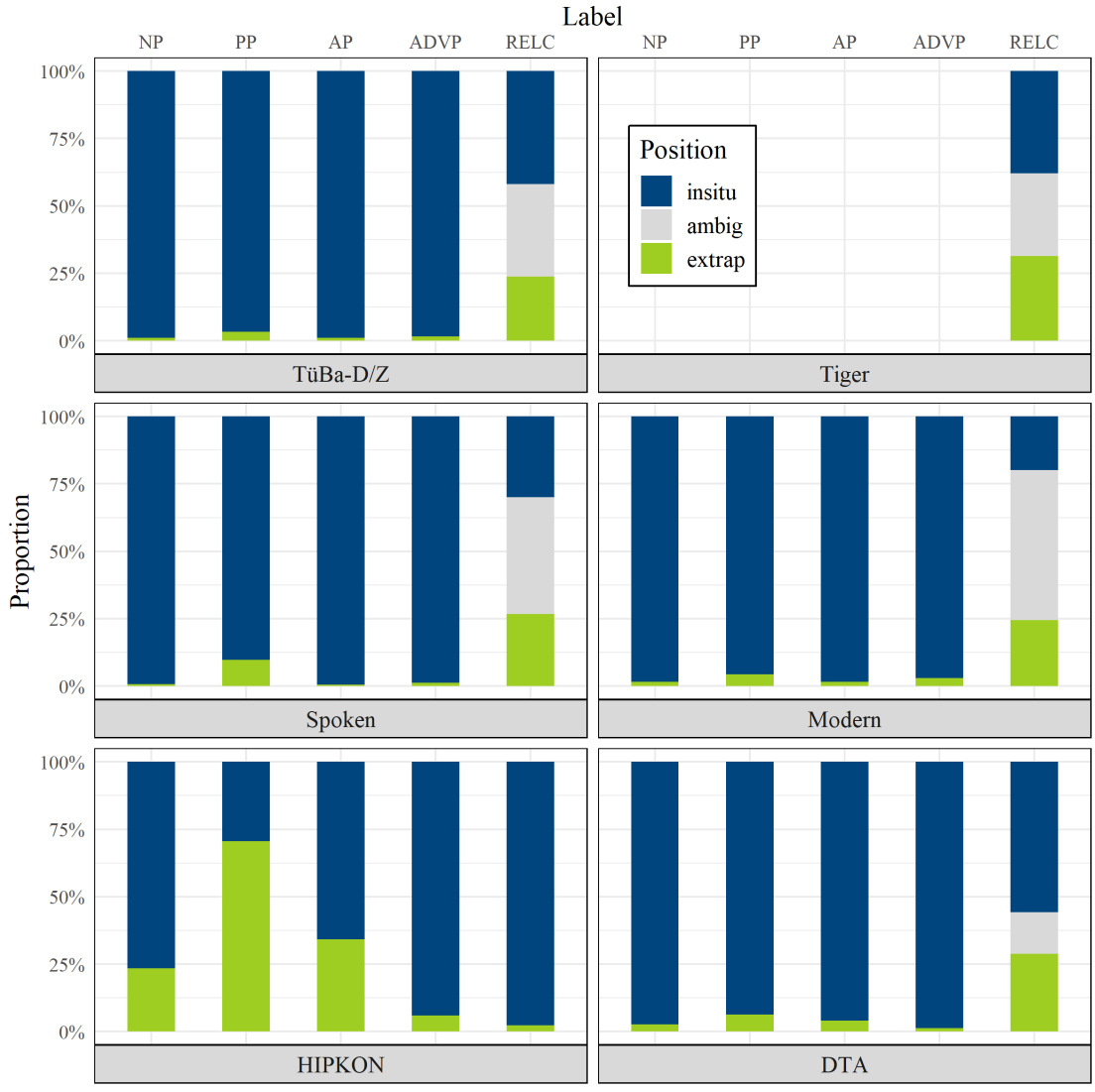


Figure 7.3.: Distribution of positions for the different extraposition candidates in the test data sets.

Position of Phrases

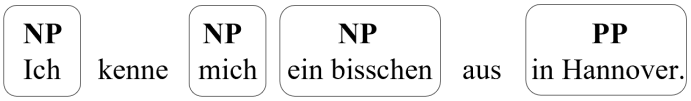
For phrases, the position is determined based on the constituency tree and the topological field analysis. Theoretically, it should be possible to merge the two trees (comparable to the TüBa-style annotation by Telljohann et al. 2017) and read off the position directly from the combined tree. However, automatically generated parses likely contain errors, which hinder the intersection of trees and cause inconsistent results. Instead, I only combine the relevant parts of the two syntactic analyses to prevent the propagation of errors as much as possible.

Step 1 Phrases of the four types `NP`, `PP`, `AP`, and `ADVP` are identified with the Berkeley parser and each of the constituency models `News1`, `News2`, `Hist`, and `Mix`. Contrary to the procedure in Chapter 6.2, the internal structure of phrases is retained because they may include other (potentially extraposed) phrases that should also be recognized, as in example (49) from the TüBa-D/Z data.

(49) Würdest du [$_{\texttt{NP-insitu}}$ den Mönchen, die jeden Tag meditieren [$_{\texttt{PP-extrap}}$ hinter ihren Mauern]], auch sagen, sie hätten einen Knastkoller?

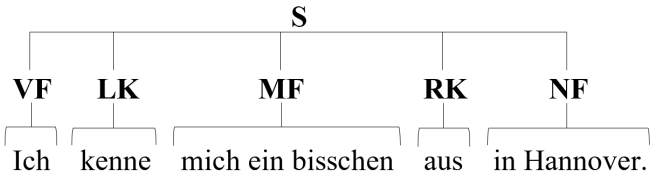
'Would you also tell the monks who meditate every day behind their walls that they suffer from prison-induced madness?'

The result of Step 1 for an example sentence from the Spoken data set could look as follows:



'I know Hannover a little bit.'

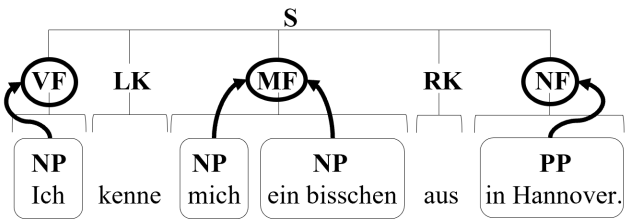
Step 2 To identify post-fields, a topological field analysis of the sentence is carried out with the `Punct` model from Chapter 5. The `News1` model achieves higher scores for two of the modern corpora, but the pure POS-based topological field model seems to generalize better to other language varieties and historical data. Therefore, the same model is used for all data sets. The following could be an analysis of the example sentence:



Step 3 For each phrase from Step 1, the corresponding topological field from Step 2 must be identified. The respective field is the lowest node in the topological field tree that dominates all tokens of the phrase (ignoring punctuation). If no matching field is found, this likely indicates a boundary error, either for the phrase or the field. In this case, the lowest node from the topological field tree that dominates the first token of the phrase is selected (again, ignoring punctuation). Experiments with the development data suggest that this makes the analysis more robust against incorrect boundaries, as illustrated by example (50) from the HIPKON data, where the automatically recognized post-field `NF` is too short and overlaps only with the first part of the `PP`.

(50) alſo ſtât h're Dauid aínes tages v̂f [$_{\texttt{NF}}$ nach mittē tage] do er hat geſclâfen.
alſo ſtât h're Dauid aínes tages v̂f [$_{\texttt{PP}}$ nach mittē tage do er hat geſclâfen].
'So one day, Shepherd David gets up after the middle of the day, where he had slept.'

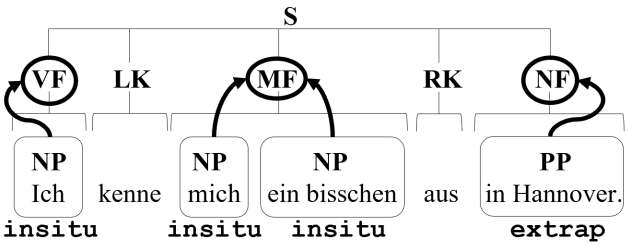
For the example sentence, the mapping of phrases to fields would be as follows:



Step 4 Finally, the position of each phrase is determined based on the selected topological field from Step 3. If the phrase is (directly) dominated by a post-field, it is extraposed. Otherwise, it is labeled as `insitu`. Embedded phrases from Step 1 are retained only if the topological field is located between the phrase and its parent, as in example (51), i.e., the parent phrase (`NP`) dominates the tokens of the field (`NF`), and the field (`NF`) dominates the tokens of the embedded phrase (`PP`).

(51) [$_{\texttt{NP}}$ den Mönchen, die jeden Tag meditieren [$_{\texttt{NF}}$ [$_{\texttt{PP}}$ hinter ihren Mauern]]]

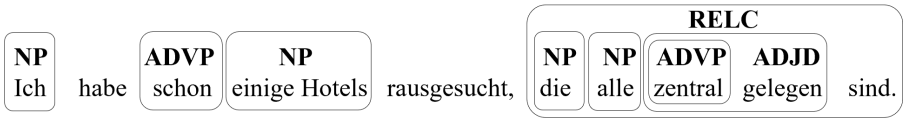
The phrases in the example sentence would be labeled like this:



Position of Relative Clauses

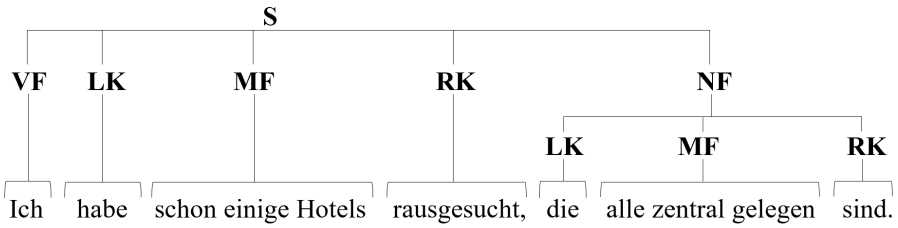
For relative clauses, the position depends not only on the topological field but also on the position of the antecedent. If the RelC is adjacent to its antecedent, by definition, it cannot be extraposed – even if both elements are located in the post-field. And if the RelC is adjacent to its antecedent, but there could be an empty right sentence bracket between them, this should be recognized as an ambiguous case. Consequently, the necessary steps for locating RelCs differ from those for phrases.

Step 1 Relative clauses are identified in the data with the Berkeley parser and one of the four models, as described in Chapter 6.3. The result for a (shortened) example sentence from the Spoken data set could look as follows (including the phrase analysis needed for antecedent identification):

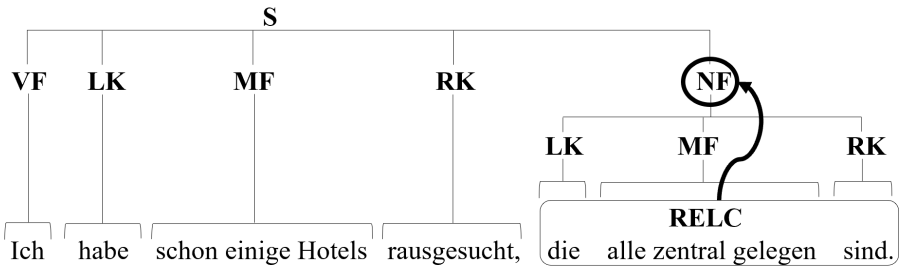


'I have already picked out a few hotels, which are located in the city center.

Step 2 A topological field analysis of the sentence is carried out with the `Punct` model from Chapter 5 (cf. above). The analysis of the example sentence would be as follows:

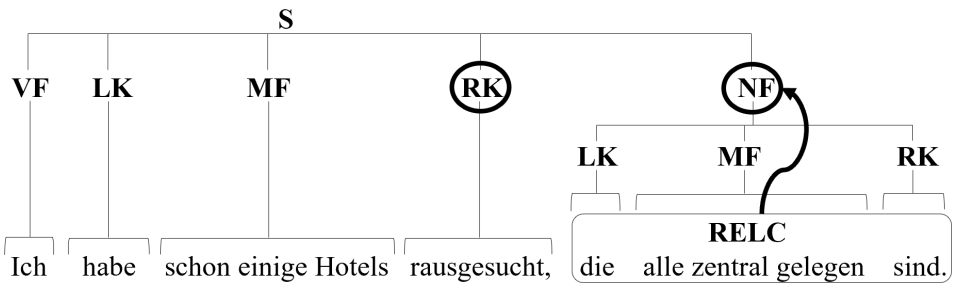


Step 3 For each RelC, the corresponding topological field is selected as the lowest node in the topological field tree that dominates all tokens of the clause (ignoring punctuation). In the example, this would be the post-field `NF`:

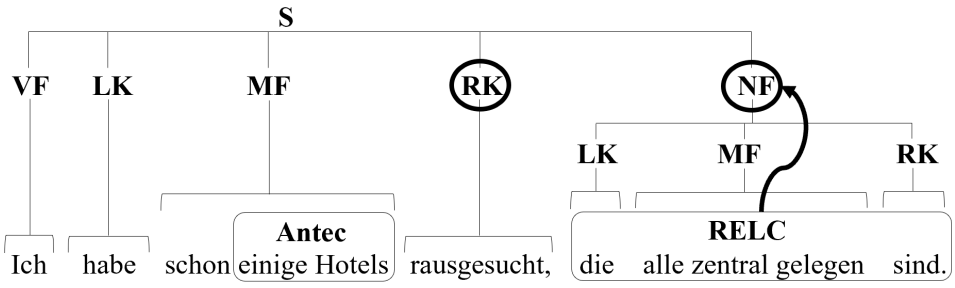


If no matching field is found, this likely indicates a boundary error, either for the RelC or the field. In this case, the lowest node from the topological field tree that dominates the first token of the RelC is identified (ignoring punctuation). Contrary to Step 3 for the (usually) non-complex phrases above, here, this field corresponds to the left sentence bracket (`LK`), and its parent field (e.g., `NF`) should in turn be the desired parent field of the relative clause.

Step 4 To distinguish between in-situ and ambiguous cases, the preceding topological field is identified. If the field from Step 3 is a post-field, the preceding field is usually a right sentence bracket or a middle field. In the example, it is the right bracket `RK`:

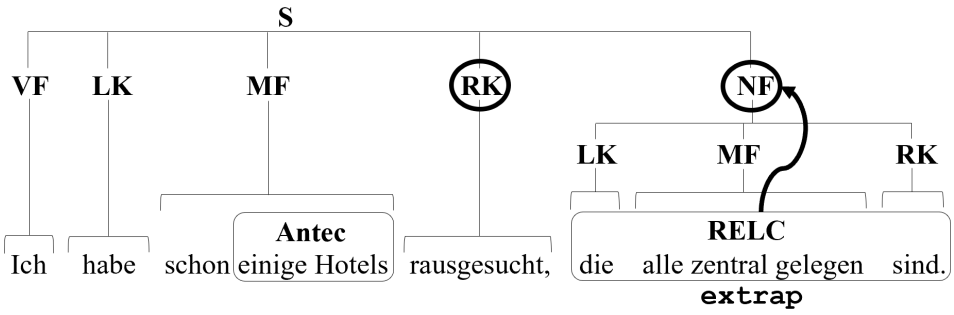


Step 5 Since the position of relative clauses eventually depends on the antecedent, the antecedent of each RelC is identified as described in Section 7.1. In the example, this is the `NP` *einige Hotels* ('some hotels') from Step 1:



Step 6 Based on the information from Steps 1–5, the position of RelCs can be determined.

(i) If the RelC is separated from its antecedent by one or more tokens (ignoring punctuation) and/or the RelC is located in the post-field behind a right sentence bracket, it is labeled as `extrap`. This is the case for the example:



(ii) If the RelC is located in the post-field, adjacent to the antecedent in the middle field, with an empty `RK` between them, the RelC is `ambig`.

(iii) Otherwise, if the RelC is adjacent to the antecedent and both are located in the post-field (possibly behind a right sentence bracket) or in another field, the RelC is labeled as `insitu`.

(iv) Coordinated RelCs share their antecedent with a preceding RelC (cf. Section 7.1) and are assigned the same position as the left-most RelC. For example, the second RelC in example (51), repeated from (44), is labeled as `insitu`, despite being separated from the antecedent by the first RelC and the conjunction.

(51) vñ ſu̍len v̍ns ſchamē [$_{\texttt{Antec}_{1+2}}$ etlicher zimlich' díngē] [$_{\texttt{RELC}_1\texttt{-insitu}}$ dū nít verbottē ſínt] / vñ [$_{\texttt{RELC}_2\texttt{-insitu}}$ dc mā wol tete].
'And we should be ashamed of some decent things that are not forbidden and that one would do.'

7.2.3. Evaluation and Results

After the position of the constituents has been determined as described in the previous section, their labels (`NP`, `RELC`, etc.) and positions (`insitu`, `extrap`, `ambig`) are concatenated (e.g., `NP-insitu` or `RELC-ambig`). This enables a standard evaluation of labeled spans, as detailed in Chapter 4. It also means that all extraposition candidates and all positions are included in the same evaluation because confusions can occur between labels, positions, or both. Only sentences with at least one candidate for extraposition are considered, and punctuation is ignored. Table 7.5 shows the results of fair evaluation.

For modern German, F_1-scores range between 86% for the Modern data set to 92% on newspaper language. On the historical data sets, the highest scores are reached with 83.5% on the HIPKON data and 74% on the DTA. Errors mainly concern incorrect boundaries (cf. Figure 7.4). The high proportion of `FPs` in the Tiger corpus compared to the other data sets originates from the fact that only RelCs can be evaluated in this corpus, leading to a small number of only 84 errors in total.

However, while the overall results suggest high accuracy for the annotation of extraposition, this is not entirely true. Table 7.6 shows fair F_1-scores for each label, differentiated by position. For the

	News1			News2			Hist			Mix		
Corpus	Prec	Rec	F_1	Prec	Rec	F_1	Prec	Rec	F_1	Prec	Rec	F_1
TüBa-D/Z	**91.12**	**91.91**	**91.52**	84.66	83.49	84.07	-	-	-	-	-	-
Tiger	87.84	92.37	90.05	**89.93**	**94.14**	**91.98**	-	-	-	-	-	-
Spoken	**88.85**	**90.21**	**89.53**	82.43	83.73	83.08	-	-	-	-	-	-
Modern	**86.09**	**86.56**	**86.32**	84.54	83.61	84.07	-	-	-	-	-	-
HIPKON	73.10	74.12	73.61	75.03	74.46	74.75	**84.21**	82.88	**83.54**	83.59	**82.97**	83.28
DTA	73.39	**74.94**	74.16	72.16	70.23	71.18	73.07	70.43	71.73	**74.80**	73.78	**74.29**

Table 7.5.: Overall precision, recall, and F_1-scores (in percent) according to *FairEval* for the extraposition analysis with different models on each data set. Models trained on historical data are only applied to the historical test sets, and the highest scores for each corpus are highlighted in bold. Traditional evaluation results can be found in Table A.12 in the appendix.

	NP		PP		AP		ADVP		RELC		
Corpus	*in situ*	extrap	*in situ*	extrap	*in situ*	extrap	*in situ*	extrap	*in situ*	ambig	extrap
TüBa-D/Z	92.16	42.50	89.53	61.97	91.49	24.14	95.53	55.05	91.43	89.93	86.96
Tiger	-	-	-	-	-	-	-	-	95.24	92.01	88.10
Spoken	91.68	41.82	77.36	63.84	85.59	16.67	93.27	39.15	76.92	83.40	87.18
Modern	88.25	40.00	85.14	43.75	84.94	50.00	85.57	40.00	96.00	89.23	78.57
HIPKON	90.22	58.52	78.31	82.41	71.43	55.56	89.51	44.44	85.00	0.00	0.00
DTA	74.50	41.56	76.18	48.82	77.78	16.67	76.15	40.00	67.14	68.42	69.05

Table 7.6.: Overall F_1-scores for each label and position (in percent) according to *FairEval* for extraposition analysis with the best performing model on each data set.

phrases, it turns out that scores are high only for the (much more frequent) *in situ* variants, whereas the (rare) cases of extraposition are recognized much less reliably. The highest scores are achieved for extraposed `PPs`, with 43%–82%. For the other phrases, F_1-scores are $\leq$58%. In general, phrasal extraposition is recognized best in the HIPKON data set with its rather simple sentences but high proportions of extraposition.

For relative clauses, the picture is less dismal. For most data sets, F_1-scores are comparable between the different positions. *In situ* RelCs are the most accurate group (except for the Spoken data) with 67%–96%, followed by ambiguous and extraposed clauses. The latter group is identified with F_1-scores >78% in modern German and 69% in the DTA. The single extraposed RelC in the HIPKON data is labeled as `ambig`, while ambiguous RelCs do not exist in the HIPKON sample. In the other data sets, F_1-scores lie between 68% and 92% for the ambiguous cases.

Figure 7.5 shows the confusion of labels and positions. For all data sets, a clear diagonal can be observed, i.e., errors mainly concern incorrect boundaries (32%–68%). `LBE` and `LE` errors are the second and third most frequent errors in most data sets. For extraposed elements, they result especially from confusions of position (`insitu` instead of `extrap`), whereas the confusion of

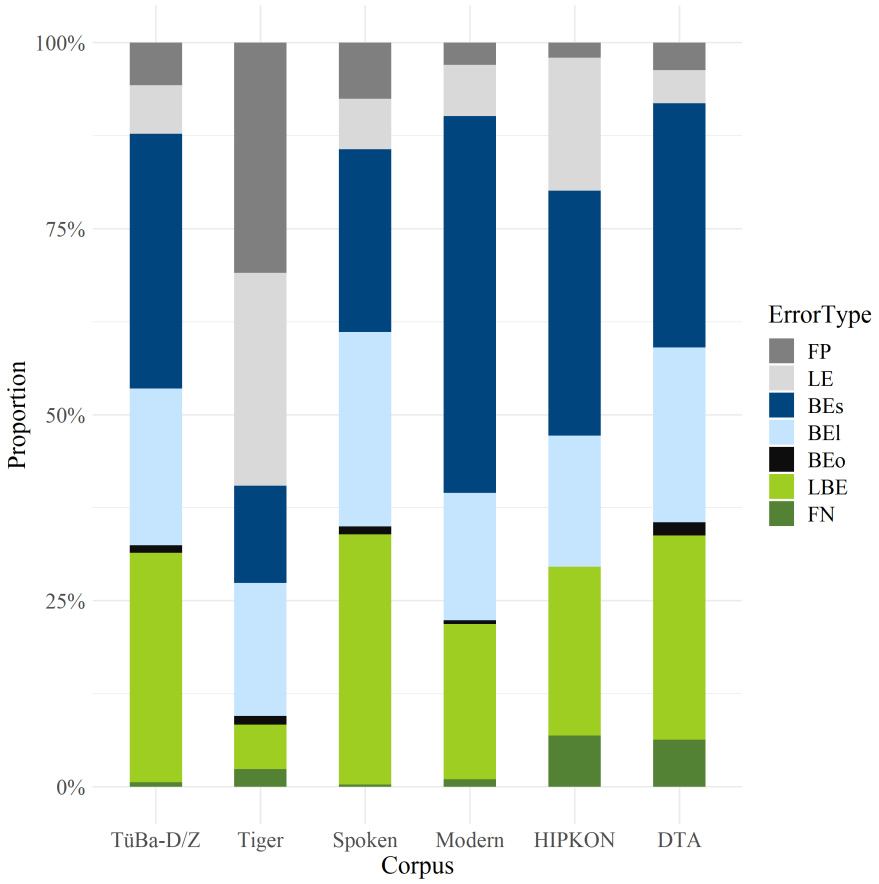


Figure 7.4.: Proportion of the different error types for extraposition analysis: false positives (FP), labeling errors (LE), shorter, longer, and overlapping boundary errors (BE_s, BE_l, BE_o), labeling-boundary errors (LBE), and false negatives (FN). Numbers are shown for the best model on each data set.

labels is more frequent for *in situ* phrases. Confusions of phrases with relative clauses or vice versa are almost non-existent and might be only an artifact of the evaluation algorithm.[77]

The better results for RelCs compared to phrases can be explained by several factors. RelCs are generally recognized with higher accuracy than phrases (cf. Chapter 6.3) due to their distinctive structure. Also, identifying the position of RelCs depends less on the post-field, which is recognized less reliably than other fields. Instead, the antecedent (and in particular its accurate right boundary; cf. Section 7.1) plays a dominant role for RelC position. Also, post-fields with internal structure, e.g., containing a relative clause, are generally recognized more reliably than post-fields

[77] Consider the discussion about error alignment in multi-level annotations in Chapter 4.

without such an internal structure, e.g., containing only a single extraposed phrase (cf. Chapter 5). Although definitive statements about error causes are difficult without a detailed qualitative analysis, the low recall of uncomplex post-fields seems to be the main bottleneck for the recognition of phrasal extraposition.

7.3. Corpus of Variants

The results from the previous section can be used for an automatic analysis of extraposition with a corpus of variants. As explained in Section 7.1, the definition of extraposition entails that the constituents are placed in the post-field instead of their base position in the middle field (or pre-field). If the extraposed elements were moved back to the base position, this would allow to compare the two positions and directly inspect possible reasons for and effects of extraposition. In our project, we termed this approach the 'corpus of variants' method because moving the elements creates an artificial variant of the sentence in which only specific linguistic aspects are manipulated while other factors are kept constant.

In Ortmann et al. (2022), we applied the method to the order of direct and indirect objects in the German middle field, comparing information profiles of the original sentences with variant sentences with swapped objects. In this thesis, the variants would correspond to sentences in which all extrapositions have been undone and the candidates for extraposition are placed *in situ*, like the PP in example (52), repeated from (39).

(52) **Original:** Das ist mir ganz klar geworden, [PP-extrap schon bei dieser kurzen Trennung].
Variant: Das ist mir [PP-insitu schon bei dieser kurzen Trennung] ganz klar geworden.
'That has become very clear to me, even from this short separation.'

However, as discussed in Section 7.1, the base position can only be determined unambiguously for attributive constituents like relative clauses. Creating a corpus of variants means moving these constituents to their position adjacent to the antecedent. The implementation is straightforward. Each extraposed RelC is placed behind the last token of its antecedent, as in example (53), repeated from (43). If the RelC is preceded by punctuation or a conjunction, this material can be moved to the left as well.[78]

(53) **Original:** Es gehe dem ſelben menſchen / wie es [Antec den Steten] ergangen iſt / [RELC-extrap die der HERR one barmhertzigkeit vmbgekert hat.]
Variant: Es gehe dem ſelben menſchen / wie es [Antec den Steten] / [RELC-insitu die der HERR one barmhertzigkeit vmbgekert hat] ergangen iſt.
'May that person be treated like the cities that the Lord converted without mercy.'

[78] Theoretically, a comma would have to be inserted after the re-located RelC, too. It has to be kept in mind that this changes the number of tokens in the sentence, which might be relevant for subsequent analyses. Given the deviant punctuation in historical texts, I ignore the issue here.

Figure 7.5.: Confusion matrix for the identification of extraposition. Only errors are displayed, i.e., the diagonal displays boundary errors.

For coordinated RelCs, the linear order of elements should be preserved, and embedded antecedents are moved together with the dominating clause (and, in turn, their relative clause), as in example (54) from the DTA.

(54)
Original: Vorausschicken müssen wir hiebei daß wir [$_{\texttt{Antec}_1}$ zu denjenigen Elsäßern] gehören [$_{\texttt{RELC-extrap}_1}$ die sich von Herzen darüber freuen daß Elsaß nun wiederum zu seiner ursprünglichen Stammesart zurückkehrt und [$_{\texttt{Antec}_2}$ seinen deutschen Charakter] wieder gewinnen soll, [$_{\texttt{RELC-extrap}_2}$ welchen es sich durch eine mehr denn zweihundertjährige französische Herrschaft hindurch großentheils zu wahren gewußt hat]].

Variant: Vorausschicken müssen wir hiebei daß wir [$_{\texttt{Antec}_1}$ zu denjenigen Elsäßern] [$_{\texttt{RELC-insitu}_1}$ die sich von Herzen darüber freuen daß Elsaß nun wiederum zu seiner ursprünglichen Stammesart zurückkehrt und [$_{\texttt{Antec}_2}$ seinen deutschen Charakter], [$_{\texttt{RELC-insitu}_2}$ welchen es sich durch eine mehr denn zweihundertjährige französische Herrschaft hindurch großentheils zu wahren gewußt hat]] wieder gewinnen soll gehören.

'We must say in advance that we are among those Alsatians who are heartily pleased that Alsace is now returning to its origins and is to regain its German character, which it has largely managed to preserve through more than two hundred years of French rule.'

In principle, one could also think of creating a corresponding extraposed variant for *in situ* constituents by artificially moving them to the post-field. This is more difficult than the opposite direction, though. First, the set of constituents would have to be restricted to elements that can be extraposed (e.g., no pronouns, Zifonun et al. 1997) and that are placed in the middle field to prevent accidentally causing an empty pre-field in V2 clauses. Also, the original sentences must have an explicit right bracket to enable unambiguous extraposition. But, at the same time, the post-field of the original sentence should still be empty. Although it is possible to place several constituents in the post-field, this is uncommon (at least in modern German) and would require establishing a valid order of post-field elements in the variant sentence (Zifonun et al. 1997). And even if all of those conditions were met, it would still be difficult to ensure that only plausible variants are created. Arbitrarily moving constituents from the middle field to the post-field will likely generate many invalid variant sentences because various factors influence whether or not a constituent is extraposed (cf. Chapter 2.2). Since only realistic variants should be used for a meaningful analysis, the problematic variant sentences would have to be filtered out, e.g., with acceptability ratings – which likely depend on the context and are generally not available for historical data. As a consequence, only the uncontroversial variants of extraposed attributive relative clauses will be created and used in the example application in Chapter 8.

7.4. Discussion

In this chapter, the automatic identification of extraposition was explored. Using the different syntactic analyses from the previous chapters, candidates for extraposition were identified in modern and historical data sets via constituency parsing, and their position within the topological field analysis was determined as *in situ* or extraposed (or ambiguous, for relative clauses).

The results show that *in situ* phrases are recognized with high accuracy, whereas the identification of extraposed phrases is not reliable yet. The highest F_1-scores are achieved for extraposed `PPs`, with 43%–82%. For relative clauses, the differences between *in situ*, ambiguous, and extraposed instances are much less pronounced. In modern German, they can be identified with F_1-scores of 77% to 96%. For historical German, results range from 67% to 85%.

The observed differences between phrases and clauses were traced back to (i) the distinctive structure of RelCs that helps with their identification and (ii) the relevance of the post-field for determining the element's position. As Chapter 5 has shown, post-fields are among the less reliably identified topological fields. And while extraposed phrases can only be recognized with an (at least partly) correct identification of post-fields, identifying the antecedent and especially its right boundary is more relevant to determine the position of RelCs.

As the evaluation in Section 7.1 has demonstrated, simple heuristics are sufficient to identify antecedents and, thus, the base position of attributive relative clauses, given the annotations from Chapter 6. In Section 7.3, this information was exploited to create a corpus of variants in which extraposed RelCs are artificially 'moved back' to their base position. Such a variant corpus can be used to compare sentences with and without extraposition, e.g., regarding their information profile (cf. Chapter 2.2.3).

Overall, the results from this chapter have shown what is and is not (yet) possible with the developed methods concerning the automatic identification of extraposition. For phrases, which are only rarely extraposed in modern standard German but also in other language registers, spoken, and historical data, the automatic recognition is not very reliable yet. Before the automatic results are used for quantitative analyses, further improvements should be made. Potential steps could include, but are not limited to:

- Normalize the historical data and experiment with word-based (neural) models for topological field analysis and constituency parsing.
- Create historical training data and train a topological field model specifically for historical German that can analyze the complex sentence structures of Early New High German.
- Create an annotated data set of non-standard language with relevant proportions of post-fields and/or use active learning (e.g., Tang et al. 2002) to improve the recognition of (uncomplex) post-fields.
- Improve constituency parsing for non-standard language in general.

For relative clauses, the results seem robust enough for first quantitative studies even though further improvements could be achieved with the steps listed above – perhaps complemented with experiments on dependency parsing for an optimized recognition of non-adjacent antecedents. In the example application in the next chapter, I will focus solely on the automatically identified RelCs, leaving the analysis of extraposed phrases for future work.

CHAPTER 8

Example Application

In this thesis, computational methods for the automatic analysis of extraposition were developed. In this chapter, the methods are exemplarily applied to modern and historical German to explore the effects of different factors on the extraposition of relative clauses. I will focus on four factors that presumably influence whether or not a relative clause is placed in the post-field: time, length, orality, and information density (cf. Chapter 2).

The goal of this chapter is not only to shed light on the causes of extraposition, though. Primarily, the intention is to demonstrate the usefulness of the developed methods for linguistic studies, particularly but not only for historical language. With manual annotation, the bottleneck of such studies will always be a lack of annotated data, simply due to natural timely and financial limits of human annotation. With the application of computational methods, these limits are significantly reduced (in the case of semi-automatic approaches) or removed entirely (for purely automatic approaches). Once the necessary tools and models are created, theoretical linguistic assumptions and qualitative observations can be tested quantitatively against almost arbitrarily large amounts of data to arrive at statistically significant conclusions without additional manual labor.

The analyses in this chapter are based on large data sets of modern and historical German from various registers. The data sets are briefly introduced in Section 8.1, and their automatic annotation and the creation of language models are described in Section 8.2. Section 8.3 presents the results of the quantitative analysis for each of the four factors: time (Section 8.3.1), length (Section 8.3.2), orality (Section 8.3.3), and information density (Section 8.3.4). The chapter concludes with a short discussion of the findings in Section 8.4.

8.1. Data

A quantitative exploration of the influence of different factors on the diachronic development of extraposition requires large corpora of modern and historical German. My data selection was guided by four main criteria:

Time period The data sets should cover the whole relevant time period, i.e., from Early New High German, when the sentence brackets were finally established, to present-day German. In the following, I organize the texts into two groups. All texts from 1900 or later are considered as 'modern', whereas older data counts as 'historical'. From a linguistic perspective and especially from a syntactic point of view, this boundary is rather arbitrary since texts from the early New High German period are already very similar to present-day German. However, a standardized German orthography only really emerged after the second orthographical conference in 1901 (Augst et al. 1997). So the historical data shows a higher degree of word form variation, which likely affects the accuracy of automatic annotations and the perplexity of language models. Data sets with orthographical normalizations and manually created annotations should generally be preferred for historical German.

Register Extraposition is usually considered a characteristic of oral language (cf. Chapter 2.2.2). To quantitatively test this claim, the selected data sets should cover different genres and registers with varying degrees of orality, ranging from very literate styles (e.g., in news or science) to oral-like data (e.g., in plays, subtitles, or transcripts). Of course, not every register is equally available for each time window. For example, data from the news or spoken registers is sparse for earlier time periods. Also, the orality (and other characteristics) of a register may change over time, as shown by Degaetano-Ortlieb et al. (2019) for scientific English. Including a multitude of genres will ensure that different degrees of orality are captured. Combined with a text-wise orality measure (Ortmann and Dipper forthcoming), this will shed light on the relationship between orality and extraposition, also beyond registers.

Annotations While the first two selection criteria concern the broader metadata, the other two criteria are related to the data itself. Firstly, the application of the developed annotation methods requires the availability of POS tags, in particular STTS tags (Schiller et al. 1999). Modern data can be tagged automatically with high accuracy (Ortmann et al. 2019), but this is problematic for the (unstandardized) historical data (see the discussion about orthography above), for which pre-trained models do not exist. Therefore, I only include historical data sets that are already provided with POS tags, which can automatically be mapped to STTS tags if necessary.

Corpus size Finally, quantitative analyses require sufficient amounts of data for meaningful results. A corpus size of >100k tokens or >500 relative clauses is desirable. The creation of language models (Section 8.2) also requires enough training data for low Out-of-Vocabulary (OOV) rates. That excludes the historical gold data sets from the previous chapters.

Based on the given criteria, I selected 25 data sets, which cover a variety of registers for the time from 1300 to 2018. All data sets provide STTS POS tags with sensible accuracy. And except for two of the historical data sets, they also offer enough data to train language models. Section 8.1.1 presents the modern data sets, some of which have already been used in the previous chapters. Section 8.1.2 introduces the historical data sets. An overview of all 25 data sets can be found in Table 8.1.

Corpus	Time	Genre(s)	#Docs	#Sents	#Toks	#Words
Modern						
$\text{Gutenberg}_{\text{Fiction}}$	1900–2012	Fiction, Narrative, Novelette	461	1,580,416	30,247,279	25,133,955
$\text{Gutenberg}_{\text{Folk-Tales}}$	1906–2012	Fable, Fairy, Legend	200	141,985	3,513,608	2,981,885
$\text{Gutenberg}_{\text{Non-Fiction}}$	1900–2009	Report, Tractate	178	612,011	15,292,839	13,118,570
$\text{Gutenberg}_{\text{Speech}}$	1903–1976	Lecture, Speech	16	15,923	446,713	388,222
$\text{OPUS}_{\text{Action}}$	1957–2015	Action, Adventure	101	128,961	800,451	601,714
$\text{OPUS}_{\text{Comedy}}$	1931–2015	Comedy	317	481,291	2,972,874	2,224,350
$\text{OPUS}_{\text{Drama}}$	1921–2016	Drama	298	331,779	2,127,887	1,619,349
SdeWaC	2006	Web	200	20,000	1,175,532	1,051,338
SermonOnline	2018	Sermon	506	86,316	1,493,357	1,257,161
Tiger	1992–1997	News	200	4,572	78,166	67,813
TüBa-D/S	2000	Spoken	14	28,696	296,942	239,897
TüBa-D/W	2014	Encyclopedia	3	28,351	476,387	409,134
TüBa-D/Z	1989–1999	News	364	10,527	196,761	167,915
Historical						
Anselm	1300–1500	Religion	61	11,116	406,263	392,532
$\text{DTA}_{\text{Science}}$	1620–1895	Medicine, Theology	24	13,480	618,565	536,553
$\text{GerManC}_{\text{DRAM}}$	1657–1798	Drama	45	9,841	116,217	95,496
$\text{GerManC}_{\text{HUMA}}$	1654–1798	Humanities	45	4,512	109,658	93,894
$\text{GerManC}_{\text{LEGA}}$	1654–1796	Legal	45	3,594	109,050	93,108
$\text{GerManC}_{\text{NARR}}$	1658–1797	Narrative	45	4,492	109,186	93,719
$\text{GerManC}_{\text{NEWS}}$	1659–1798	News	66	3,943	113,599	98,519
$\text{GerManC}_{\text{SCIE}}$	1663–1799	Science	45	3,995	108,810	93,325
$\text{GerManC}_{\text{SERM}}$	1654–1798	Sermon	45	5,200	107,743	91,916
KaJuK	1625–1889	Autobiography, Chronicle, Diary, Letter, Philosophy	8	2,750	119,838	105,274
ReF.RUB	1350–1605	Chronicle, Devotionals, Fiction, Non-Fiction, Science	39	4,309	142,822	128,754
RIDGES	1482–1652	Herbology	23	4,828	80,555	69,201

Table 8.1.: Overview of the modern and historical data sets for the example analysis. For each data set, the covered time periods, genres, and basic statistics are given. #Words refers to the number of tokens without punctuation.

8.1.1. Modern Data Sets

For modern written German, i.e., texts produced after 1900, an ever-increasing amount of data is available. Still, the temporal distribution and the diversity of registers are limited. Linguistic studies often default to using large newspaper and web corpora from the last decades. Expanding to other data sources can be hindered by copyright or license issues and the availability of curated data sets. In the example analysis, I use standard data sets with protective licenses like the TüBa corpora as well as freely available and less commonly used data sets like modern sermons that must be pre-processed first. The following modern data sets are included:

Gutenberg The Gutenberg project is an online library with over 60k free digital, mostly English books.[79] The German version Projekt Gutenberg-DE offers copyright-free German literature. For this thesis, I use Edition 14,[80] which contains over 8,000 texts by more than 1,700 authors. For the analysis in this chapter, I selected four different subsets of genres:

Gutenberg$_{\text{Fiction}}$	**Gutenberg$_{\text{Folk-Tales}}$**	**Gutenberg$_{\text{Non-Fiction}}$**	**Gutenberg$_{\text{Speech}}$**
Fiction, narrative, novelette	Fable, fairy, legend	Report, tractate	Lecture, speech

Only texts from one of the given genres with a publication date after 1900 are included. In case of conflicting meta information, a semi-automatic check was conducted to filter out texts with a clearly historical orthography, e.g., *vnd* instead of *und* 'and' or *seyn* instead of *sein* 'be'. Also, foreign language texts and dialectal texts were semi-automatically excluded. For the fiction and non-fiction samples, only one text per author is retained in the data set to reduce the effect of personal stylistic preferences. However, due to inconsistencies in the metadata, there may still be more than one text per author in some cases, e.g., if there are different spellings of the author's name.

The Gutenberg corpus is provided as HTML files, which were automatically parsed to extract the text content.[81] Title pages and tables of contents are not included in the output. Sentence and word tokenization were performed with the standard tokenizers from the NLTK,[82] combined with a list of abbreviations and additional heuristics. Earlier experiments showed F_1-scores >95% for sentence segmentation and >99% for word tokenization. In total, the genre subsets comprise between 400k and 30M tokens. STTS POS tags were added automatically with the spaCy tagger.[83] Based on previous evaluations, tagging accuracy can be expected to lie above 94%.

[79] https://www.gutenberg.org/

[80] The corpus as of June 2016 was purchased at https://gutenberg.abc.de/.

[81] Pre-processing of the Gutenberg, OPUS, and SermonOnline corpora was done in the context of my Master thesis and our paper Ortmann et al. (2019), and the resulting data sets are reused here.

[82] https://www.nltk.org/

[83] https://spacy.io/; German transformer model de_dep_news_trf-3.2.0

OPUS The OpenSubtitles corpus (short: OPUS; Lison and Tiedemann 2016) is a large collection of parallel movie subtitles from the OpenSubtitles database.[84] The monolingual German version of 2018[85] contains over 46k subtitles, from which I selected three genre subsets:

$\textbf{OPUS}_{\textbf{Action}}$	$\textbf{OPUS}_{\textbf{Comedy}}$	$\textbf{OPUS}_{\textbf{Drama}}$
Action, adventure	Comedy	Drama

Based on the available metadata, I only included one subtitle per movie and only original German movies because translations will likely affect sentence structure (among other things). The corpus is provided with sentence boundaries and was tokenized and tagged as described for the Gutenberg corpus above. In total, the subsets contain between 800k and about 3M tokens.

SdeWaC The SdeWaC corpus (short for Stuttgart deWaC; Faaß and Eckart 2013)[86] is a large collection of German web pages with more than 800M tokens. The corpus is provided with automatic sentence and word tokenization and STTS POS tags, as well as further annotations like lemmas and syntactic dependencies. I use the first 20k sentences from the data set and re-tag them with the spaCy tagger (see above) because tagger accuracy will likely have improved since the release of the corpus. Since no meta information about the individual web pages is provided, I consider the publication year of the original deWaC corpus (2006) as the date of origin. In total, the data subset contains about 1M tokens.

SermonOnline While religious texts were a common register in historical time periods, nowadays, the genre is much less common. To allow for diachronic comparisons, I use German sermons from the SermonOnline database, which provides free Christian sermons in various languages.[87] The texts were automatically sentence segmented, tokenized, and tagged as described above. In total, the data set comprises about 1.5M tokens. It has to be mentioned that the 506 texts were written by only 18 different authors who contributed between 1 and 292 sermons. Since no additional meta information is provided, I use the year in which the data was crawled (2018) as the date of origin.

Tiger The Tiger corpus was already used as training and test data in the previous chapters. Here, I use the test section with 200 newspaper articles for the example analyses and the training section to train the language models. For more information on the corpus, see Chapter 3.

[84] `http://www.opensubtitles.org`
[85] `http://opus.nlpl.eu/download.php?f=OpenSubtitles/v2018/raw/de.zip`
[86] `https://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/sdewac`
[87] The texts were automatically crawled in August 2018 from `http://www.sermon-online.de`.

TüBa-D/S The TüBa-D/S corpus is included as a representation of modern spoken German and was already used in the previous chapters. For more information on the corpus, see Chapter 3. I consider the publication year (2000) as the date of origin, even though the conversations were likely recorded a few years earlier.

TüBa-D/W The TüBa-D/W corpus includes automatically annotated articles from the German Wikipedia of 2014 (De Kok 2014).[88] The data is provided with automatically created sentence segmentation, tokenization, and STTS tagging. For the example analyses, I use the first three files with approximately 28k sentences and about 475k tokens. For language model creation, I also added the next eight files as training data. The corpus is licensed under CC BY-SA 3.0.

TüBa-D/Z The TüBa-D/Z corpus was already used as training and test data in the previous chapters. Here, I use the test section with 364 newspaper articles for the example analyses and the training section to train the language models. For more information on the corpus, see Chapter 3.

8.1.2. Historical Data Sets

The availability of historical data sets is much more restricted than for modern German. Especially for earlier time periods, written sources are sparse, and careful manual curation is necessary before the data can be analyzed automatically with computational methods. After the invention of the printing press in the 15th century, the amount of written (German) language has been steadily increasing. And this process was accelerated even more with the digital revolution. At the same time, the variety of registers increased significantly. Early writings often treat religious topics, whereas other registers like science or news only emerged over time.

Besides the skewed distribution of registers, a look at available data sets of historical German also shows that many do not meet my other selection criteria. Either they do not provide POS tags (e.g., Bonner Frühneuhochdeutschkorpus (Fisseni 2017), Mannheimer Korpus Historischer Zeitungen und Zeitschriften (IDS 2013), Kasseler Junktionskorpus (Ágel and Hennig 2008), Wikisource,[89] Gutenberg[90]), or the available POS tags have low accuracy, especially for older texts (e.g., DTA; BBAW 2021). Due to the manual effort that is required to create a historical corpus, a lot of the data sets are also too small for a quantitative analysis or to train language models on them (e.g., HIPKON (Coniglio et al. 2014), Fürstinnenkorrespondenzen (Lühr et al. 2013), Mercurius (Demske 2005), RIDGES (Lüdeling et al. 2022)).

Since POS tags are the only indispensable requirement for my analysis, I only considered data sets with sensible POS tagging and selected the ones that meet as many of the other criteria as possible. The following historical data sets are included in the example analysis:

[88] http://www.sfs.uni-tuebingen.de/ascl/ressourcen/corpora/tueba-dw.html
[89] https://de.wikisource.org/
[90] https://gutenberg.abc.de/

Anselm The Anselm corpus (Dipper and Schultz-Balluff 2013) contains writings from the 14th to 16th century in Early New High German, Middle Low German, and Middle Dutch. The 61 German texts that are used in this thesis deal with the religious treatise of Anselm of Canterbury asking questions to Mary about the Passion of Jesus (*Interrogatio Sancti Anselmi de Passione Domini*, 'Questions by Saint Anselm about the Lord's Passion'). The corpus is provided with normalizations, lemmas, morphology, and custom POS tags, which were mapped to the modern STTS tagset.[91] Annotations and language models in this chapter are based on the provided modernized word forms. In total, the data set contains about 400k tokens. It is licensed under CC BY-SA 4.0.

DTA$_{\text{Science}}$ As mentioned in Chapter 3, the German Text Archive (BBAW 2021) provides large amounts of automatically annotated texts from the 16th to the early 20th century, including various genres. In the previous chapters, I used a small subset from this corpus that was manually annotated for evaluation purposes. However, for the quantitative analyses in this chapter, more data is necessary. Since the automatic POS annotations are too unreliable (cf. Chapters 3, 5), I chose a subset of 24 texts (11 medical, 13 theological texts; 17th–20th century) that were used in our project to investigate extraposition.[92] At the time of the analyses, manually corrected POS tags were available for the oldest 19 texts. The annotations and models in this chapter are based on the provided orthographic normalizations. Only sentences that are relevant to the project (i.e., containing some candidate for extraposition) are included. In total, the sample comprises about 618k tokens. The DTA is licensed under CC BY-SA 4.0.

GerManC The GerManC data set (Bennett et al. 2007) is a representative corpus of German between 1650 – 1800, intended for comparative diachronic studies of grammar and vocabulary in German and English. It contains texts from seven different registers, which I treat as separate subcorpora for the example analysis:

GerManC$_{\text{DRAM}}$	**GerManC$_{\text{HUMA}}$**	**GerManC$_{\text{LEGA}}$**	**GerManC$_{\text{NARR}}$**
Plays	Humanities	Legal texts	Narratives
GerManC$_{\text{NEWS}}$	**GerManC$_{\text{SCIE}}$**	**GerManC$_{\text{SERM}}$**	
News	Science	Sermons	

The corpus is provided with custom POS tags, lemmas, morphology, dependency annotation, and normalization. POS tags are automatically mapped to the STTS tagset.[93] In total, the data set contains over 700k tokens (about 100k per register) and is licensed under CC BY-NC-SA 3.0 DE.

[91] The following POS tags were mapped to STTS tags: ADJN → ADJD, ADVREL → ADV, PDS_PRELS → PRELS, PPOSN → PPOSAT, PRELF → PRELS, PTKVZ_APPR → PTKVZ

[92] https://github.com/rubcompling/C6Samples

[93] The following POS tags were mapped to STTS tags: NA → NN, PROAV → PAV, PAVREL → PAV, PWAVREL → PWAV, PWREL → PWS, SENT → $., $- → $., $' → $(, $) → $(, _ → $(

KaJuK The Kasseler Junktionskorpus (short: KaJuK; Ágel and Hennig 2008) was compiled for the investigation of conceptual orality. It contains 6 conceptually oral and 2 conceptually literate texts from the 17th and 19th century. The data is manually enriched with detailed information for the orality analysis but originally lacks basic annotations like POS tags. For our latest study on orality, we automatically annotated the corpus with POS tags using the spaCy tagger and some basic normalization (estimated tagging accuracy is about 88%; Ortmann and Dipper forthcoming). With about 120k tokens, the corpus is relatively small. It is licensed under CC BY 3.0.

ReF.RUB The Reference Corpus of Early New High German (Wegera et al. 2021) is a representative, balanced data set of German from 1350 – 1650. The syntactically annotated part (ReF.UP) was already used in Chapter 6 to train and evaluate chunking and constituency parsing. Here, I selected the larger ReF.RUB subcorpus,[94] which is annotated with POS tags, lemmas, and morphology. For the analyses, I use only the manually annotated part of the data set (about 142k tokens), whereas the complete subcorpus (>1.2M tokens) is used for language model creation. The HiTS POS tags (Dipper et al. 2013) were mapped to STTS tags following the rules in Table A.4 (in the appendix). The corpus does not include normalization, but the modern tokenization and simplified word forms with only ASCII characters already reduce some variation. ReF is licensed under CC BY-SA 4.0.

RIDGES Herbology The final data set stems from the RIDGES project (Register in Diachronic German Science) and contains 23 scientific texts from the mid 15th to the 20th century (Lüdeling et al. 2022). The corpus is provided with POS tags, lemmas, morphology, and dependency annotations. With about 80k tokens, the corpus is relatively small. It is licensed under CC BY 3.0.

8.2. Annotation

For the example analysis in this chapter, the data sets from the previous section are automatically enriched with the necessary annotations. Section 8.2.1 describes the automatic identification of relative clauses and extraposition. In Section 8.2.2, the calculation of conceptual orality is explained. Finally, Section 8.2.3 describes the creation of language models and surprisal calculation for an information-theoretic analysis of the data.

8.2.1. Extraposition

To investigate the post-field placement of relative clauses, the 25 modern and historical data sets are automatically annotated with topological fields, relative clauses, and extraposition with the methods developed in this thesis. For the modern data, the `Punct` and `News1` models are used because they performed best across different modern registers. The only exception are the three corpora

[94] There is some overlap between the different subcorpora of ReF. In particular, seven texts from the ReF.RUB data set are also part of the ReF.UP subcorpus, five of which are provided with manual annotations.

Tiger, TüBa-D/S, and TüBa-D/Z, for which I already have gold annotations from Chapter 7 that are reused here.

The historical data sets are annotated with the `Punct` and `Mix` models, using the orthographic normalization if possible.[95] Since the `Mix` model is trained on modern and historical German, it will likely achieve the best results across the entire historical time period from 1300 to 1900. Only for the ReF.RUB data, I decided to use the `Hist` model instead because no modernized word forms are available and the data is highly similar to the model's training data, which was mostly taken from ReF.UP, another sub-corpus of ReF (cf. Chapter 6). Only sentences with a maximum of 350 tokens are annotated.

Table 8.2 shows the number of automatically identified relative clauses in each of the data sets, ranging from 330 to 278k RelCs. In total, 563k relative clauses with their antecedents have been identified and located within the topological field structure. These numbers are, obviously, unthinkable for manual annotation, underlining the power of computational methods and their usefulness for linguistic analyses. Instead of spending years on expensive and effortful manual work, complete movies or books can now be annotated with topological fields, a constituency analysis, and RelC extraposition in minutes.[96]

The identified relative clauses are assigned to the date of their source text (Table 8.1) for an inspection of the diachronic development of extraposition (Section 8.3.1). Also, for each RelC, the number of included words is retrieved to explore length as a factor for extraposition (Section 8.3.2). While the automatic annotations are always created for complete sentences (including punctuation), the example analyses will only consider words, i.e., tokens without punctuation (for a discussion on measures of length vs. complexity, see Chapter 2.2.1).

8.2.2. Orality Score

As explained in Chapter 2, extraposition is considered mainly an oral phenomenon. However, the term 'orality' is often used rather vaguely. Based on our previous work (Ortmann and Dipper 2019; Ortmann and Dipper 2020; Ortmann and Dipper forthcoming), I aim at an objective investigation of the relationship between orality and extraposition. The degree of orality is operationalized in two ways.

[95] As mentioned in Chapter 6.2, the parser models are essentially unlexicalized. However, the constituency model can still fall back on the word forms if no parse is found for the given POS sequence. Supplying normalized word forms will likely improve the result in these cases. Normalization becomes more relevant in the context of language model creation (Section 8.2.3).

[96] Annotation speed depends on several factors, including the length and complexity of sentences, the selected models, and the available hardware. Most of the required time is consumed by the topological field annotation and constituency analysis with the Java-based parser. Better computational efficiency may be achieved with another (faster) parser.

Corpus	#RelC insitu	ambig	extrap	all
Modern				
$\text{Gutenberg}_{\text{Fiction}}$	106,871	94,322	77,690	278,883
$\text{Gutenberg}_{\text{Folk-Tales}}$	9,907	10,457	8,407	28,771
$\text{Gutenberg}_{\text{Non-Fiction}}$	74,682	54,609	49,042	178,333
$\text{Gutenberg}_{\text{Speech}}$	2,466	1,497	1,914	5,877
$\text{OPUS}_{\text{Action}}$	568	620	320	1,508
$\text{OPUS}_{\text{Comedy}}$	1,784	2,048	972	4,804
$\text{OPUS}_{\text{Drama}}$	1,656	1,812	1,032	4,501
SdeWaC	6,310	3,668	6,143	16,121
SermonOnline	5,831	5,458	3,949	15,238
Tiger	205	176	175	556
TüBa-D/S	100	142	88	330
TüBa-D/W	750	1,441	680	2,871
TüBa-D/Z	658	550	397	1,605
Historical				
Anselm	452	560	253	1,265
$\text{DTA}_{\text{Science}}$	5,454	3,274	3,683	12,411
$\text{GerManC}_{\text{DRAM}}$	292	168	235	695
$\text{GerManC}_{\text{HUMA}}$	605	299	469	1,373
$\text{GerManC}_{\text{LEGA}}$	579	127	313	1,019
$\text{GerManC}_{\text{NARR}}$	485	343	557	1,385
$\text{GerManC}_{\text{NEWS}}$	552	187	493	1,232
$\text{GerManC}_{\text{SCIE}}$	641	245	471	1,357
$\text{GerManC}_{\text{SERM}}$	567	341	426	1,334
KaJuK	219	220	202	641
ReF.RUB	443	318	281	1,042
RIDGES	202	89	134	425
Total	222,279	182,972	158,326	563,577

Table 8.2.: Number of automatically identified relative clauses by position (`insitu`, `ambig`, `extrap`) and overall in the modern and historical data sets.

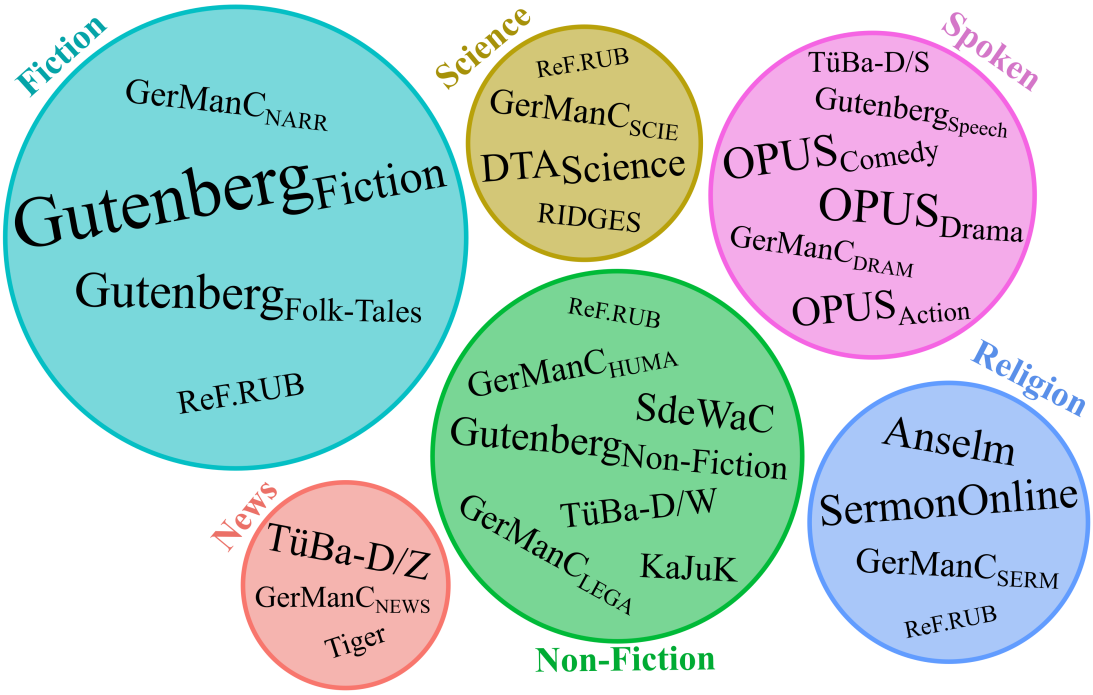


Figure 8.1.: Mapping of data sets to registers. The ReF.RUB corpus includes texts from more than one register (Fiction, Non-Fiction, Religion, and Science).

Firstly, each data set is mapped to one of six registers: News, Science, Non-Fiction, Fiction, Religion, and Spoken. The order of registers corresponds to their (diachronically) expected degree of orality from most literate (News) to most oral (Spoken). I only expect small differences between the news and science texts, with the latter becoming less oral over time, as shown by Degaetano-Ortlieb et al. (2019) for scientific English. Fiction texts are usually more oral than non-fiction but significantly less oral than actual spoken language. Religious texts are traditionally intended for spoken reproduction (e.g., sermons are usually meant to be read aloud), so I expect similarities with the spoken register.

For my analyses, the News register includes newspaper articles from one historical and two modern corpora. Science texts are only available for historical German from four different data sets. The Non-Fiction register comprises a variety of different genres (four historical, and three modern data sets). The Fiction register includes narrations from two modern and two historical data sets. In the Religion register, data from three historical corpora and modern sermons are available. The Spoken set consists of speech transcripts for modern German and spoken-like data from subtitles, written speeches, and historical plays. Figure 8.1 illustrates the mapping of data sets to registers.

As we have shown in Ortmann and Dipper (2019) and Ortmann and Dipper (2020), the categorization into registers is a useful approximation of the general orality of a given text type. However,

as described in Chapter 2.2.2, orality can only be sensibly determined for individual texts. So, in addition to the register mapping, each text is rated with our orality score (Ortmann and Dipper forthcoming) using the COAST implementation.[97] As explained in Chapter 2.2.2, the score is based on the linguistic features of individual texts and allows to objectively compare the degree of orality for large amounts of data. An overview of the features was given in Table 2.1. For easy reference, the features included in the orality score are repeated here with their respective weights and definitions:

mean_word	-0.819	Mean word length.
subord	-0.314	Ratio of subordinating conjunctions (tagged as KOUS or KOUI) to full verbs.
V:N	0.528	Ratio of full verbs to nouns.
PRON1st	0.717	Ratio of 1st person pronouns with lemmas *ich* 'I' and *wir* 'we' to all words.
DEM	0.060	Ratio of demonstrative pronouns (tagged as PDS) to all words.
DEMshort	0.365	Proportion of demonstrative pronouns (tagged as PDS) with lemmas *diese* or *die* 'this/these', which are realized as the short form (lemma *die*).
PTC	0.104	Proportion of answer particles (*ja* 'yes', *nein* 'no', *bitte* 'please', *danke* 'thanks') to all words.
INTERJ	0.276	Proportion of primary, i.e., one-word interjections (e.g., *ach*, *oh, o*, *bravo*, *halleluja*, *hmm*) to all words.

As can be seen from the definitions, the features are based on word forms, STTS POS tags, and lemmas. Since the lemma-based features explicitly distinguish between specific lemmas, e.g., *die* vs. *diese*, the data must contain precisely these lemmas to obtain correct results. For data sets that are provided with a similar but slightly different lemma annotation, the relevant lemmas are mapped to the target lemmas. For example, short demonstrative pronouns are lemmatized as *der* in the ReF.RUB data and *d-* in the Anselm data, which are mapped to *die* for the orality analysis. If the lemma analysis of a data set is entirely different (RIDGES, SdeWaC, Tiger) or if no lemmatization is available at all (Gutenberg, KaJuK, OPUS, SermonOnline, TüBa-D/S), the necessary lemmas are determined based on word forms using the simple rules described in Ortmann and Dipper (forthcoming).

Given the required annotations, the features are then determined for each of the 25 data sets. As explained in Chapter 2.2.2, the features can take on very different values (e.g., an average word length of 5 letters vs. a proportion of interjections of 0.1%), so values are scaled with a linear transformation to the area between 0 and 1. Since the scaling is based on the minimum and maximum feature values in the data set, the orality score is a relative measure, and scores are only comparable *within* one data set. To compare texts from different corpora, the results of all texts must be scaled in the same way, i.e., first, the 25 data sets are joined and then scaled to a common space based on

[97] https://github.com/rubcompling/COAST

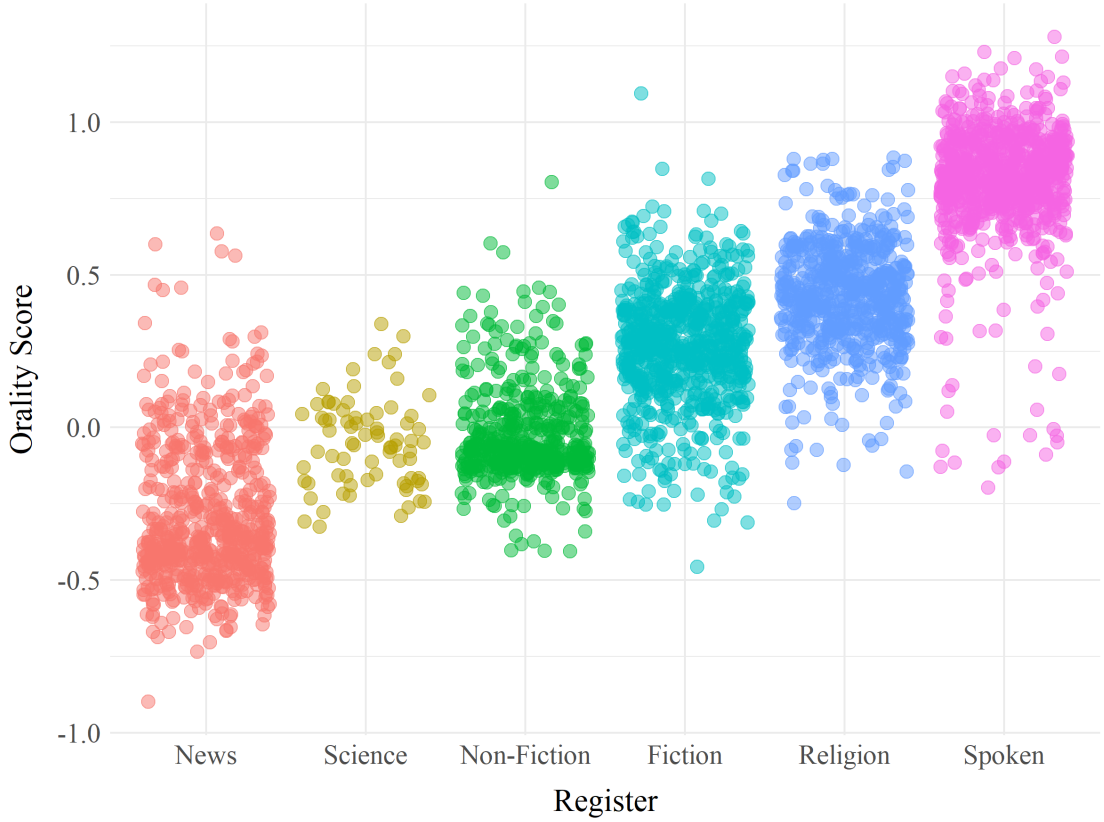


Figure 8.2.: Distribution of orality scores within the six registers (from left to right): News, Science, Non-Fiction, Fiction, Religion, and Spoken. Scores range from -0.9 to 1.3, with lower scores indicating a more literate and higher scores a more oral style.

the minimum and maximum feature values *across* all data sets. This way, the orality score of a religious text from, e.g., the Anselm data, is comparable to the score of a sermon in the SermonOnline data.

Figure 8.2 shows the distribution of orality scores in the six different registers. The expected differences in the general orality of registers are clearly visible in the plot and also confirmed by a one-way ANOVA on a stratified sample of 75 texts per register with a large main effect of register ($F(5, 444) = 285.8$, $p < 0.001$, $\eta^2 = 0.76$).[98] A post hoc pairwise comparison with Tukey's HSD test returns significant differences between all registers except Science and Non-Fiction. Still, the plot also shows the variability within registers with standard deviations between $s = 0.15$ and

[98] I use a stratified sample instead of all data points to reduce the influence of large sample size on the statistical results (see also the discussion in Sections 8.3.2 and 8.3.3). All statistical tests in this chapter are performed with the R software (R Core Team 2018), and the car and lsr packages.

$s = 0.23$. A large part of this variation can be attributed to differences between the included data sets, e.g., between transcripts of spoken language with a score greater than 1 and written speeches with scores below 0.25. Or between historical news with scores around -0.2 and modern news with scores below -0.36. In general, grouping texts into registers is a good, practical operationalization of orality, but more precise, nuanced judgments are possible with the orality score.

8.2.3. Language Models and Surprisal

The exploration of information-theoretic measures as a factor for extraposition (Section 8.3.4) requires the creation of probabilistic language models (LMs). As explained in Chapter 2.2.3, such language models are used to predict how probable or, in other words, how *surprising* a word is in a given context. For meaningful predictions, a language model should be trained on data that is similar to the data it should predict. A model that is trained on newspaper text will make realistic predictions regarding the probability of words in other newspaper text, but will be very surprised when confronted, e.g., with spoken language transcripts. Similarly, a model trained on modern data will be surprised by historical data. As demonstrated by Bizzoni et al. (2020) or Degaetano-Ortlieb et al. (2021), this effect can be exploited for the investigation of language change and differences between registers. However, the influence of information density on extraposition, i.e., the synchronic choice between extraposing the relative clause or leaving it *in situ*, can only be explored with good fitting LMs for specific registers and time periods.

Therefore, I trained separate language models for each of the data sets. Since the limited training data, especially for historical language and non-standard registers, leads to data sparsity problems for n-grams with values of $n > 2$, I use classical bigram LMs with Jeffreys-Perks smoothing ($\lambda = 0.5$; Jeffreys 1946) for the example analysis. Models are trained on (normalized) word forms representing the lexical level and STTS POS tags representing the syntactic level. Punctuation is ignored during training and surprisal calculation.

Contrary to the other analyses, I focus only on the difference between extraposed and *in situ* RelCs here, ignoring the ambiguous cases. As test data, I randomly select 250 sentences with at least one extraposed and 250 sentences with at least one *in situ* RelC per data set. If a data set contains less than 250 extraposed RelCs, the amount of *in situ* RelCs is also limited to approximately the same number. However, as one sentence can include both types of RelCs, the total number of RelCs may vary slightly.

The remaining sentences from the data sets are used as training data for the language models. This approach ensures a good fit of the models to the test data without including the test data in the training procedure. Since only a subset of relative clauses is held out for the analysis, the training data still contains enough RelCs for a representative LM. If a data set was deemed too small to serve as its own training data, I included additional similar data, e.g., other scientific texts from the DTA for $\text{DTA}_{\text{Science}}$ or the training sets of the Tiger and TüBa-D/Z corpora. Table 8.3 gives an overview of which data is used to train which model. For two of the historical data sets, KaJuK and RIDGES, there is not enough training data available, so they are not included in the information-theoretic analyses.

Language Model	#Words	#Types	Training Data
Modern			
Gutenberg	41,547,796	811,576	Texts from the genres fiction, narrative, novelette, fable, fairy, legend, report, tractate, lecture, speech
OPUS	4,425,747	147,544	German subtitles from the genres action, adventure, comedy, and drama
SdeWaC	1,024,900	111,787	First 200k sentences of SdeWaC
SermonOnline	1,244,247	55,496	All German sermons from SermonOnline
Tiger	689,141	83,685	Train and test section of the Tiger corpus
TüBa-D/S	236,160	6,368	Complete TüBa-D/S corpus
TüBa-D/W	1,475,327	161,205	Files 0–10 from the TüBa-D/W corpus
TüBa-D/Z	1,490,229	149,472	Train and test section of the TüBa-D/Z corpus
Historical			
Anselm	363,189	8,067	Complete Anselm corpus
DTA	17,700,513	470,864	All medical and theological texts from the DTA
GerManC	525,582	57,809	Complete GerManC corpus
ReF	1,163,530	104,318	ReF.RUB (manual and automatic part)

Table 8.3.: Overview of the trained language models. #Words gives the number of tokens without punctuation. #Types is the number of unique words. For historical data sets, the orthographic normalization is used instead of actual word forms. The last column specifies which data was used for training. For example, all four subsets of the Gutenberg corpus were used to train a joint model *Gutenberg* with 41M words and 811k unique word types. The approx. 500 test sentences per data set are not included in the training data.

Table 8.4 shows the Out-of-Vocabulary (OOV) rates for the models on each test set. Between 0.8% and 7.8% of the words from the test data are not included in the training data and, hence, in the language model. This mainly concerns low-frequency words, as is reflected in the considerably higher OOV rates for types (i.e., unique words) with 2.7% to 25.5%. Relative clauses tend to contain a lower proportion of unknown words than the sentence as a whole, except for the $\text{OPUS}_{\text{Action}}$, TüBa-D/S, Anselm, and ReF.RUB data sets.

Given the trained language models, there are two ways in which I want to explore the effects of information density on extraposition. As explained in Chapter 2.2.3, extraposition could be triggered by high surprisal of the extraposed constituents to prevent peaks of information in the middle field. Or it could be a means to smooth the overall information profile of the sentence. To address the first hypothesis, bigram surprisal values are calculated for all (normalized) word forms and POS tags in the test data with the respective language models (ignoring punctuation). For each extraposed and *in situ* RelC, the surprisal values are summed up and divided by the number of words to calculate mean RelC surprisal.

Model	Corpus	#Words	#Types	$\#Words_{RelC}$	OOV (%) Words	Types	RelC
Modern							
Gutenberg	$Gutenberg_{Fiction}$	15,997	5,432	4,871	0.93	2.74	0.86
	$Gutenberg_{Folk\text{-}Tales}$	19,813	6,002	4,877	1.03	3.37	0.86
	$Gutenberg_{Non\text{-}Fiction}$	19,849	6,837	6,047	1.49	4.29	1.24
	$Gutenberg_{Speech}$	19,177	5,379	6,462	0.79	2.77	0.65
OPUS	$OPUS_{Action}$	6,285	2,188	2,867	2.31	6.58	2.41
	$OPUS_{Comedy}$	6,662	2,185	3,045	2.06	6.18	1.90
	$OPUS_{Drama}$	6,719	2,223	2,979	2.10	6.34	2.01
SermonOnline	SermonOnline	12,914	3,138	4,002	2.07	8.38	1.90
SdeWaC	SdeWaC	26,438	8,701	8,691	7.13	21.18	6.21
Tiger	Tiger	8,701	3,772	3,289	7.52	17.13	7.15
TüBa-D/S	TüBa-D/S	3,737	851	1,110	2.41	10.46	3.06
TüBa-D/W	TüBa-D/W	13,445	5,621	5,206	6.98	16.46	6.30
TüBa-D/Z	TüBa-D/Z	14,264	5,662	5,603	6.05	15.17	5.05
Historical							
Anselm	Anselm	29,343	2,896	4,407	1.22	10.26	1.34
DTA	$DTA_{Science}$	22,546	6,791	7,074	1.88	6.10	1.81
GerManC	$GerManC_{DRAM}$	9,969	3,175	3,678	5.02	15.59	4.87
	$GerManC_{HUMA}$	19,001	5,775	6,402	6.83	21.68	6.78
	$GerManC_{LEGA}$	30,237	7,668	10,226	7.04	25.48	6.71
	$GerManC_{NARR}$	18,448	5,403	6,088	6.28	20.97	6.19
	$GerManC_{NEWS}$	21,294	6,467	6,971	7.80	24.60	6.97
	$GerManC_{SCIE}$	19,840	5,594	6,828	6.71	22.51	6.69
	$GerManC_{SERM}$	15,606	4,244	6,005	4.29	15.29	4.25
ReF	ReF.RUB	25,927	7,628	5,936	6.31	20.11	6.35

Table 8.4.: Out-of-Vocabulary (OOV) rates for the language models on the test data sets. #Words gives the number of tokens in the test data set without punctuation. #Types is the number of unique words. $\#Words_{RelC}$ is the number of words that are part of *in situ* and extraposed relative clauses. For historical data sets, the orthographic normalization was used instead of actual word forms. OOV_{Words} is the percentage of words, and OOV_{Types} the percentage of types from the test data that are not included in the language model. OOV_{RelC} is the percentage of words from the test data that are part of *in situ* or extraposed relative clauses but not included in the language model.

The second hypothesis can be tested using DORM and the corpus of variants method. For each test sentence with at least one extraposed RelC, a variant sentence is generated as described in Chapter 7.3, and surprisal values are re-calculated for the newly ordered words.[99] Depending on the number of originally extraposed RelCs, variant sentences may include different amounts of change, which is not further considered here. For each test sentence, $\text{DORM}_{\text{orig}}$ and $\text{DORM}_{\text{variant}}$ values are calculated. As described in Chapter 2.2.3, DORM is defined as the sample variance of the rolling means of adjacent surprisal scores. However, the definition does not restrict how the surprisal scores are obtained. In this chapter, I experiment with different options including:

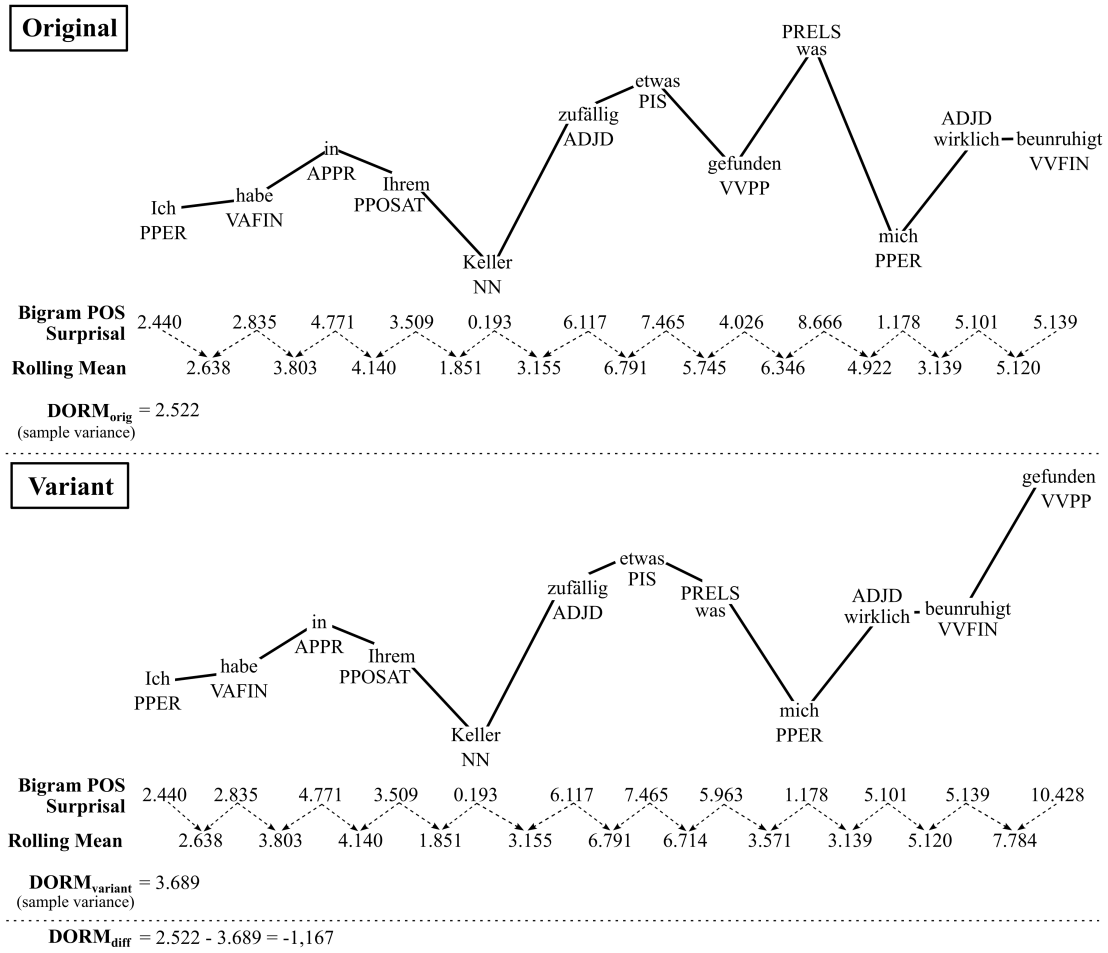
(i) Bigram surprisal of (normalized) word forms and POS tags

(ii) Bigram surprisal of words and mean bigram surprisal of constituents

Constituents are read off from the automatically generated constituency trees and roughly correspond to phrases as defined in Chapter 6.2, without the restriction to only four phrase types. Relative clauses are analyzed as a single constituent.

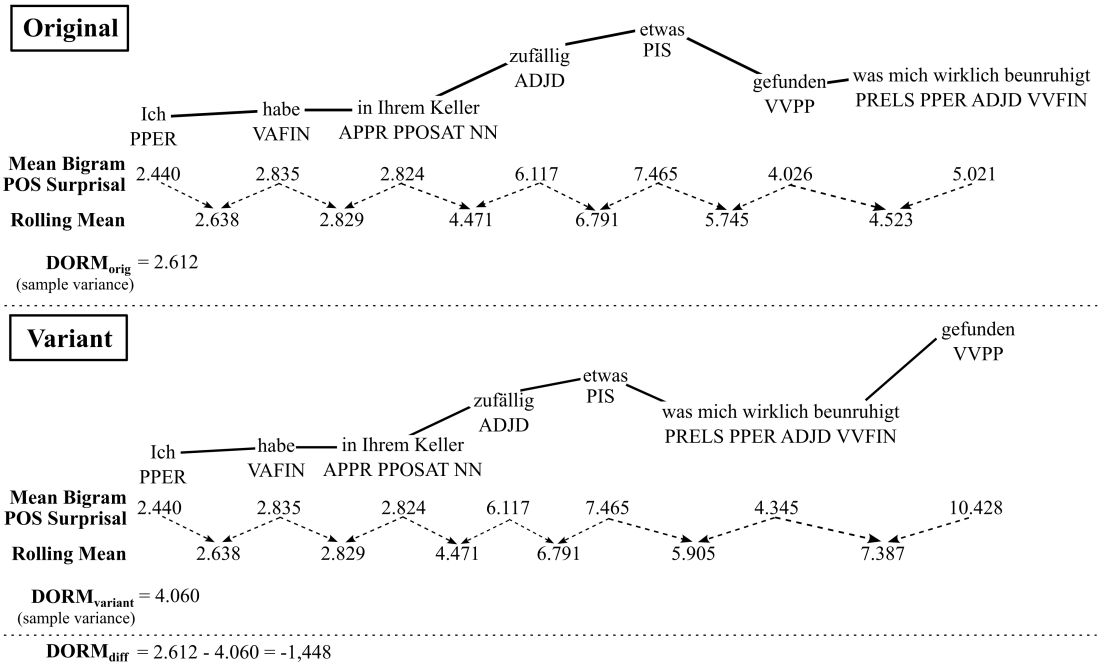
The resulting $\text{DORM}_{\text{orig}}$ and $\text{DORM}_{\text{variant}}$ values based on the different surprisal scores are used to calculate $\text{DORM}_{\text{diff}}$ scores, as described in Chapter 2.2.3. A negative $\text{DORM}_{\text{diff}}$ value indicates that extraposition improves the overall information profile of the sentence. A positive value signifies that leaving the RelC *in situ* would have resulted in a smoother distribution. An example calculation based on bigram POS surprisal of words can be found in Figure 8.3. Figure 8.4 shows the calculation with mean constituent surprisal.

[99] While unigram probabilities are independent of the surrounding context, re-calculation is necessary for n-grams with $n > 1$. The original study by Cuskley et al. (2021) with DORM and UIDO only considers unigram surprisal because determining the optimal information profile becomes computationally expensive with conditional probabilities that must be re-calculated for all possible orders. With our $\text{DORM}_{\text{diff}}$ measure and the corpus of variants, only two alternatives are compared, which makes the probability calculation unproblematic.



'I happened to find something in your basement that really worries me.'

Figure 8.3.: Example calculation of DORM values based on bigram POS surprisal for a sentence from the $\text{OPUS}_{\text{Drama}}$ test set. For each word in the original and variant sentence, bigram POS surprisal is calculated with the OPUS language model. As the curves illustrate, surprisal values differ between original and variant wherever a new bigram is created by undoing the extraposition (in the example: *etwas/PIS-was/PRELS*, *beunruhigt/VVFIN-gefunden/VVPP*). $\text{DORM}_{\text{orig}}$ and $\text{DORM}_{\text{variant}}$ values are then determined by taking the rolling mean of every two adjacent surprisal scores and calculating the sample variance of the rolling means. $\text{DORM}_{\text{diff}}$ is obtained by subtracting the DORM value of the variant sentence from the DORM value of the original sentence. A negative $\text{DORM}_{\text{diff}}$ value indicates that the original sentence has a smoother information profile, whereas a positive value means that the variant profile would be more uniform.



'I happened to find something in your basement that really worries me.'

Figure 8.4.: Example calculation of DORM values based on mean bigram POS surprisal of constituents for a sentence from the $OPUS_{Drama}$ test set. First, bigram POS surprisal is calculated for each word in the original and variant sentence with the OPUS language model (cf. Figure 8.3). Then, mean constituent surprisal is calculated as the sum of the individual surprisal scores divided by the number of words in a constituent. $DORM_{orig}$ and $DORM_{variant}$ values correspond to the sample variance of the rolling means of every two adjacent constituent surprisal scores. $DORM_{diff}$ is obtained by subtracting the DORM value of the variant sentence from the DORM value of the original sentence. A negative $DORM_{diff}$ value indicates that the original sentence has a smoother information profile, whereas a positive value means that the variant profile would be more uniform.

8.3. Quantitative Analysis

In the following Sections 8.3.1–8.3.4, the automatically created annotations are used to exemplarily inspect the effects of time, length, orality, and information density on the extraposition of relative clauses.

8.3.1. Time

As already mentioned in Chapter 2, the frequency of extraposition has changed over time. In older stages of German, the proportion of sentences with filled post-fields was higher than in modern German (Schildt 1976). Possible explanations for the diachronic reduction of extraposition include the gradual establishment of the sentence frame, increasing deviation from spoken language, and the development of a written style with dense middle fields.

However, the development did not affect all constituents in the same way. While studies agree that phrases are extraposed less frequently in modern German than in historical German, the picture is less clear for relative clauses. Sahel (2015) reports that the proportion of extraposed RelCs may actually have increased from 64% to 72% between 1650 and 1800. And there is still a significant amount of extraposed RelCs in modern German, with approx. 24–25% in newspaper text (Uszkoreit et al. 1998, also see Chapter 7). However, numbers from these studies are not comparable, as they focus on different subsets of relative clauses, e.g., disregarding ambiguous cases and/or RelCs that are not located in the middle field or post-field. Therefore, temporal developments are difficult to predict, and the role of ambiguous RelCs also remains unknown. Conservatively, I expect to see some change in the proportion of extraposed RelCs, although the direction and other specifics of this change are unclear.

> **Hypothesis**
> The proportion of (unambiguously) extraposed RelCs has changed over time.

Figure 8.5 (top) shows the general diachronic development of RelC position. Except for the earliest time period, the highest percentage of relative clauses is placed *in situ*, closely followed by ambiguous cases in the youngest and oldest data. Unambiguously extraposed RelCs are always less frequent than *in situ* RelCs.

Over time, the data suggests a parallel development of extraposition and embedding in opposition to the number of ambiguous cases. Until 1700, the proportion of extraposed and *in situ* RelCs increases to about 35% and 45%, respectively, before it decreases again. In contrast, the proportion of ambiguous cases decreases to about 20% in 1700. This may be caused by a higher amount of explicit right sentence brackets during the Early New High German period due to sociolinguistic reasons ('prestige' of a complete sentence frame, cf. Takada 1998), which allows to unambiguously determine the position of RelCs as either *in situ* or extraposed. In present-day German, the RelC position is ambiguous for about 35% of the relative clauses.

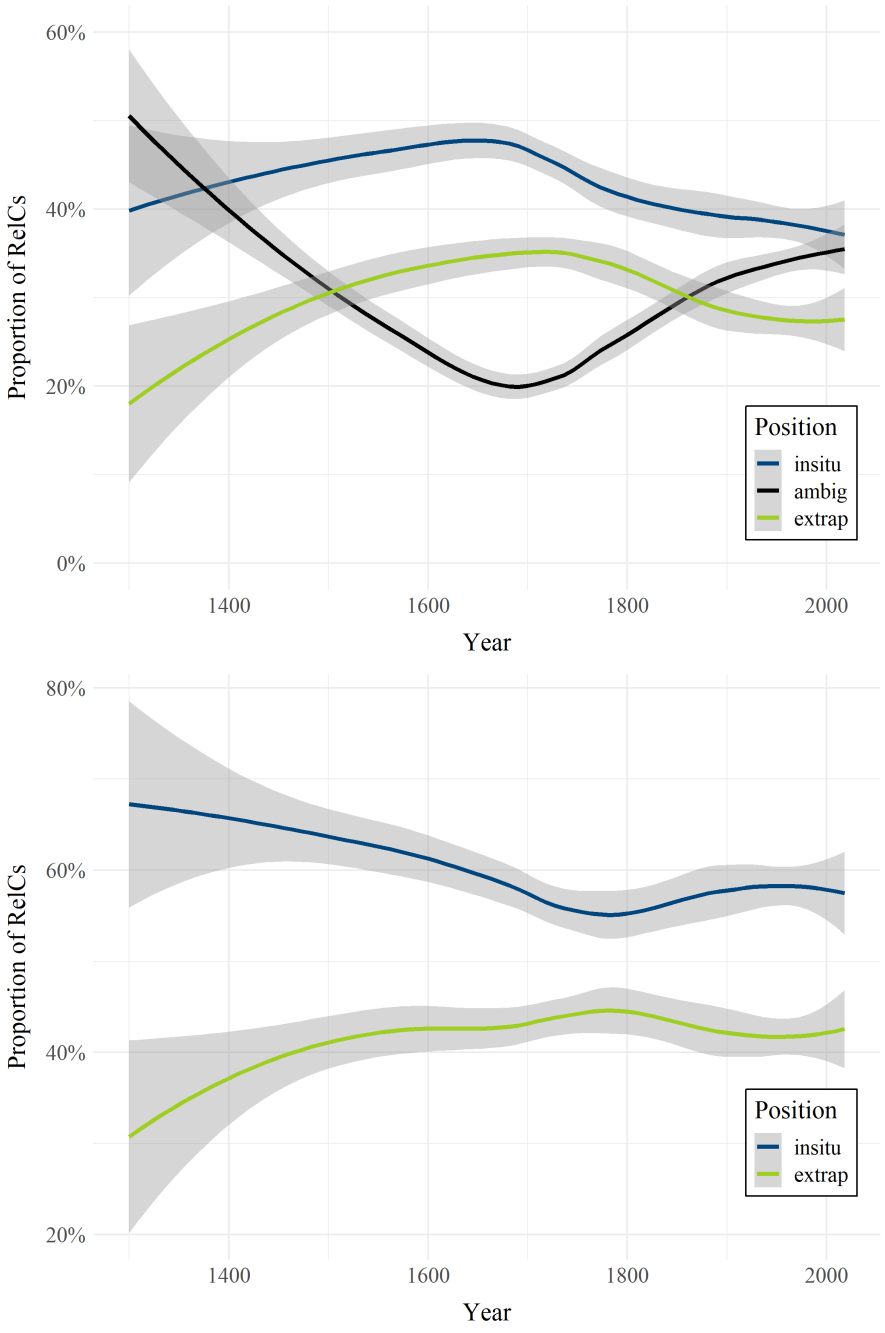


Figure 8.5.: Development of RelC positions over time, including all three positions (top) or only unambiguous RelCs (bottom). For each year, the proportion of *in situ* (blue), ambiguous (black), and extraposed (light green) relative clauses is determined. The plots show the local regression lines (LOESS smoothing) with a confidence interval of 0.95.

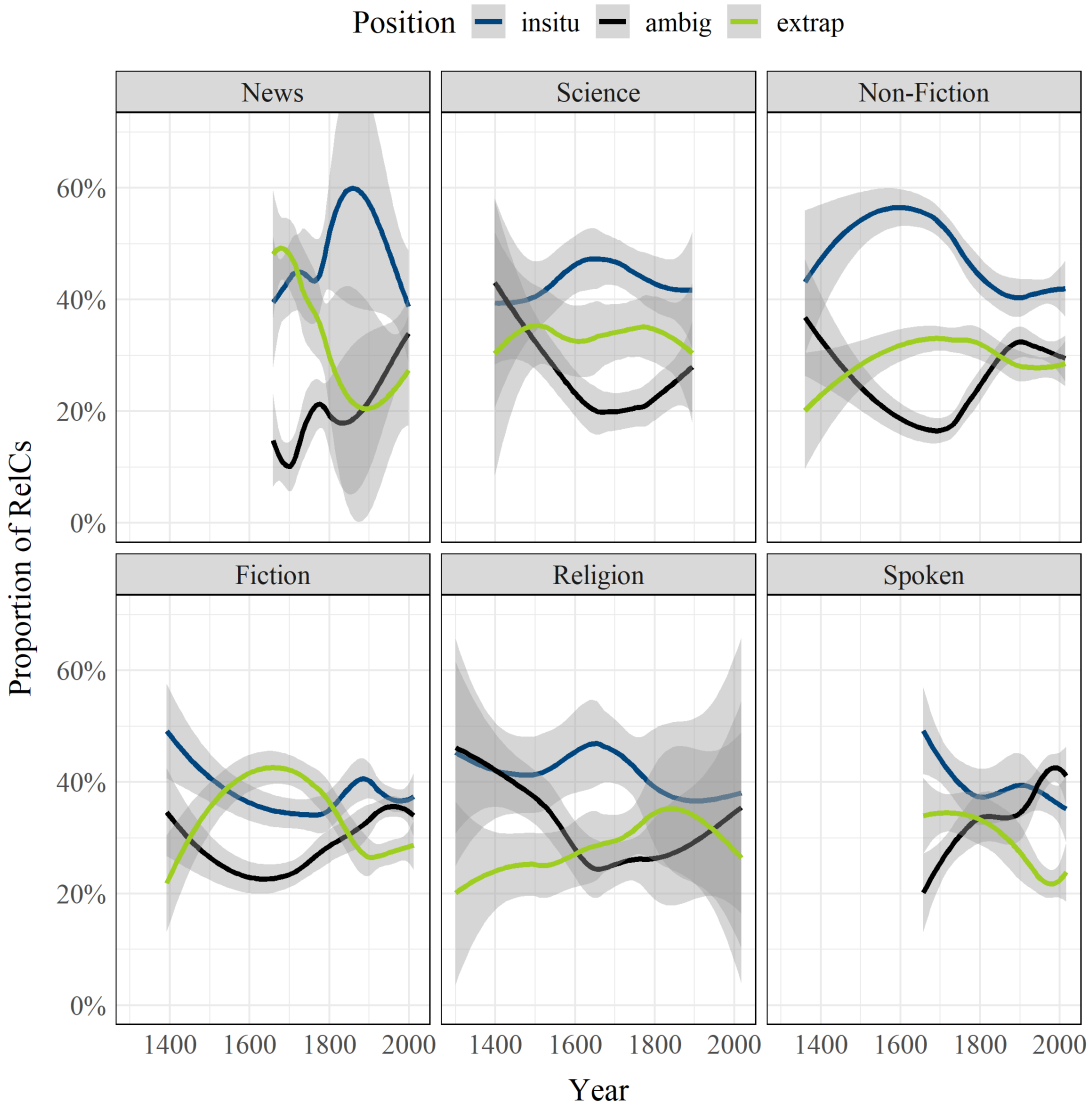


Figure 8.6.: Development of RelC positions over time in the six registers. For each year, the proportion of *in situ* (blue), ambiguous (black), and extraposed (light green) relative clauses is determined. The plot shows the local regression lines (LOESS smoothing) with a confidence interval of 0.95.

When only *in situ* and extraposed RelCs are compared (Figure 8.5, bottom), only small diachronic changes are visible. The data suggests a slight increase of extraposed RelCs until 1800, as reported by Sahel (2015), although these changes are much less pronounced when all *in situ* and extraposed RelCs are considered. After 1800, the previous increase is counter-balanced with a slight decrease. A general reduction of extraposition, like for phrases, cannot be observed for relative clauses. Instead, the proportion of extraposed RelCs compared to *in situ* RelCs seems rather stable. Overall, the number of *in situ* and extraposed RelCs changes mainly depending on the number of ambiguous cases, for which no clear position can be determined. This speaks in favor of the decision from Chapter 7 to treat ambiguous RelCs as a separate category and not exclude them from the analysis as in other studies.

Figure 8.6 shows that the distribution of RelC positions differs substantially between registers. Interestingly, it is the spoken register that shows a decrease of extraposition (although in favor of ambiguous and not *in situ* cases), whereas no such development is visible for the scientific or non-fiction texts. This observation may hint at the importance of orality for extraposition (Section 8.3.3).

Result
The proportion of extraposed vs. *in situ* RelCs remains relatively stable over time. Overall, *in situ* and extraposed RelCs show a mostly parallel development, which depends primarily on the number of ambiguous cases, with a peak of unambiguous structures around 1700.

8.3.2. Length

The second factor that is generally assumed to influence extraposition is length. Longer constituents are more likely to be extraposed (cf. Chapter 2.2.1). From a processing perspective, this may be explained with less memory strain on the middle field or shorter dependencies between the sentence brackets. Here, it means that extraposed RelCs should be longer than *in situ* RelCs. For ambiguous cases, no prior expectations exist, but it seems plausible to assume that they lie between the other two classes.

Hypothesis
Extraposed RelCs are longer on average than ambiguous and *in situ* RelCs.

Figure 8.7 shows the development of RelC length over time. While relative clauses are rather short in the oldest data sets, the average length increases to about 12.5 words in 1700 before it decreases again to about 8.5 words in present-day German. The median also reaches a plateau at around 8 words between 1700 and 1900 before it decreases again. The temporal changes are observed for all three RelC positions but are more pronounced for extraposed RelCs than for ambiguous and *in situ* cases. Interestingly, the pattern is very similar to that from the previous section: The increasing length occurs in the same time window as the decrease of ambiguous cases. This observation could speak for a higher sentence complexity in the early New High German period, with a peak in the 18th century.

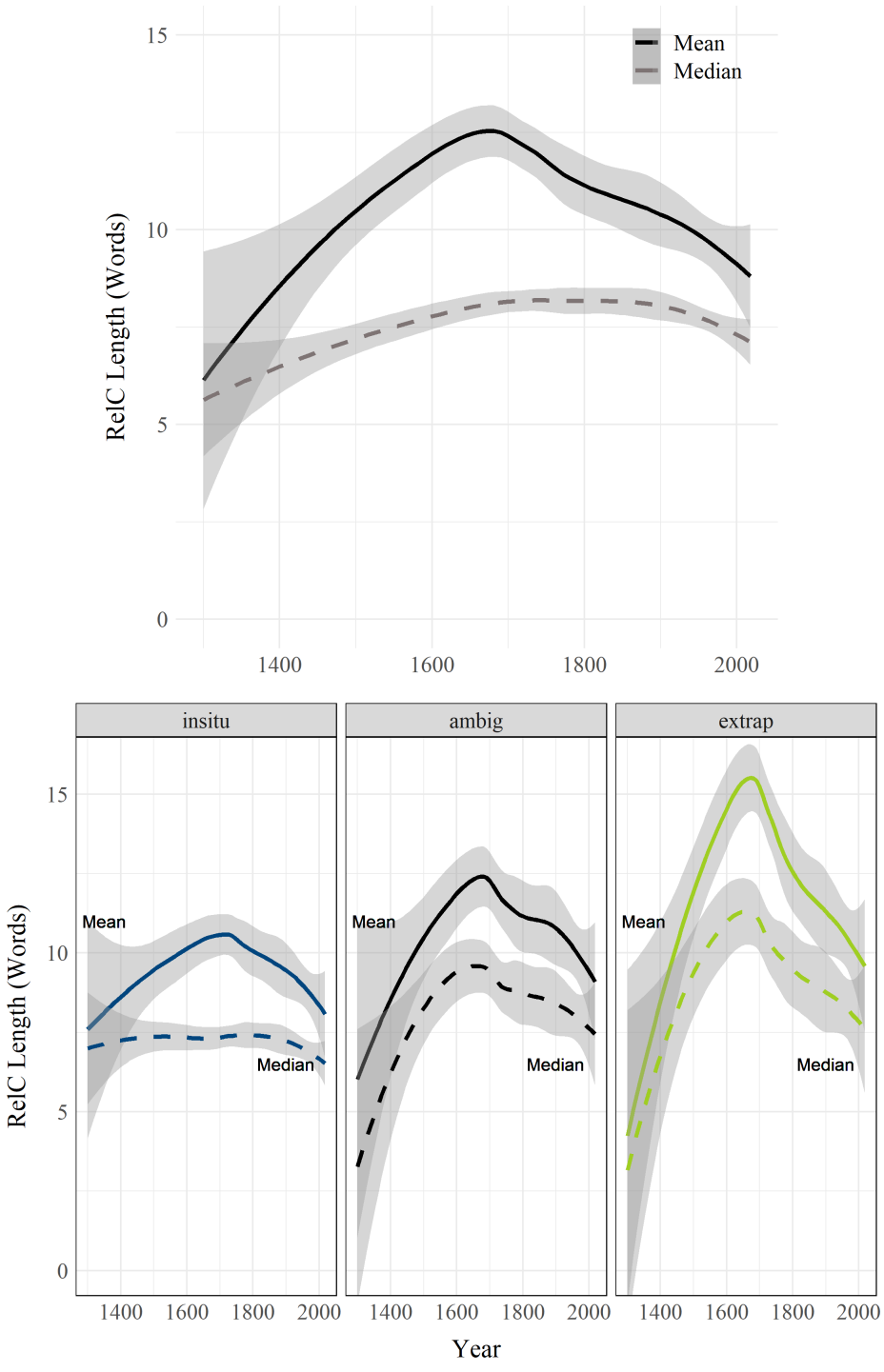


Figure 8.7.: Mean (solid line) and median (dashed line) length of relative clauses over time, for all RelCs (top) and by position (bottom) for *in situ* (blue), ambiguous (black), and extraposed (light green) RelCs. The plots show the local regression lines (LOESS smoothing) with a confidence interval of 0.95.

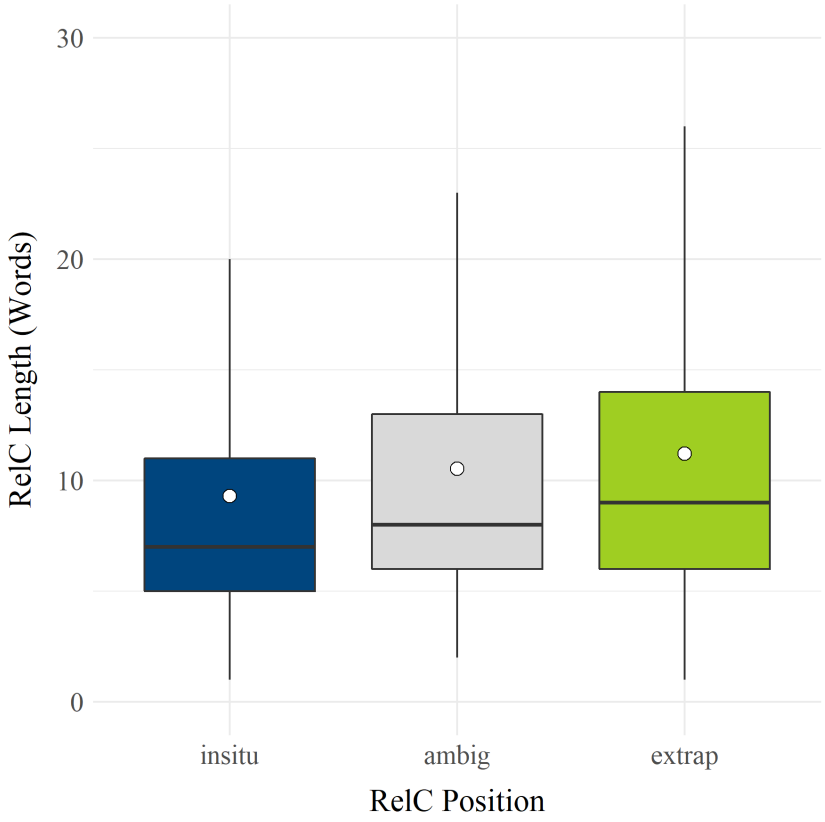


Figure 8.8.: Average length of *in situ* (blue), ambiguous (light gray), and extraposed (light green) relative clauses. The boxes show the interquartile range from first to third quartile, with a black line for median RelC length. The mean is indicated with a white dot. For better readability, outliers (i.e., longer RelCs) are not displayed here.

RelC length differs depending on orality. Orally-oriented registers contain shorter RelCs than more literate registers (mean: 7.6–9.6 vs. 11.0–11.9 words, median: 6–8 vs. 9 words; cf. Figure A.5 in the appendix; also see Figure A.6). The temporal development is roughly comparable between registers (see Figure A.7 in the appendix).

In Figure 8.8, the general relationship between RelC position and length is shown. As expected, extraposed RelCs are longer on average than *in situ* RelCs (mean: 11.2 vs. 9.3 words, median: 9 vs. 7 words). This difference is larger than the one reported by Uszkoreit et al. (1998), with 1.3 words in modern newspaper text. Ambiguous cases, which are usually discarded by studies on RelC extraposition, indeed lie between the *in situ* and extraposed RelCs (mean: 10.5 words, median: 8 words). To test whether the differences are systematic or only occurred due to chance, a one-way ANOVA is performed on a stratified random sample of 50 RelCs per position per data set, i.e.,

1,250 RelCs per position and 3,750 RelCs in total.[100] The ANOVA reveals a small main effect of position on RelC length ($F(2, 3747) = 23.14, p < 0.001, \eta^2 = 0.012$).[101] A post hoc pairwise comparison with Tukey's HSD test confirms that the differences are highly significant between all groups ($p < 0.001$). The results can be interpreted as evidence for the hypothesis that length affects the extraposition of relative clauses.

Result
Extraposed RelCs are longer than ambiguous RelCs, which are longer than *in situ* RelCs. Relative clauses are longest in conceptually literate registers and the 17th–18th century.

8.3.3. Orality

The previous analyses have already highlighted differences between registers regarding RelC length and diachronic development. However, as explained in Section 8.2.2, grouping texts into registers is a rather broad categorization. In this section, the effects of orality on extraposition are explored with our orality score as a more precise text-wise measure. Since extraposition is considered an oral phenomenon, higher proportions of extraposed RelCs are expected in texts with higher orality scores.

Hypothesis
Extraposed RelCs are more frequent in conceptually oral data.

Figure 8.9 displays the development of orality over time. Overall, orality scores in the data set decrease from 1300 to about 1700, followed by an increase until present-day German. This can partly be attributed to the distribution of registers in the data set: Early texts stem mainly from the oral-style religious data, whereas the less oral registers are primarily available for later time periods. Most registers have become less oral over time, as is often hypothesized in the literature. The only clear exception is the spoken register, which can be explained by the included data. While older data in the spoken register was meant to be recited orally (e.g., speeches, plays), the youngest data contains transcripts of spoken language, which is expected to be the most oral form of written language.

[100] Larger sample sizes are (almost) always better because they give better estimates of the (unknown) population. However, it is known that in statistical testing, large sample sizes can make even minor differences look 'significant'. To account for this, I use a smaller sample of only 3,750 RelCs instead of 563,577 RelCs for the statistical tests. The sample is stratified, which means it is balanced regarding categories (equal amounts of *in situ*/ambig/extrap) and data sets (150 RelCs per data set). Since the resulting number of RelCs is still higher than in most previous studies, I report effect sizes to indicate whether the results are only caused by the large sample size (i.e., actually meaningless) or if there is a relevant effect of the predictor variable.

[101] A one-way ANOVA is the preferred statistical test to compare the means of three groups, e.g., *in situ* vs. ambiguous vs. extraposed. It requires normally distributed data and the homogeneity of variances, which are both violated by the data set. Since the test is said to be robust against these violations, I nevertheless report the results. In addition, I also performed the non-parametric Kruskal-Wallis test ($\chi^2 = 64.268, df = 2, p < 0.001$) and a post hoc Wilcoxon test with Bonferroni adjustment, which confirm the highly significant differences between all groups.

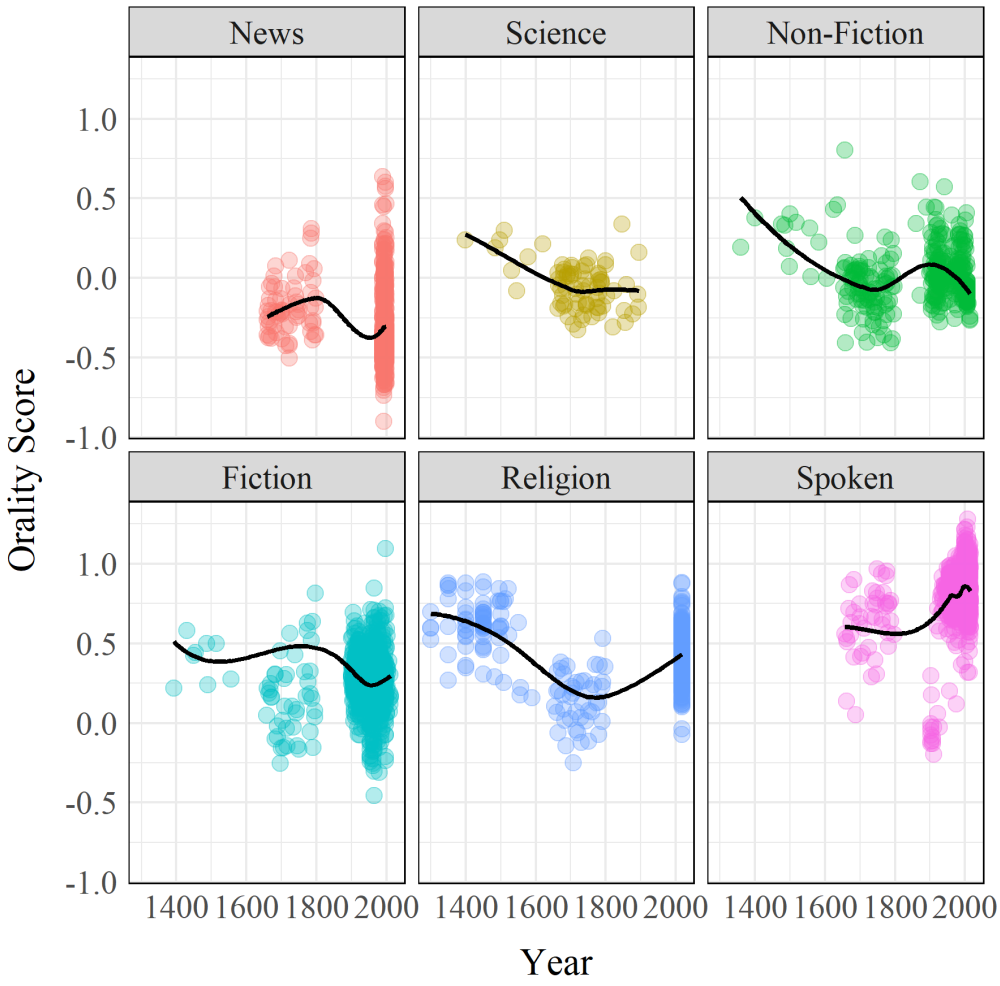


Figure 8.9.: Development of orality scores by register over time. The plot shows the individual data points and local regression lines (LOESS smoothing) over the average score per year.

However, the dip of orality scores in several registers between 1600 and 1800 also coincides with the observed increases in complexity (i.e., higher RelC length and fewer ambiguous cases) from the previous sections.

Figure 8.10 shows the proportion of *in situ*, ambiguous, and extraposed RelCs depending on the orality score. As observed for the temporal distribution (Section 8.3.1), *in situ* and extraposed RelCs again follow a parallel development. Texts with higher orality scores contain fewer *in situ* but also fewer extraposed RelCs in favor of more ambiguous cases. Only for the most oral texts, a slight increase in extraposition and a decrease in embedding can be found.

Figure 8.10.: RelC positions depending on orality. For visualization, orality scores are grouped into bins of size 0.1. For each orality bin, the proportion of *in situ* (blue), ambiguous (black), and extraposed (light green) relative clauses is determined. The plot shows the local regression lines over bins (LOESS smoothing) with a confidence interval of 0.95.

As a consequence, ambiguous RelCs are associated with slightly higher orality scores than the other groups (mean: 0.19 vs. 0.17; Figure 8.11), reflecting the parallel decrease of *in situ* and extraposed RelCs in Figure 8.10. However, the small difference in orality between RelC positions is not significant, as confirmed by a one-way ANOVA on a stratified sample of 50 RelCs per position and data set ($F(2, 3597) = 0.158, p = 0.854$).[102]

[102] This is an example of how a large sample size can make even minor differences look 'significant'. A one-way ANOVA on the whole data set returns a highly significant main effect of position ($F(2, 563149) = 749.6, p < 0.001$), with the difference lying primarily between the ambiguous RelCs and the other two groups. However, effect size $\eta^2 = 0.003$ is lower than 0.01, which means that, despite the significant results, there is actually no effect of position on orality. On the smaller stratified sample, the ANOVA does not return this meaningless effect in the first place, which speaks for using a smaller sample for significance testing.

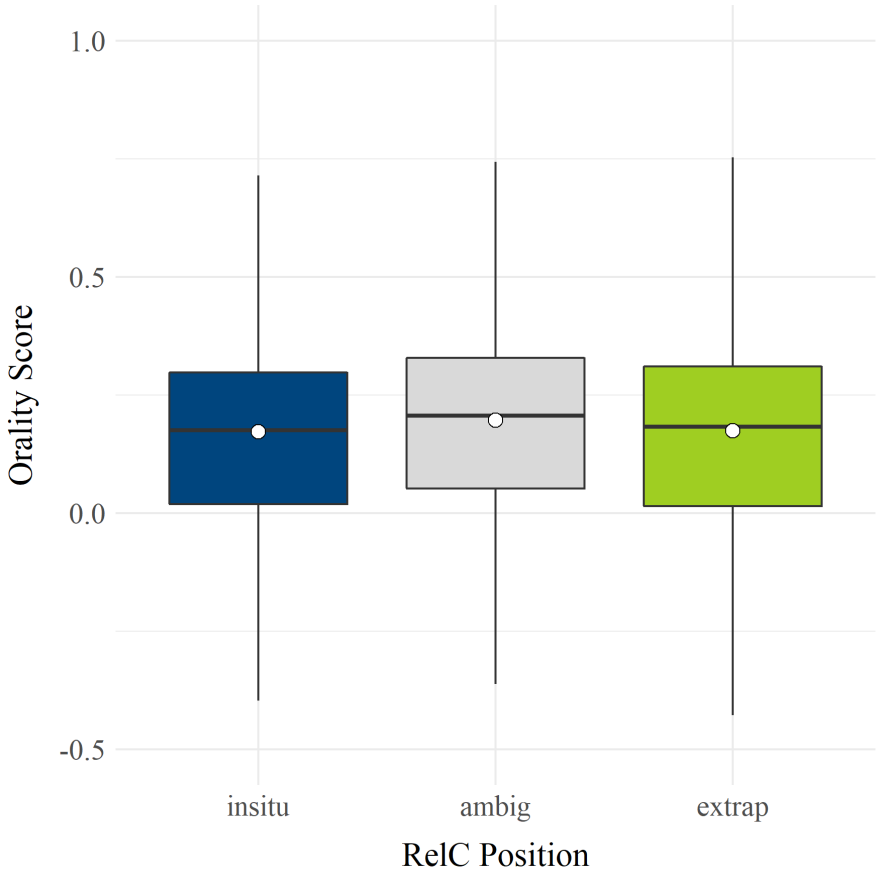


Figure 8.11.: Orality scores of *in situ* (blue), ambiguous (light gray), and extraposed (light green) relative clauses. The boxes show the interquartile range from first to third quartile, with a black line for the median orality score. The mean is indicated with a white dot. For better readability, outliers are not displayed here.

Perhaps, a general effect of orality exists only for the extraposition of phrases and not for relative clauses, which can be extraposed also in formal standard German. The observed differences in this section are obviously driven by sentence complexity, with less complex sentences and, hence, more ambiguous cases in oral language.

Result
There is no evidence for more (unambiguously) extraposed RelCs in conceptually oral data. Instead, higher orality scores are linked to more ambiguous cases with a mostly parallel decrease of *in situ* and extraposed RelCs.

8.3.4. Information Density

The fourth and final factor in the example analysis is information density. In our project, we were particularly interested in the information-theoretic perspective on extraposition and how it may ease processing by creating a better distribution of information. Voigtmann and Speyer (2021a) and Voigtmann and Speyer (2021b) looked at mean surprisal in historical scientific texts, assuming that more informative phrases and clauses are extraposed to prevent peaks of information and reduce memory strain in the middle field. We also suppose that extraposition could improve the overall information profile of the sentence in accordance with the Uniform Information Density (UID) hypothesis, but this was difficult to quantify (cf. Chapter 2.2.3). In our recent study Ortmann et al. (2022), we started experimenting with the DORM value as an objective measure of information uniformity, which will be applied to extraposition for the first time in this thesis.

However, contrary to the previous analyses, the results in this section are not directly comparable between registers and time periods. As explained in Section 8.2.3, both measures, mean RelC surprisal and DORM, depend on surprisal values that are calculated with the help of corpus-specific language models. To achieve low perplexities (i.e., good predictions), the LMs were trained separately for each data set. As a result, probabilities and the derived surprisal scores and DORM values cannot be compared between language models with different vocabulary sizes.

To illustrate why this is the case, consider a toy language with a uniform distribution of words. Given a training set with 10 different words, each word would be assigned a probability of $\frac{1}{10}$ by a unigram LM (without smoothing). If additional training data with 90 more different words from the same uniform distribution was added, each word would now be assigned a probability of $\frac{1}{100}$, simply because more training data was available. This illustrates how vocabulary size influences probability and, hence, surprisal scores, independently of the underlying distribution.

To make probabilities comparable, the smaller model from the toy example could be scaled to the same size as the larger model by simulating more training data because the underlying (uniform) distribution is known. However, the reality is more complex since natural languages do not follow any linear distribution (think of Zipf's law). And even different data sets from the same language likely follow different distributions, e.g., depending on the degree of orality. As a consequence, type-token ratios (TTR) change non-linearly with more training data, and there is no established way to simulate a larger realistic training sample and transform the probabilities of one LM so that they are comparable to those of another (larger) model. Thus, scaling or normalizing the probabilities or surprisal values, as it was done for orality scores (Section 8.2.2), neither makes sense nor is it clear what the result of such a manipulation should be.

There are two ways to address this problem if probabilities should be compared between language models:

1. Use a general language model for all data sets (or one for historical and one for modern data, both of equal size). Such a model would fit the data less well than a corpus-specific model, leading to more unreliable predictions, e.g., for genre-specific words like 'God' in general vs. in religious texts. In addition, training a general (historical) LM would first require the same

orthographic normalizations for all historical data sets to prevent creating multiple 'separate vocabularies' within one model.

2. Train all models on the same amount of (domain-specific) data. Since the available training data differs considerably between registers and time periods, this would mean cutting all models to the smallest available model size. That would significantly increase Out-of-Vocabulary rates and perplexity for all larger models, especially on data sets with high type-token ratios. If a data set has a low TTR (e.g., the Anselm data with 10%), a small model can be sufficient, whereas for data sets with high TTRs (e.g., the Tiger test data with 43%), a larger model is vital for meaningful results.

In light of the given problem, I decided to use the largest possible corpus-specific models for the example analysis in this section and only compare values within data sets for a first impression of the relationship between information density and extraposition. Future experiments should try to build comparable models for different time periods and registers and compare the resulting surprisal values between data sets to uncover even more interesting patterns.

Surprisal

In the first part of this section, the effects of mean surprisal on extraposition are explored. It is known from previous research that highly informative constituents are more likely candidates for extraposition. Hence, extraposed relative clauses should be more informative on average than embedded RelCs. In this section, I will focus only on the distinction between extraposed and *in situ* RelCs.

> **Hypothesis 1**
> Extraposed RelCs show higher mean surprisal than *in situ* RelCs.

Since surprisal values are not directly comparable between data sets, I do not report absolute values. Instead, I inspect the direction of the difference between mean surprisal values of *in situ* vs. extraposed RelCs in each data set (Eq. 8.1). A negative difference means that the extraposed RelCs are more surprising, whereas a positive difference indicates that *in situ* RelCs have a higher mean surprisal.

$$surprisal_{\text{diff}} = mean(surprisal_{\text{insitu}}) - mean(surprisal_{\text{extrap}}) \tag{8.1}$$

I experiment with bigram surprisal based on (normalized) word forms representing the lexical level and bigram surprisal based on POS tags representing the syntactic level. Tables 8.5 and 8.6 show the results for each of the 23 data sets from Section 8.2.3.

Corpus	$\text{Surpr}_{\text{diff}}$	df	t	p-value		Cohen's d	Effect size	
$\text{Gutenberg}_{\text{Fiction}}$	+	571	1.123	0.262		0.09		
$\text{Gutenberg}_{\text{Folk-Tales}}$	+	570	2.406	< 0.05	*	0.20	small	x
$\text{Gutenberg}_{\text{Non-Fiction}}$	+	601	1.229	0.219		0.10		
$\text{Gutenberg}_{\text{Speech}}$	+	623	1.542	0.124		0.12		
$\text{OPUS}_{\text{Action}}$	+	506	0.444	0.658		0.04		
$\text{OPUS}_{\text{Comedy}}$	+	518	0.304	0.761		0.03		
$\text{OPUS}_{\text{Drama}}$	-	515	-0.117	0.907		0.01		
SdeWaC	+	636	2.458	< 0.05	*	0.19		
SermonOnline	-	573	-0.964	0.335		0.08		
Tiger	+	353	0.918	0.359		0.10		
TüBa-D/S	+	173	0.272	0.786		0.04		
TüBa-D/W	+	516	4.859	< 0.001	***	0.43	small	x
TüBa-D/Z	-	557	-0.710	0.478		0.06		
Anselm	-	542	-2.466	< 0.05	*	0.21	small	✓
$\text{DTA}_{\text{Science}}$	+	624	0.881	0.379		0.07		
$\text{GerManC}_{\text{DRAM}}$	+	476	2.187	< 0.05	*	0.20	small	x
$\text{GerManC}_{\text{HUMA}}$	+	630	0.242	0.809		0.02		
$\text{GerManC}_{\text{LEGA}}$	+	693	0.996	0.320		0.08		
$\text{GerManC}_{\text{NARR}}$	+	614	0.749	0.454		0.06		
$\text{GerManC}_{\text{NEWS}}$	+	619	0.891	0.373		0.07		
$\text{GerManC}_{\text{SCIE}}$	-	630	-0.351	0.726		0.03		
$\text{GerManC}_{\text{SERM}}$	-	602	-0.950	0.343		0.08		
ReF.RUB	+	594	0.005	0.996		0.00		

Table 8.5.: Difference in mean bigram word surprisal between *in situ* and extraposed RelCs. Since surprisal values are not comparable between language models, only the direction of the difference is given. A positive difference (+) means that *in situ* relative clauses have a higher mean surprisal than extraposed RelCs. If the extraposed RelCs have a higher mean surprisal, $\text{Surpr}_{\text{diff}}$ is negative (−). For each corpus, the table also shows the results of an un-paired two samples t-test (or a Welch t-test if homogeneity of variances is violated, according to an F test) and the effect size according to Cohen's d. If $\text{Surpr}_{\text{diff}}$ is negative with a significant p-value and at least a small effect $d \geq 0.2$, the hypothesis is confirmed (✓). If $\text{Surpr}_{\text{diff}}$ is positive with a significant p-value and at least a small effect $d \geq 0.2$, the hypothesis is rejected (x). Otherwise, there is no relevant evidence for or against the hypothesis.

Corpus	$\text{Surpr}_{\text{diff}}$	df	t	p-value		Cohen's d	Effect size	
$\text{Gutenberg}_{\text{Fiction}}$	-	571	-1.681	< 0.1	.	0.14		
$\text{Gutenberg}_{\text{Folk-Tales}}$	-	569	-0.782	0.434		0.06		
$\text{Gutenberg}_{\text{Non-Fiction}}$	-	580	-1.044	0.297		0.08		
$\text{Gutenberg}_{\text{Speech}}$	-	623	-2.517	< 0.05	*	0.20	small	✓
$\text{OPUS}_{\text{Action}}$	-	506	-3.365	< 0.001	***	0.30	small	✓
$\text{OPUS}_{\text{Comedy}}$	-	518	-4.909	< 0.001	***	0.43	small	✓
$\text{OPUS}_{\text{Drama}}$	-	515	-5.033	< 0.001	***	0.44	small	✓
SdeWaC	-	639	-0.046	0.964		0.00		
SermonOnline	-	573	-5.554	< 0.001	***	0.46	small	✓
Tiger	+	353	0.830	0.407		0.09		
TüBa-D/S	-	173	-4.393	< 0.001	***	0.66	medium	✓
TüBa-D/W	+	516	0.988	0.324		0.09		
TüBa-D/Z	-	557	-1.885	< 0.1	.	0.16		
Anselm	-	542	-6.582	< 0.001	***	0.57	medium	✓
$\text{DTA}_{\text{Science}}$	-	629	-0.322	0.748		0.03		
$\text{GerManC}_{\text{DRAM}}$	+	444	0.737	0.462		0.07		
$\text{GerManC}_{\text{HUMA}}$	+	623	0.166	0.869		0.01		
$\text{GerManC}_{\text{LEGA}}$	+	685	1.733	< 0.1	.	0.13		
$\text{GerManC}_{\text{NARR}}$	+	601	2.372	< 0.05	*	0.19		
$\text{GerManC}_{\text{NEWS}}$	+	598	2.297	< 0.05	*	0.18		
$\text{GerManC}_{\text{SCIE}}$	+	618	2.208	< 0.05	*	0.17		
$\text{GerManC}_{\text{SERM}}$	-	599	-1.752	< 0.1	.	0.14		
ReF.RUB	-	594	-1.495	0.136		0.12		

Table 8.6.: Difference in mean bigram POS surprisal between *in situ* and extraposed RelCs. Since surprisal values are not comparable between language models, only the direction of the difference is given. A positive difference (+) means that *in situ* relative clauses have a higher mean surprisal than extraposed RelCs. If extraposed RelCs have a higher mean surprisal, $\text{Surpr}_{\text{diff}}$ is negative (−). For each corpus, the table also shows the results of an unpaired two-samples t-test (or a Welch t-test if homogeneity of variances is violated, according to an F test) and the effect size according to Cohen's d. If $\text{Surpr}_{\text{diff}}$ is negative with a significant p-value and at least a small effect $d \geq 0.2$, the hypothesis is confirmed (✓). If $\text{Surpr}_{\text{diff}}$ is positive with a significant p-value and at least a small effect $d \geq 0.2$, the hypothesis is rejected (x). Otherwise, there is no relevant evidence for or against the hypothesis.

For POS-based surprisal, the majority of data sets show the expected direction with higher mean surprisal for extraposed RelCs compared to *in situ* RelCs. For word forms, the opposite is true. To test whether the differences are significant, I conducted paired two-sample t-tests for each data set (or Welch t-tests if homogeneity of variances is violated, according to an F test). Most differences turn out to be negligible. For word form surprisal, only the Anselm data shows a small effect in favor of the hypothesis (Cohen's $d \geq 0.2$), while three data sets show a small negative effect ($\text{Gutenberg}_{\text{Folk-Tales}}$, TüBa-D/W, and $\text{GerManC}_{\text{DRAM}}$).

On the syntactic level, i.e., based on POS bigram surprisal, seven data sets show a small to medium effect in favor of the hypothesis (Cohen's $d \geq 0.2$ and $d \geq 0.5$, respectively). Interestingly, all these data sets belong to the two orally oriented registers Religion and Spoken: $\text{Gutenberg}_{\text{Speech}}$, the OPUS data sets, SermonOnline, TüBa-D/S, and the Anselm corpus. Perhaps, this result hints at the higher relevance of processing costs in oral language.

> **Result 1**
> In conceptually oral data, extraposed RelCs show higher syntactic surprisal than *in situ* RelCs. In conceptually literate registers and on the lexical level, there is no evidence for a significant relationship between mean RelC surprisal and extraposition.

DORM

Besides moving RelCs with high mean surprisal to the post-field to prevent peaks of information in the middle field, the movement could also smooth the overall information profile of the sentence compared to leaving the RelC *in situ*.

> **Hypothesis 2**
> Sentences with extraposition have a more uniform information profile than variant sentences in which the extraposition was undone.

Again, I look at the difference in DORM values between the original sentences with extraposition and the variant sentences with *in situ* RelCs (Eq. 8.2). I do not report absolute values because they are not directly comparable between language models. A negative difference means that the original sentence with extraposition has a more uniform information profile. A positive difference indicates that leaving the relative clause *in situ* would have resulted in a smoother information distribution.

$$DORM_{\text{diff}} = DORM_{\text{orig}} - DORM_{\text{variant}} \tag{8.2}$$

I experimented with bigram surprisal based on (normalized) word forms representing the lexical level and bigram surprisal based on POS tags representing the syntactic level. In addition, I calculated DORM values based on word surprisal and constituent surprisal. Intuitively, the calculation based on constituents seems more meaningful because the choice of extraposing a complete constituent or leaving it *in situ* is unlikely to depend only on single words (e.g., producing the verb in the right bracket vs. the relative pronoun first).

Corpus	$DORM_{diff}$	df	t	p-value		Cohen's d	Effect size	
$Gutenberg_{Fiction}$	-	238	-0.307	0.759		0.02		
$Gutenberg_{Folk\text{-}Tales}$	+	239	2.092	< 0.05	*	0.14		
$Gutenberg_{Non\text{-}Fiction}$	+	235	3.057	< 0.01	**	0.20		
$Gutenberg_{Speech}$	+	230	2.273	< 0.05	*	0.15		
$OPUS_{Action}$	+	245	0.578	0.564		0.04		
$OPUS_{Comedy}$	+	240	0.055	0.956		0.00		
$OPUS_{Drama}$	+	238	0.306	0.760		0.02		
SdeWaC	+	220	1.657	< 0.1	.	0.11		
SermonOnline	+	232	1.429	0.154		0.09		
Tiger	+	167	1.059	0.291		0.08		
TüBa-D/S	-	84	-0.901	0.370		0.10		
TüBa-D/W	+	241	2.002	< 0.05	*	0.13		
TüBa-D/Z	+	249	0.165	0.869		0.01		
Anselm	-	187	-6.154	< 0.001	***	0.45	small	✓
$DTA_{Science}$	-	196	-0.086	0.932		0.01		
$GerManC_{DRAM}$	+	188	1.845	< 0.1	.	0.13		
$GerManC_{HUMA}$	+	203	2.362	< 0.05	*	0.16		
$GerManC_{LEGA}$	-	195	-0.168	0.866		0.01		
$GerManC_{NARR}$	+	202	2.136	< 0.05	*	0.15		
$GerManC_{NEWS}$	+	217	0.151	0.880		0.01		
$GerManC_{SCIE}$	+	203	3.565	< 0.001	***	0.25	small	x
$GerManC_{SERM}$	+	212	0.432	0.666		0.03		
ReF.RUB	+	193	1.489	0.138		0.11		

Table 8.7.: Difference in DORM values between original and variant sentences based on mean bigram word form surprisal of constituents. Since surprisal values are not comparable between language models, only the direction of the difference is given. A negative difference (−) means that the original sentence with extraposition has a lower DORM value (i.e., a smoother information profile) than the variant sentence. If the variant sentence would be smoother, the $DORM_{diff}$ value is positive (+). For each corpus, the table also shows the results of a one-sample t-test and the effect size according to Cohen's d. If $DORM_{diff}$ is negative with a significant p-value and at least a small effect $d \geq 0.2$, the hypothesis is confirmed (✓). If $DORM_{diff}$ is positive with a significant p-value and at least a small effect $d \geq 0.2$, the hypothesis is rejected (x). Otherwise, there is no relevant evidence for or against the hypothesis.

Corpus	$DORM_{diff}$	df	t	p-value		Cohen's d	Effect size	
$Gutenberg_{Fiction}$	-	238	-5.706	< 0.001	***	0.37	small	✓
$Gutenberg_{Folk\text{-}Tales}$	-	239	-8.576	< 0.001	***	0.55	medium	✓
$Gutenberg_{Non\text{-}Fiction}$	-	235	-6.702	< 0.001	***	0.44	small	✓
$Gutenberg_{Speech}$	-	230	-5.804	< 0.001	***	0.38	small	✓
$OPUS_{Action}$	-	245	-7.789	< 0.001	***	0.50	small	✓
$OPUS_{Comedy}$	-	240	-8.531	< 0.001	***	0.55	medium	✓
$OPUS_{Drama}$	-	238	-8.459	< 0.001	***	0.55	medium	✓
SdeWaC	-	220	-6.767	< 0.001	***	0.46	small	✓
SermonOnline	-	232	-6.903	< 0.001	***	0.45	small	✓
Tiger	-	167	-6.442	< 0.001	***	0.50	small	✓
TüBa-D/S	-	84	-5.510	< 0.001	***	0.60	medium	✓
TüBa-D/W	-	241	-11.554	< 0.001	***	0.74	medium	✓
TüBa-D/Z	-	249	-6.292	< 0.001	***	0.40	small	✓
Anselm	-	187	-4.516	< 0.001	***	0.33	small	✓
$DTA_{Science}$	-	196	-4.499	< 0.001	***	0.32	small	✓
$GerManC_{DRAM}$	-	188	-4.386	< 0.001	***	0.32	small	✓
$GerManC_{HUMA}$	-	203	-5.602	< 0.001	***	0.39	small	✓
$GerManC_{LEGA}$	-	195	-4.619	< 0.001	***	0.33	small	✓
$GerManC_{NARR}$	-	202	-7.753	< 0.001	***	0.54	medium	✓
$GerManC_{NEWS}$	-	217	-6.048	< 0.001	***	0.41	small	✓
$GerManC_{SCIE}$	-	203	-6.167	< 0.001	***	0.43	small	✓
$GerManC_{SERM}$	-	212	-5.644	< 0.001	***	0.39	small	✓
ReF.RUB	-	193	-2.381	< 0.05	*	0.17		

Table 8.8.: Difference in DORM values between original and variant sentences based on mean bigram POS surprisal of constituents. Since surprisal values are not comparable between language models, only the direction of the difference is given. A negative difference (−) means that the original sentence with extraposition has a lower DORM value (i.e., a smoother information profile) than the variant sentence. If the variant sentence would be smoother, the $DORM_{diff}$ value is positive (+). For each corpus, the table also shows the results of a one-sample t-test and the effect size according to Cohen's d. If $DORM_{diff}$ is negative with a significant p-value and at least a small effect $d \geq 0.2$, the hypothesis is confirmed (✓). If $DORM_{diff}$ is positive with a significant p-value and at least a small effect $d \geq 0.2$, the hypothesis is rejected (x). Otherwise, there is no relevant evidence for or against the hypothesis.

Tables 8.7 and 8.8 show the results for $\text{DORM}_{\text{diff}}$ based on mean constituent surprisal for word forms and POS tags. For completeness, the results for DORM calculations based on word surprisal can be found in the appendix (Tables A.13 and A.14).

Overall, the results are similar to those for surprisal above. Only four data sets show a negative $\text{DORM}_{\text{diff}}$ value on the lexical level. A one-sample t-test reveals that the values are not significantly different from zero except for one data set with a small negative ($\text{GerManC}_{\text{Science}}$) and one with a small positive effect (Anselm). In contrast, all data sets show a significantly negative $\text{DORM}_{\text{diff}}$ value on the syntactic level (i.e., based on POS tags). Except for the ReF.RUB corpus, there is a small to medium effect for each data set (Cohen's $d \geq 0.2$ and $d \geq 0.5$, respectively). In other words, original sentences indeed have more uniform information profiles on the syntactic level than their variants.

Possible explanations for the differences between word forms and POS-based models could be that the phenomenon of extraposition is syntactic in nature, and effects are only visible on the syntactic level. Also, the word-based language models could not be powerful enough due to data sparsity. If too many bigrams are unseen, this may obscure existing effects, which is less problematic for POS-based models with their much smaller vocabulary size.

Result 2
Sentences with RelC extraposition exhibit a more uniform information profile on the syntactic level than variant sentences in which the RelCs are placed *in situ*.

8.4. Discussion

In this chapter, I have exemplarily applied the methods from this thesis to illustrate their usefulness for linguistic studies. I selected 25 large data sets of modern and historical German from 1300 to 2018, which were automatically annotated with topological fields, constituency trees, and the extraposition of relative clauses. The resulting database includes more than 560k relative clauses from different registers and time periods. For the example analysis, each text was analyzed concerning its conceptual orality and automatically rated with our orality score. In addition, large n-gram language models were created based on word forms and POS tags. The LMs were used to calculate information-theoretic measures, in particular, mean RelC surprisal and DORM. For the latter, a corpus of variants was generated to compare the information profiles of original sentences and variant sentences in which the extrapositions have been undone.

Using the automatically created resources, I explored the effects of four factors that are said to influence extraposition: time, length, orality, and information density. The example analysis revealed that there is neither evidence for a general increase or reduction of extraposed RelCs over time nor for a general effect of orality on RelC extraposition. Instead, extraposed and *in situ* RelCs turned out to behave very similarly, in opposition to the number of RelCs in the ambiguous position at the boundary of middle field and post-field. For RelC length, the analysis confirmed a significant difference between RelCs in the three possible positions. As hypothesized, extraposed RelCs are longer than ambiguous RelCs, which are longer than *in situ* RelCs. In the final analysis, I observed an

effect of information density on the syntactic level, with higher mean surprisal of extraposed RelCs in oral registers and more uniform information profiles in original sentences with extraposition compared to variant sentences with *in situ* RelCs.

The example analyses have given a good first impression of patterns in the data and revealed interesting starting points for future studies. For example, the observed peak of complexity with long RelCs, low orality scores, and few ambiguous cases around 1700 could be investigated more closely to identify the linguistic factors that are involved in the changes before and after the peak. Also, the observed differences between registers regarding the temporal development of extraposition or the changes of orality in specific registers could be explored in more detail. Furthermore, additional factors from Chapter 2.2.4 could be integrated into the analysis, e.g., the distance between RelCs and their antecedents.

With this thesis, I have laid the foundation for such studies, demonstrating the benefits of computational methods for creating large amounts of annotated data in a very short time and without tremendous manual effort. In the next chapter, the contributions will be summarized and complemented with suggestions for future improvements.

CHAPTER 9

Conclusion

In this thesis, the automatic syntactic analysis of modern and historical German was explored. Traditionally, studies on historical German and diachronic syntactic change are based on small, manually annotated data samples. Such qualitative investigations provide valuable insights, but they lack the generalizability and statistical significance that quantitative approaches can offer. With my work, I wanted to adopt a pragmatic perspective and create tools that are of practical value for such linguistic studies by helping to compile large annotated data sets without the usual need for costly and time-consuming manual labor. Along the way, I explored different types of syntactic annotations and created models and data sets that previously did not exist for (historical) German. This pioneering work can now serve as a foundation for future studies on various syntactic phenomena – and for computational linguistic work with historical German in general.

Due to the high degree of variation and the ubiquitous lack of data and models, historical language poses several challenges to the application of standard computational methods. Previous studies on the automatic syntactic analysis of historical German have found different ways to deal with these conditions. Approaches range from developing rule-based methods that heavily depend on expert knowledge but do not require training data to transferring modern statistical models to historical language. However, the accuracy of such approaches often remains unclear because annotations are not evaluated, and tools and models are not made available. This prevents the enhancement of created resources as well as the application in practical contexts. In contrast, I have developed flexible, probabilistic methods for modern and historical German that can be applied in future projects and require only basic POS annotations as input. In addition, I have emphasized the thorough evaluation of each annotation step, creating gold-standard data to guarantee maximum transparency of the obtained results. The code and created models and data sets from this thesis are made freely available for reuse and future enhancement. The developed methods are released as the CLASSIG pipeline (**C**omputational **L**inguistic **A**nalysis of **S**yntactic **S**tructures **I**n **G**erman) at https://github.com/rubcompling/classig-pipeline. Data sets, models, and evaluation results are provided for download at https://github.com/rubcompling/classig-data and https://doi.org/10.5281/zenodo.7180973. The following paragraphs summarize the contributions of this thesis and provide suggestions for future improvement.

Summary of this Thesis

The overarching goal of the thesis was to develop methods for the automatic identification of extraposition in modern and historical German. Extraposition was defined as the 'movement' of constituents from the middle field (or sometimes the pre-field) of the sentence to the post-field. Consequently, identifying extraposed constituents requires (i) the analysis of topological fields and (ii) the recognition of candidates for extraposition.

I compiled training and evaluation data from different registers and time periods (Chapter 3) and started with the identification of topological fields (Chapter 5). In a pilot study, I tested different approaches for the recognition of sentence brackets, and trained parser models for a general topological field analysis. Since no historical training data was available, the models were transferred from modern newspaper data to other registers and historical German, using the shared level of POS tags as input. For modern German, this approach yields overall F_1-scores of 92%–97%, while results for historical German range between 85% and 93%. The best results are achieved for the sentence brackets, followed by middle and pre-fields. Post-fields, which are particularly relevant for the recognition of extraposition, are recognized less reliably with 64%–87%.

During my experiments, I noticed that the application of traditional evaluation metrics leads to undesirable effects when applied to labeled spans like topological fields. In response, I developed a new evaluation method called *FairEval* (Chapter 4), which provides more meaningful results than traditional metrics by preventing double penalties for overlapping spans. Simultaneously, it enables a fine-grained error analysis for a detailed understanding of the underlying error causes. In this thesis, all evaluations were performed with the new method for the most insightful analysis. Results according to traditional evaluation metrics are provided in the appendix for comparison.

After the analysis of topological fields, I worked on the recognition of candidates for extraposition (Chapter 6). I started with a study on chunking before proceeding to constituency analysis. My focus was on constituents that are expected to show at least some variability regarding their position in the middle field and post-field, namely noun phrases, prepositional phrases, adjective and adverb phrases, and (attributive) relative clauses. I compared state-of-the-art neural sequence labeling to unlexicalized probabilistic parsing. While sequence labeling yields very good results for chunking modern German with F_1-scores of 93%–97% and chunking historical German with 90%–94%, parsing creates better results for the recognition of more complex constituents. I trained models on different modern and historical treebanks and found that phrases can be recognized with F_1-scores of 86%–91% in modern German and 72%–85% in historical data. The identification of relative clauses is even more accurate due to their distinctive structure, with F_1-scores of 91%–96% in modern and 76%–89% in historical German.

Building on these results, the automatic analysis of extraposition was explored (Chapter 7). First, I considered the base position of extraposed elements, i.e., the original, unmarked position in the middle field, and implemented the automatic identification of antecedents for attributive constituents. I focused on attributive relative clauses and found that simple heuristics are sufficient to reliably determine their base position. Given the topological field analysis and information about the base position, candidates for extraposition were then labeled as *in situ* or extraposed (or ambigu-

ous, in the case of relative clauses at the boundary of middle field and post-field). The evaluation revealed that the analysis is not reliable for phrases yet. While *in situ* phrases are recognized with high accuracy, the highest F_1-scores for extraposed phrases are reached for PPs with 43%–82%. For relative clauses, results are more accurate, with F_1-scores of 77% to 96% for modern German and 67% to 85% in historical data.

The final chapter (Chapter 8) focused on the extraposition of relative clauses and demonstrated the usefulness of the developed methods with an example application. Large modern and historical data sets were automatically annotated with topological fields, constituency trees, and extraposition, yielding a database of over 560k relative clauses from different registers and time periods. With our orality score and several trained n-gram language models, I exemplarily explored the effects of different factors on RelC extraposition. The quantitative analyses revealed interesting patterns in the data, confirming hypotheses about the effects of length and information density on RelC extraposition but disproving the expected influence of time and orality. The example application thus illustrated the benefits of computational methods for linguistic studies by creating large amounts of annotated data in a very short time to verify existing hypotheses and spot interesting trends in the data without tremendous manual effort.

Discussion and Future Work

The explorations in this thesis have shown what is and is not (yet) possible with the available resources, highlighting opportunities for improvement. In the course of our project, the first hindrance was the multitude of data formats and tagsets for (historical) German. If projects decide to develop their own formats, converters to at least one standard format (e.g., CoNLL, TEI) should be provided. Also, data sets should be enriched at least with basic linguistic annotations like sentence and token boundaries, POS tags, and normalized word forms or lemmas, which are a prerequisite for almost all computational linguistic approaches. If custom tagsets are used, they should be accompanied by an official mapping to a standard tagset such as the STTS.

The experiments on topological field analysis have also revealed a startling lack of annotated historical data. While (by now) there is at least one large constituency treebank of Early New High German, which can be used to train and evaluate statistical models, the highly relevant topological field structure has not received enough attention yet. Especially fields that are infrequent in the existing modern data suffer from low recall, which directly affects the accuracy of subsequent analyses like the recognition of phrasal extraposition. It would be desirable if future studies created new training data for this widely used syntactic framework – perhaps in a semi-automatic manner using the models from this thesis – to improve the analysis of modern non-standard registers and historical data.

The methods in this thesis were developed primarily based on POS tags. If good uniform normalizations are available for all data sets, follow-up experiments should explore topological field analysis and constituency parsing with lexicalized, potentially neural models. Combined with additional training data, these steps could improve the results for phrasal extraposition and enable a reliable automatic analysis of influencing factors comparable to the example analysis of RelC ex-

traposition. It can be expected that several of the hypotheses that were rejected for relative clauses are more relevant to phrases, e.g., a diachronic decrease of extraposition or influences of orality.

The data that was created for this thesis could also be used for further analyses of interactions between different factors, changes within registers, or interesting phenomena at specific time points. Additional factors could be added to investigate their effects, validate qualitative assumptions, and reveal new patterns in the data that only become visible from a quantitative perspective.

In the final chapter, I used corpus-specific language models of very different sizes, which did not allow for direct comparisons between data sets. It would be interesting to enhance the information-theoretic analyses by creating comparable models, possibly beyond bigrams, and study the effects for different time periods, registers, and data sets. The example analysis was also the first time the new DORM measure was applied to investigate the influence of extraposition on the information profile of the sentence and quantify effects that previously were difficult to grasp. Since I found very different results depending on the underlying surprisal values (POS vs. word forms, tokens vs. constituents), the advantages and disadvantages of different calculation methods should be inspected in more detail.

When I began this project, there was only little previous work on the automatic syntactic analysis of historical German. With this thesis, I hope to have laid a foundation for future work in this field, encouraging the application of computational methods to expand the possibilities of linguistic studies beyond traditional limits.

Appendix A

Additional Material

A.1. Data

Mercurius	STTS
$!	$.
$:	$.
$;	$.
$?	$.
--	XY
KOMPE	Tag of the following word
NNE	NN
PROAV	PAV
UNKNOWN	XY
VVPG	ADJD

Table A.1.: Mapping rules used to derive STTS POS tags (Schiller et al. 1999) from the custom POS tags in the Mercurius corpus (Demske 2005). Tags that are not listed in the table remain unchanged.

ReF.UP	STTS
--	XY
$!	$.
$(	$(
$,	$,
$.	$.
$:	$.
$;	$.
$?	$.
$MK	$,
$MSBI	$.
$QL	$(
$QR	$(
ADJA	ADJA
ADJD	ADJD
ADJN	ADJD
ADJS	ADJA
ADJV	ADJD
ADV	ADV
APPO	APPO
APPR	APPR
APPRDARTB	APPRART
APZR	APZR
AVD	ADV
AVNEG	ADV
AVREL	ADV
AVW	PWAV
CARD	CARD
DARTB	ART
DARTU	ART
DDEM	PDAT
DINDEF	PIAT
DPOS	PPOSAT
DW	PWAT

ReF.UP	STTS
FM	FM
ITJ	ITJ
KOKOM	KOKOM
KON	KON
KOUI	KOUI
KOUS	KOUS
NA	NN
NE	NE
PAVAP	ADV
PAVD	ADV
PAVDAP	PAV
PAVREL	ADV
PAVRELAP	PAV
PAVW	PWAV
PAVWAP	PWAV
PDEM	PDS
PINDEF	PIS
PPER	PPER
PPOS	PPOSS
PRELAT	PRELAT
PRELS	PRELS
PRF	PRF
PTKA	PTKA
PTKANT	PTKANT
PTKNEG	PTKNEG
PTKREL	ADV
PTKVZ	PTKVZ
PTKZU	PTKZU
PW	PWS
PWAV	PWAV
SPELL	XY
TRUNC	TRUNC

ReF.UP	STTS
VAFIN	VAFIN
VAIMP	VAIMP
VAINF	VAINF
VAINFS	VAINF
VAPP	VAPP
VAPPA	ADJA
VAPPD	ADJD
VAPPN	VAPP
VAPSA	ADJA
VAPSD	ADJD
VAPSN	VAPP
VAPSS	NN
VMFIN	VMFIN
VMIMP	VMIMP
VMINF	VMINF
VMINFS	NN
VMPP	VMPP
VVFIN	VVFIN
VVIMP	VVIMP
VVINF	VVINF
VVINFS	NN
VVIZU	VVIZU
VVPP	VVPP
VVPPA	ADJA
VVPPD	ADJD
VVPPN	VVPP
VVPPS	NN
VVPS	VVPP
VVPSA	ADJA
VVPSD	ADJD
VVPSN	VVPP
VVPSS	NN

Table A.2.: Mapping rules used to derive STTS POS tags (Schiller et al. 1999) from the custom POS tags in the ReF.UP corpus (Demske 2019).

HIPKON	STTS
$_	$. $, $(
ADJ	ADJA
ADJD	ADJD
ADJO	ADJA
ADJOD	CARD
ADJOS	NN
ADJS	ADJA
ADV	ADV
ADVNEG	ADV
ADVREL	ADV
APPO	APPO
APPR	APPR
APPRART	APPRART
APZR	APZR
AVD	ADV
CARD	CARD
CARDD	CARD
DD	PDAT
DDA	ART
DDREL	PRELS
DDS	PDS
DDSREL	PRELS
DI	PIAT
DIA	ART
DINEG	PIAT
DIS	PIS
DPOS	PPOSAT
DPOSS	PPOSS
FM	FM
ITJ	ITJ
KOKOM	KOKOM
KON	KON
KOUS	KOUS

HIPKON	STTS
NA	NN
NE	NE
PAV	PAV
PAVREL	ADV
PI	PIS
PINEG	PIS
PPER	PPER
PRF	PRF
PTKA	PTKA
PTKNEG	PTKNEG
PTKREL	ADV
PTKVZ	PTKVZ
PTKZU	PTKZU
PW	PWS
PWAV	PWAV
PWAVREL	PWAV
PWREL	PRELS
ſ	$. $,
VAFIN	VAFIN
VAINF	VAINF
VAPP	VAPP
VMFIN	VMFIN
VMIMP	VMIMP
VMINF	VMINF
VN	VVPP
VVFIN	VVFIN
VVIMP	VVIMP
VVINF	VVINF
VVPP	VVPP
VVPPA	ADJA
VVPS	VVPP
VVPSD	ADJD
VVPSS	NN

Table A.3.: Mapping rules used to derive STTS POS tags (Schiller et al. 1999) from the customized POS tagset of the HIPKON corpus (Coniglio et al. 2014). For punctuation symbols ($_), the appropriate STTS tag $., $, or $(is determined based on the surface form of the token. In all other cases, the replacement is independent of the particular words and contexts. There is no original POS annotation available for 36 tokens from 3 sentences. The correct STTS tags for those tokens were added manually.

Lemma-POS	Token-POS	STTS
$(	$(	$(
$_	$_	$. or $,
ADJ	ADJA	ADJA
VVPP	ADJA	ADJA
VVPS	ADJA	ADJA
ADJ	ADJD	ADJD
VVPP	ADJD	ADJD
VVPS	ADJD	ADJD
ADJ	ADJN	ADJD
VVPP	ADJN	ADJD
ADJ	ADJS	ADJA
VVPP	ADJS	ADJA
VVPS	ADJS	ADJA
AP	APPO	APPO
AP	APPR	APPR
AP	APPRDDART	APPRART
AP	APZR	APZR
ADJ	AVD	ADV
AVD	AVD	ADV
VVPP	AVD	ADV
VVPS	AVD	ADV
AVG	AVG	PWAV
AVD	AVNEG	ADV
AVW	AVW	PWAV
CARD	CARDA	CARD
CARD	CARDD	CARD
CARD	CARDN	CARD
CARD	CARDS	CARD
DD	DDA	PDAT
DD	DDART	ART
DD	DDD	PDS
DD	DDN	PDAT
DD	DDS	PDS
DI	DIA	PIAT
DI	DIART	ART
DI	DID	PIS
DI	DIN	PIAT
DI	DIS	PIS
DI	DNEGA	PIAT
DI	DNEGS	PIS
DPOS	DPOSA	PPOSAT
DPOS	DPOSD	PPOSS
DPOS	DPOSN	PPOSAT
DPOS	DPOSS	NN
DD	DRELS	PRELS or PRELAT
DW	DWA	PWAT
DW	DWS	PWS
FM	FM	FM
ITJ	ITJ	ITJ
KO	KO*	KOUS
KO	KOKOM	KOKOM
KO	KON	KON
KO	KOUI	KOUI
KO	KOUS	KOUS
NA	NA	NN
VAPP	NA	NN
VVINF	NA	NN
VVPP	NA	NN
VVPS	NA	NN
NE	NE	NE
AP	PAVAP	APPR
AVD	PAVAP	PAV
AVD	PAVD	ADV
AVDAP	PAVDAP	PAV
AVDAP	PAVRELAP	PAV
AVW	PAVW	PWAV
AVDAP	PAVWAP	PWAV
PG	PG	PWS
NA	PI	PIS
PI	PI	PIS
PW	PI	PIS
PI	PNEG	PIS
PPER	PPER	PPER
PPER	PRF	PRF
PRF	PRF	PRF
PTK	PTKA	PTKA
PTK	PTKANT	PTKANT
PI	PTKNEG	PTKNEG
PTK	PTKNEG	PTKNEG
PTK	PTKREL	PRELS
AVD	PTKVZ	PTKVZ
PTK	PTKZU	PTKZU
PW	PW	PWS or PWAT
SPELL	SPELL	XY
SYM	SYM	XY
NA	TRUNC	TRUNC
UNK	UNK	XY
VA	VAFIN	VAFIN
VA	VAIMP	VAIMP
VA	VAINF	VAINF
VA	VAPP	VAPP
VA	VAPS	ADJD
VM	VMFIN	VMFIN
VM	VMIMP	VMIMP
VM	VMINF	VMINF
VM	VMPP	VMPP
VM	VMPS	ADJD
VV	VVFIN	VVFIN
VV	VVIMP	VVIMP
VV	VVINF	VVINF
VV	VVPP	VVPP
VV	VVPS	ADJD

Table A.4.: Mapping rules used to derive STTS tags from the HiTS tags (Dipper et al. 2013) in the ReF.RUB corpus (Wegera et al. 2021), each of which consists of one tag for the lemma and one for the token. Three rules are context dependent: $_>$_ is mapped to $. or $, depending on the symbol and the punc annotation. DD>DRELS is mapped to PRELAT if the following STTS tag is ADJA or NN and to PRELS otherwise. Similarly, PW>PW is mapped to PWAT if the next tag is ADJA/NN and to PWS otherwise.

A.2. Topological Fields

State \ Event	KOUI	KOUS	OTH	PREL	PTK	PW	VFin	VImp	VNonFin
C1	-	C1, LK	MF1	C1, LK	RK2, RK	C1, LK	RK3, RK	-	RK2, RK
C2	C2, LK	-	MF2	-	RK3, RK	-	-	-	RK3, RK
C3	-	C3, LK	MF4	C3, LK	RK4, RK	C3, LK	RK5, RK	-	RK4, RK
C4	C4, LK	-	MF5	-	RK5, RK	-	-	-	RK5, RK
C5	-	C5, LK	MF6	C5, LK	RK6, RK	C5, LK	RK7, RK	-	RK6, RK
C6	C6, LK	-	MF7	-	RK7, RK	-	-	-	RK7, RK
LK	-	-	MF3	-	RK1, RK	-	-	-	RK1, RK
MF1	-	-	MF1	-	RK2, RK	-	RK3, RK	-	RK2, RK
MF2	-	-	MF2	-	RK3, RK	-	-	-	RK3, RK
MF3	C6, LK	C5, LK	MF3	C5, LK	RK1, RK	C5, LK	LK, LK	LK, LK	RK1, RK
MF4	-	-	MF4	-	RK4, RK	-	RK5, RK	-	RK4, RK
MF5	-	-	MF5	-	RK5, RK	-	-	-	RK5, RK
MF6	-	-	MF6	-	RK6, RK	-	RK7, RK	-	RK6, RK
MF7	-	-	MF7	-	RK7, RK	-	-	-	RK7, RK
RK1	C2, LK	C1, LK	START	C1, LK	RK1, RK	C1, LK	LK, LK	LK, LK	RK1, RK
RK2	C2, LK	C1, LK	START	C1, LK	RK2, RK	C1, LK	RK3, RK	-	RK2, RK
RK3	C2, LK	C1, LK	START	C1, LK	RK3, RK	C1, LK	LK, LK	LK, LK	RK3, RK
RK4	C4, LK	C3, LK	VF	C3, LK	RK4, RK	C3, LK	RK5, RK	-	RK4, RK
RK5	C4, LK	C3, LK	VF	C3, LK	RK5, RK	C3, LK	LK, LK	LK, LK	RK5, RK
RK6	C6, LK	C5, LK	MF3	C5, LK	RK6, RK	C5, LK	RK7, RK	-	RK6, RK
RK7	C6, LK	C5, LK	MF3	C5, LK	RK7, RK	C5, LK	LK, LK	LK, LK	RK7, RK
START	C2, LK	C1, LK	VF	C1, LK	RK1, RK	C1, LK	LK, LK	LK, LK	RK1, RK
VF	C4, LK	C3, LK	VF	C3, LK	RK1, RK	C3, LK	LK, LK	LK, LK	RK1, RK

Table A.5.: Transition table of the finite state transducer from the pilot study on sentence bracket identification. For every input tag (*Event*), the transducer transitions from its current state (*State*) to the state given in the respective table cell. If the state is part of a sentence bracket, the second value from the cell, `LK` or `RK`, is output. Regarding the events, KOUI and KOUS correspond to the respective POS tags. PREL includes words tagged as PRELS and PRELAT, and PW includes PWAT, PWAV, and PWS. PTK refers to PTKZU and PTKVZ. Finite verbs are captured with VFin. VImp includes all imperatives, and VNonFin corresponds to the remaining verbs. OTH includes all other words and is also used to make the transducer more robust against ungrammatical sentences and fragments. If an input word is not accepted at the current state, OTH is tried instead of rejecting the sentence, thus enabling partial analyses. So, for example, if the transducer is in state VF (corresponding to the pre-field of the sentence) and a finite verb (VFin) is encountered, the transducer transitions to state LK (corresponding to a left bracket) and outputs the label `LK`. From its new state, the transducer can only transition to MF3 (corresponding to a middle field) or RK1 (a right sentence bracket). If the next token was an imperative (VImp), this could not be accepted in the current state, so the transducer would try OTH and transition to the middle field MF3 instead of rejecting the whole sentence.

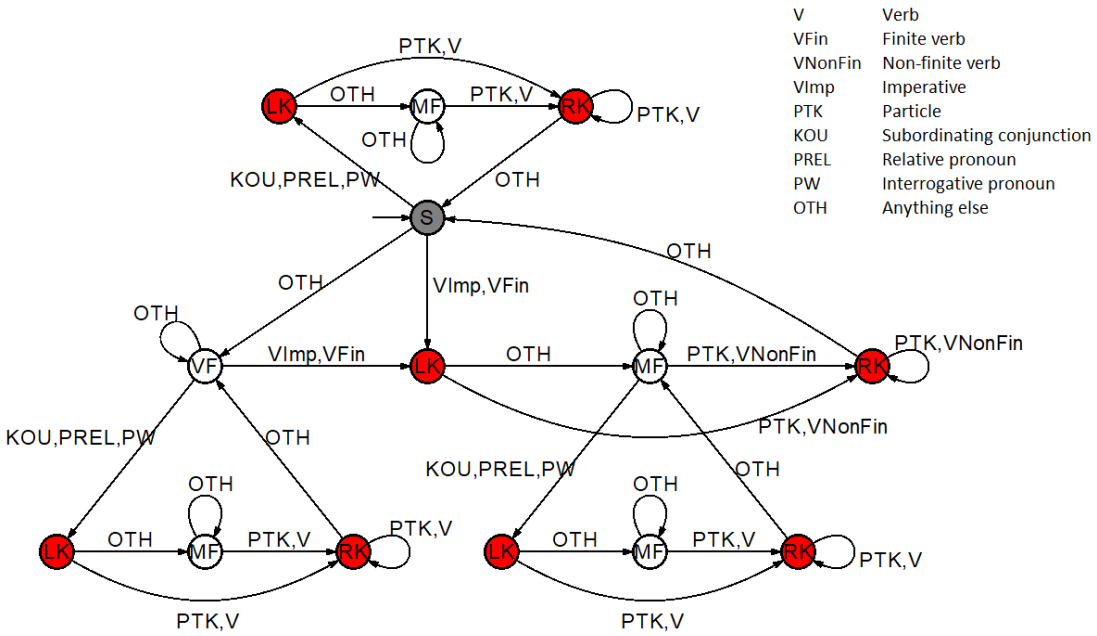


Figure A.1.: Simplified graphical representation of the finite state transducer from the pilot study on sentence bracket identification. States that correspond to sentence brackets are colored in red. Starting from the initial state S (grey), all clause types (V1, V2, and VL) can be parsed. The large triangle in the middle roughly corresponds to a V2 clause with pre-field (state `VF`), left bracket (`LK`), middle field (`MF`), and right bracket (`RK`). For V1 clauses, the transducer may directly transition to the left bracket. VL clauses can be parsed with the small triangle at the top and may also be embedded in the pre- or middle field (small triangles at the bottom). The complete transition table can be found in Table A.5. The image was created with the AutomataEditor by Kriz (2011).

```
LK:
<PW(S|AV|AT)>{<V.FIN>}          #1) Finite verb directly following an interrogative
{<PW(S|AV|AT)>}<[^L].*>         #2) Interrogative not followed by a left bracket
{<PREL(S|AT)|KOU(I|S)>+}        #3) Relative pronouns and conjunctions
RK:
<LK><.*>*?{<PTK(ZU|VZ)|V.+>+}   #4) Particles and/or verbs following a left bracket
LK:
{<V.(FIN|IMP)>}                 #5) Verbal left bracket
RK:
{<PTK(ZU|VZ)|V.+>+}             #6) Remaining verbs and particles
```

	Er	ist	gekommen	,	um	das	,	was	der	Teufel	tut	,	zu	zerstören	.
	PPER	VAFIN	VVPP	$,	KOUI	PDS	$,	PRELS	ART	NN	VVFIN	$,	PTKZU	VVINF	$.
1)															
2)															
3)					**LK**			**LK**							
4)					LK			LK			**RK**			**RK**	
5)		**LK**			LK			LK			RK			RK	
6)		LK	**RK**		LK			LK			RK			RK	

He has come to destroy what the devil does.

Figure A.2.: Rules used by the regular expression parser in the pilot study on sentence bracket identification. The rules are applied in the given order and return non-overlapping matches of the expressions in curly brackets. All matches are chunked and labeled with the corresponding tag. The chinks, as defined by everything outside of curly brackets, restrict the annotation context. Rules 1, 2, 3, and 5 identify the left sentence bracket and rule 4 and 6 the right sentence bracket. In the example from the Modern data set, rules 1 and 2 do not match because there is no interrogative pronoun in the sentence. Rule 3 matches the subordinating conjunction and the relative pronoun in the subordinate clauses and labels them as left brackets before rule 4 matches the finite verbs in the corresponding right brackets. Finally, rules 5 and 6 match the left and right bracket in the main clause.

```
#Sentence
S -> V2 | VL | FRAG

#Without pre-field (V1)
V2 -> LK | LK MF | LK NF | LK MF RK-V2 | LK RK-V2 | LK MF RK-V2 NF | LK RK-V2 NF

#With pre-field (V2)
V2 -> VF LK | VF LK MF | VF LK NF | VF LK MF RK-V2 | VF LK RK-V2 |
      VF LK MF RK-V2 NF | VF LK RK-V2 NF
V2 -> KOORD V2 | V2 KOORD V2

#VL clause without LK (with infinitive)
VL -> RK-Inf | MF RK-Inf | MF RK-Inf NF | RK-Inf NF

#VL clause with LK
VL -> LK-C RK-VL | LK-C MF RK-VL | LK-C MF RK-VL NF | LK-C RK-VL NF
VL -> LV VL | KOORD VL | VL KOORD VL

#Fragment
FRAG -> OTH | OTH FRAG

#Main fields
LV -> OTH | OTH LV
VF -> OTH | OTH VF | RK-Inf | VL | PW
MF -> OTH | OTH MF | VL
NF -> OTH | OTH NF | VL | V2 | FRAG | PW

#Right sentence bracket
RK-V2 -> 'PTKVZ' | 'PTKVZ' RK-V2 | VNonFin | VNonFin RK-V2 | RK-V2 KOORD RK-V2
RK-VL -> V | V RK-VL | RK-VL KOORD RK-VL
RK-Inf -> VInf | VNonFin | 'PTKVZ' | VNonFin VInf | RK-Inf KOORD RK-Inf

#Left sentence bracket
LK-C -> PREL | KOU | PW | PREL LK-C | KOU LK-C | PW LK-C | LK-C KOORD LK-C
LK -> LK-Fin | LK-Imp
LK-Fin -> VFin | LK-Fin KOORD LK-Fin
LK-Imp -> VImp | LK-Imp KOORD LK-Imp

#Coordination field
KOORD -> 'KON'

#Verb categories
V -> VFin | VNonFin
VNonFin -> VInf | VPP
VFin -> 'VVFIN' | 'VAFIN' | 'VMFIN'
VImp -> 'VVIMP' | 'VAIMP' | 'VMIMP'
VInf -> 'VVINF' | 'VAINF' | 'VMINF' | 'VVIZU' | 'PTKZU' VInf
VPP -> 'VVPP' | 'VAPP' | 'VMPP'

#Complementizer categories
PW -> 'PWS' | 'PWAV' | 'PWAT'
PREL -> 'PRELS' | 'PRELAT'
KOU -> 'KOUS' | 'KOUI'

#Remaining POS tags
OTH -> 'ART' | 'ADJA' | 'ADJD' | 'APPR' | 'APPRART' | 'APZR' | 'APPO' | 'ADV' |
       'CARD' | 'FM' | 'ITJ' | 'KOKOM' | 'KON' | 'NN' | 'NE' | 'PDS' | 'PDAT' |
       'PIS' | 'PIAT' | 'PIDAT' | 'PPER' | 'PPOSS' | 'PPOSAT' | 'PRF' | 'PAV' |
       'PTKNEG' | 'PTKANT' | 'PTKA' | 'TRUNC' | 'XY' | '$,' | '$.' | '$('
```

Figure A.3.: Hand-written context-free grammar used by the bottom-up left-corner chart parser in the pilot study on sentence bracket identification. The grammar is based on the basic topological field model depicted in Figure 5.1 and provides rules for the different clause types while also considering possibly empty fields. A rule's left-hand side specifies the parent node, while the right-hand side lists possible child nodes. Alternatives are separated by pipes. For example, a sentence S can consist of either a V2 clause, a VL clause, or a fragment. Terminal nodes, i.e., POS tags, are surrounded by quotation marks. Terminal nodes dominated by an LK or LK-C node are counted as left sentence bracket, while terminal nodes dominated by an RK-V2, RK-VL, or RK-Inf node are counted as right sentence bracket.

	Punct			News1		
Corpus	Prec	Rec	F_1	Prec	Rec	F_1
Traditional						
TüBa-D/Z	98.61	99.10	98.85	**99.42**	**99.21**	**99.31**
Spoken	97.15	98.84	97.99	**98.30**	**99.11**	**98.70**
Modern	97.97	**99.16**	**98.56**	**98.17**	98.40	98.29
HIPKON	**94.29**	**94.53**	**94.41**	87.73	82.71	85.15
DTA	**88.23**	**91.53**	**89.85**	88.08	89.88	88.97
FairEval						
TüBa-D/Z	99.06	99.36	99.21	**99.58**	**99.39**	**99.48**
Spoken	97.60	99.29	98.44	**98.59**	**99.40**	**98.99**
Modern	98.56	**99.39**	**98.97**	**98.81**	98.81	98.81
HIPKON	**96.88**	**94.82**	**95.84**	90.91	83.39	86.98
DTA	92.25	**92.46**	**92.36**	**92.35**	91.18	91.77

Table A.6.: Overall precision, recall, and F_1-scores (in percent) for sentence bracket recognition according to traditional and fair evaluation for the different models on each data set. The highest scores for each corpus are highlighted in bold.

	Punct			News1		
Corpus	Prec	Rec	F_1	Prec	Rec	F_1
Traditional						
TüBa-D/Z	94.15	95.14	94.64	**96.36**	**96.47**	**96.41**
Spoken	86.55	90.54	88.50	**89.27**	**91.68**	**90.46**
Modern	94.81	**93.28**	**94.04**	**94.86**	92.50	93.66
HIPKON	**90.99**	**89.72**	**90.35**	82.27	77.40	79.76
DTA	**79.22**	**81.30**	**80.25**	78.67	78.63	78.65
FairEval						
TüBa-D/Z	95.25	96.67	95.96	**97.31**	**97.44**	**97.37**
Spoken	88.72	93.58	91.08	**91.30**	**94.14**	**92.70**
Modern	96.10	**94.38**	**95.23**	**96.78**	93.63	95.18
HIPKON	**93.99**	**92.69**	**93.34**	86.42	83.36	84.86
DTA	85.22	**85.61**	**85.42**	**85.61**	82.80	84.18

Table A.7.: Overall precision, recall, and F_1-scores (in percent) for topological field parsing according to traditional and fair evaluation for the different models on each data set. The highest scores for each corpus are highlighted in bold.

A.3. Chunks

```
PC:
 1) {<KOKOM>*<APPR><(ART|PPOSAT|PDAT|PIAT|PWAT|CARD|ADJA|ADJD|ADV|PTKNEG|$,|$\(
                    |KON|TRUNC)>*<(NN|NE)>+<APZR>*}
 2) {<KOKOM>*<APPRART><(CARD|ADJA|ADJD|ADV|PTKNEG|$,|$\(|KON|TRUNC)>*
     <(NN|NE)>+<APZR>*}
 3) {<KOKOM>*<(ART|PPOSAT|PDAT|PIAT|PWAT|CARD|ADJA|TRUNC)>
     <(ART|PPOSAT|PDAT|PIAT|PWAT|CARD|ADJA|ADJD|ADV|PTKNEG|$,|$\(|KON|TRUNC)>*
     <(NN|NE)>+<APPO>+}
 4) {<KOKOM>*<(ART|PPOSAT|PDAT|PIAT|PWAT|CARD|ADJA|TRUNC)>*<(NN|NE)>+<APPO>+}
 5) {<KOKOM>*<APPR><ART><(PIS|PPOSS)><APZR>*}
 6) {<KOKOM>*<ART><(PIS|PPOSS)><APPO>}
 7) {<KOKOM>*<(APPR|APPRART)><(PIS|PDS|PWS|PPER|PPOSS|PRELS|PRF)><APZR>*}
 8) {<KOKOM>*<(PIS|PDS|PWS|PPER|PPOSS|PRELS|PRF)><APPO>}
 9) {<KOKOM>*<APPR>*<PAV>}

NC:
10) {<KOKOM>*<(ART|PPOSAT|PDAT|PIAT|PWAT|CARD|ADJA|TRUNC)>
     <(ART|PPOSAT|PDAT|PIAT|PWAT|CARD|ADJA|ADJD|ADV|PTKNEG|$,|$\(|KON|TRUNC)>*
     <(NN|NE)>+}
11) {<KOKOM>*<(ART|PPOSAT|PDAT|PIAT|PWAT|CARD|ADJA|TRUNC)>*<(NN|NE)>+}
12) {<KOKOM>*<ART><(PIS|PPOSS)>}
13) {<KOKOM>*<(PIS|PDS|PWS|PPER|PPOSS|PRELS|PRF)>}

AC:
14) {<KOKOM>*<(ADJA|ADV|PTKNEG|PTKA)>*<ADJD>+}

ADVC:
15) {<KOKOM>*<(ADV|PTKNEG)>+}

NC:
16) {<KOKOM>*<CARD>+}

sPC:
17) {<KOKOM>*<(APPR|APPRART)><(ART|PPOSAT|PDAT|PIAT|PWAT|ADJA)>*}

sNC:
18) {<KOKOM>*<(ART|PPOSAT|PDAT|PIAT|PWAT|ADJA)>}
```

Figure A.4.: Rules used by the POS-based regular expression chunker. The rules are applied in the given order and return non-overlapping matches of the expressions in curly brackets. First, rules 1–9 identify prepositional chunks. Then, rules 10–13 add noun chunks, etc. POS tags are from the STTS (Schiller et al. 1999). Linebreaks and rule numbers are added here for better readability. For an application example of regex-based chunking to sentence bracket recognition, see Figure A.2.

	News1			News2			Hist			Mix		
Corpus	Prec	Rec	F_1	Prec	Rec	F_1	Prec	Rec	F_1	Prec	Rec	F_1
Traditional												
TüBa-D/Z	**95.80**	**94.53**	**95.16**	90.02	84.92	87.40	-	-	-	-	-	-
Tiger	83.97	89.69	86.74	**92.06**	**92.80**	**92.43**	-	-	-	-	-	-
Modern	88.97	**92.97**	90.92	**91.12**	91.06	**91.09**	-	-	-	-	-	-
Mercurius	84.92	86.94	85.92	84.15	82.33	83.23	88.25	**90.33**	**89.28**	**88.82**	88.06	88.44
ReF.UP	85.55	85.30	85.43	84.13	81.53	82.81	89.68	**89.21**	**89.45**	**89.95**	88.30	89.12
HIPKON	88.12	90.19	89.14	89.50	88.10	88.79	90.00	**91.24**	**90.61**	**90.29**	90.65	90.47
DTA	85.07	85.48	85.27	83.27	80.97	82.10	85.15	**85.69**	**85.42**	**87.00**	83.73	85.34
FairEval												
TüBa-D/Z	**96.85**	**96.77**	**96.81**	91.45	91.02	91.24	-	-	-	-	-	-
Tiger	90.38	90.99	90.68	**94.99**	**95.04**	**95.01**	-	-	-	-	-	-
Modern	93.56	93.63	93.59	**94.12**	**93.81**	**93.96**	-	-	-	-	-	-
Mercurius	90.61	90.14	90.38	88.56	88.52	88.54	**92.83**	**92.78**	**92.81**	92.24	92.13	92.18
ReF.UP	90.44	89.05	89.74	87.94	87.80	87.87	**93.12**	**92.24**	**92.67**	92.95	91.95	92.45
HIPKON	92.80	92.74	92.77	92.45	92.39	92.42	**93.94**	**93.75**	**93.84**	93.93	93.74	93.84
DTA	90.16	88.65	89.40	87.38	86.70	87.04	90.50	**88.72**	**89.60**	**90.63**	88.57	89.59

Table A.8.: Overall precision, recall, and F_1-scores (in percent) according to traditional and fair evaluation for chunking with the different models on each data set. Models trained on historical data are only applied to the historical test sets, and the highest scores for each corpus are highlighted in bold.

A.4. Phrases

	News1			News2			Hist			Mix		
Corpus	Prec	Rec	F_1	Prec	Rec	F_1	Prec	Rec	F_1	Prec	Rec	F_1
Traditional												
TüBa-D/Z	**86.95**	**89.07**	**88.00**	77.86	74.17	75.97	-	-	-	-	-	-
Tiger	75.92	82.29	78.98	**79.66**	**84.27**	**81.90**	-	-	-	-	-	-
Spoken	**85.16**	**85.80**	**85.48**	76.84	74.61	75.71	-	-	-	-	-	-
Modern	**83.25**	**84.33**	**83.79**	81.40	77.77	79.54	-	-	-	-	-	-
Mercurius	51.54	58.15	54.64	56.22	59.17	57.66	**74.53**	**77.35**	**75.91**	74.27	76.66	75.45
ReF.UP	51.01	51.42	51.21	49.24	53.75	51.39	**79.54**	80.07	**79.80**	78.97	**80.49**	79.72
HIPKON	65.58	68.32	66.92	63.76	71.99	67.62	**78.19**	81.33	**79.73**	77.27	**81.59**	79.37
DTA	66.08	**62.75**	**64.37**	63.99	56.25	59.87	62.96	56.80	59.72	**66.14**	57.89	61.74
FairEval												
TüBa-D/Z	**91.35**	**91.35**	**91.35**	82.04	81.10	81.57	-	-	-	-	-	-
Tiger	83.49	84.75	84.11	**85.66**	**87.14**	**86.39**	-	-	-	-	-	-
Spoken	**88.98**	**89.84**	**89.41**	80.26	82.30	81.27	-	-	-	-	-	-
Modern	**88.27**	**88.23**	**88.25**	84.77	84.11	84.44	-	-	-	-	-	-
Mercurius	61.81	64.34	63.05	66.00	65.77	65.88	**81.25**	**81.82**	**81.53**	81.04	81.29	81.16
ReF.UP	58.55	58.93	58.74	59.07	59.03	59.05	**84.02**	**84.30**	**84.16**	83.98	84.15	84.07
HIPKON	74.54	74.75	74.64	75.10	75.45	75.27	**84.88**	**84.96**	**84.92**	84.69	84.77	84.73
DTA	**73.03**	**70.07**	**71.52**	69.61	64.90	67.17	69.10	64.80	66.88	70.45	66.75	68.55

Table A.9.: Overall precision, recall, and F_1-scores (in percent) according to traditional and fair evaluation for phrase recognition with the different models on each data set. Models trained on historical data are only applied to the historical test sets, and the highest scores for each corpus are highlighted in bold.

	News1	News2	Hist	Mix
TüBa-D/Z	**91.96**	n.a.	n.a.	n.a.
Tiger	n.a.	**86.42**	n.a.	n.a.
Mercurius	n.a.	52.27	**77.68**	77.44
ReF.UP	n.a.	45.15	78.97	**79.13**

Table A.10.: Overall labeled F_1-score for the four trained parser models on the test data, excluding virtual root nodes. Training and test trees are modified as described in Section 6.2.2, and models are only evaluated on test data that follows the same syntactic annotation scheme as the training data.

A.5. Relative Clauses

	News1			News2			Hist			Mix		
Corpus	Prec	Rec	F_1	Prec	Rec	F_1	Prec	Rec	F_1	Prec	Rec	F_1
Traditional												
TüBa-D/Z	**93.41**	**92.78**	**93.09**	84.85	83.64	84.24	-	-	-	-	-	-
Tiger	90.46	90.46	90.46	**93.44**	**93.11**	**93.27**	-	-	-	-	-	-
Spoken	**85.93**	**84.38**	**85.15**	83.64	81.38	82.50	-	-	-	-	-	-
Modern	**87.69**	**87.69**	**87.69**	80.00	80.00	80.00	-	-	-	-	-	-
HIPKON	81.08	65.22	72.29	81.25	56.52	66.67	**86.67**	**84.78**	**85.71**	84.78	**84.78**	84.78
DTA	**72.19**	63.74	**67.70**	64.10	58.48	61.16	66.25	61.99	64.05	66.87	**64.91**	65.88
FairEval												
TüBa-D/Z	**96.47**	**95.85**	**96.16**	91.74	90.45	91.09	-	-	-	-	-	-
Tiger	94.64	94.64	94.64	**96.52**	**96.17**	**96.34**	-	-	-	-	-	-
Spoken	**92.13**	**90.35**	**91.23**	90.94	88.27	89.59	-	-	-	-	-	-
Modern	**93.44**	**91.94**	**92.68**	88.89	87.39	88.14	-	-	-	-	-	-
HIPKON	85.71	68.18	75.95	86.67	59.09	70.27	**89.66**	87.64	**88.64**	88.64	**88.64**	**88.64**
DTA	**81.95**	72.19	76.76	76.92	69.93	73.26	77.09	72.35	74.65	77.89	**75.77**	**76.82**

Table A.11.: Overall precision, recall, and F_1-scores (in percent) according to traditional and fair evaluation for RelC identification with the different models on each data set. Models trained on historical data are only applied to the historical test sets, and the highest scores for each corpus are highlighted in bold.

A.6. Extraposition

	News1			News2			Hist			Mix		
Corpus	Prec	Rec	F_1	Prec	Rec	F_1	Prec	Rec	F_1	Prec	Rec	F_1
Traditional												
TüBa-D/Z	**86.54**	**89.03**	**87.77**	79.18	76.96	78.05	-	-	-	-	-	-
Tiger	82.33	86.30	84.27	**85.46**	**89.26**	**87.32**	-	-	-	-	-	-
Spoken	**83.93**	**86.28**	**85.09**	76.96	77.41	77.19	-	-	-	-	-	-
Modern	**78.26**	**82.93**	**80.53**	76.69	78.91	77.78	-	-	-	-	-	-
HIPKON	63.35	65.39	64.35	63.41	67.69	65.48	**76.13**	76.36	**76.25**	75.45	**76.44**	75.94
DTA	63.77	**68.54**	**66.07**	63.16	62.08	62.62	63.30	63.55	63.42	**65.63**	66.33	65.98
FairEval												
TüBa-D/Z	**91.12**	**91.91**	**91.52**	84.66	83.49	84.07	-	-	-	-	-	-
Tiger	87.84	92.37	90.05	**89.93**	**94.14**	**91.98**	-	-	-	-	-	-
Spoken	**88.85**	**90.21**	**89.53**	82.43	83.73	83.08	-	-	-	-	-	-
Modern	**86.09**	**86.56**	**86.32**	84.54	83.61	84.07	-	-	-	-	-	-
HIPKON	73.10	74.12	73.61	75.03	74.46	74.75	**84.21**	82.88	**83.54**	83.59	**82.97**	83.28
DTA	73.39	**74.94**	74.16	72.16	70.23	71.18	73.07	70.43	71.73	**74.80**	73.78	**74.29**

Table A.12.: Overall precision, recall, and F_1-scores (in percent) according to traditional and fair evaluation for extraposition analysis with the different models on each data set. Models trained on historical data are only applied to the historical test sets, and the highest scores for each corpus are highlighted in bold.

A.7. Example Analysis

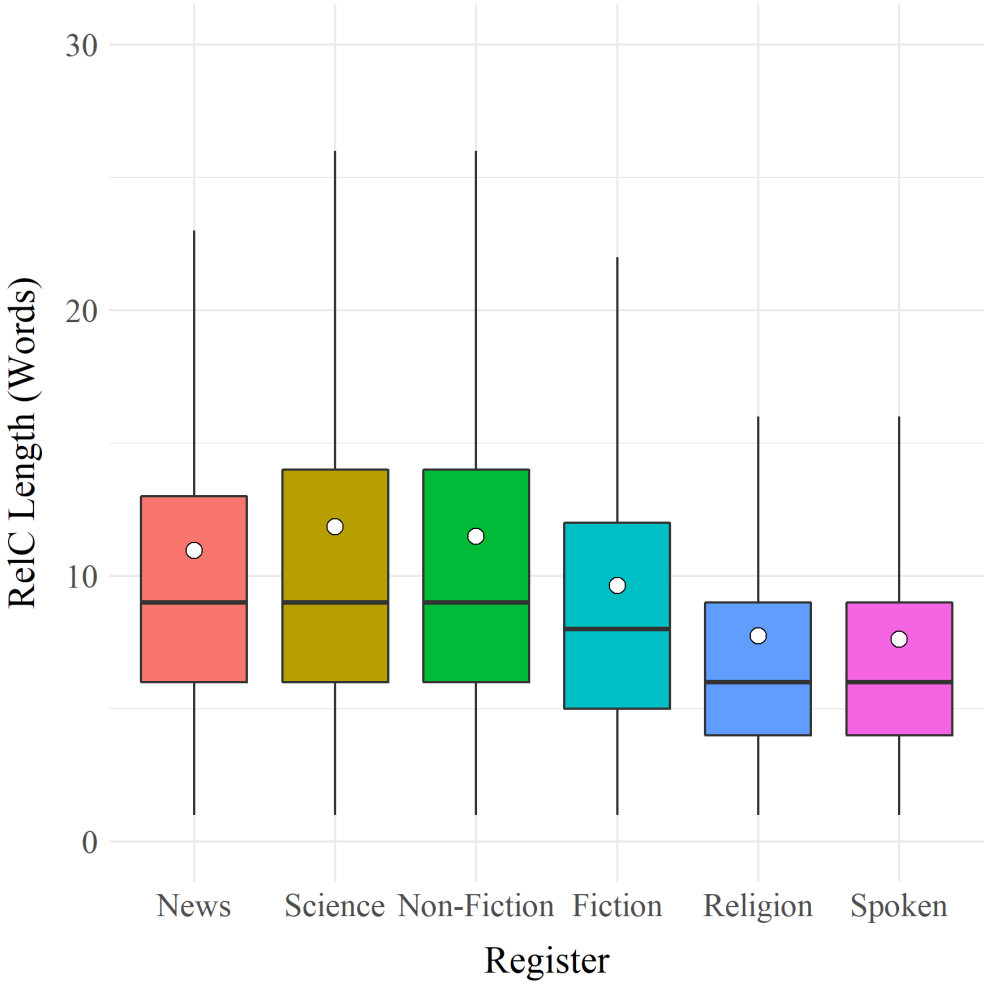


Figure A.5.: Average length of relative clauses per register. The boxes show the interquartile range from first to third quartile, with a black line for median RelC length. The mean is indicated with a white dot. For better readability, outliers (i.e., longer RelCs) are not displayed here.

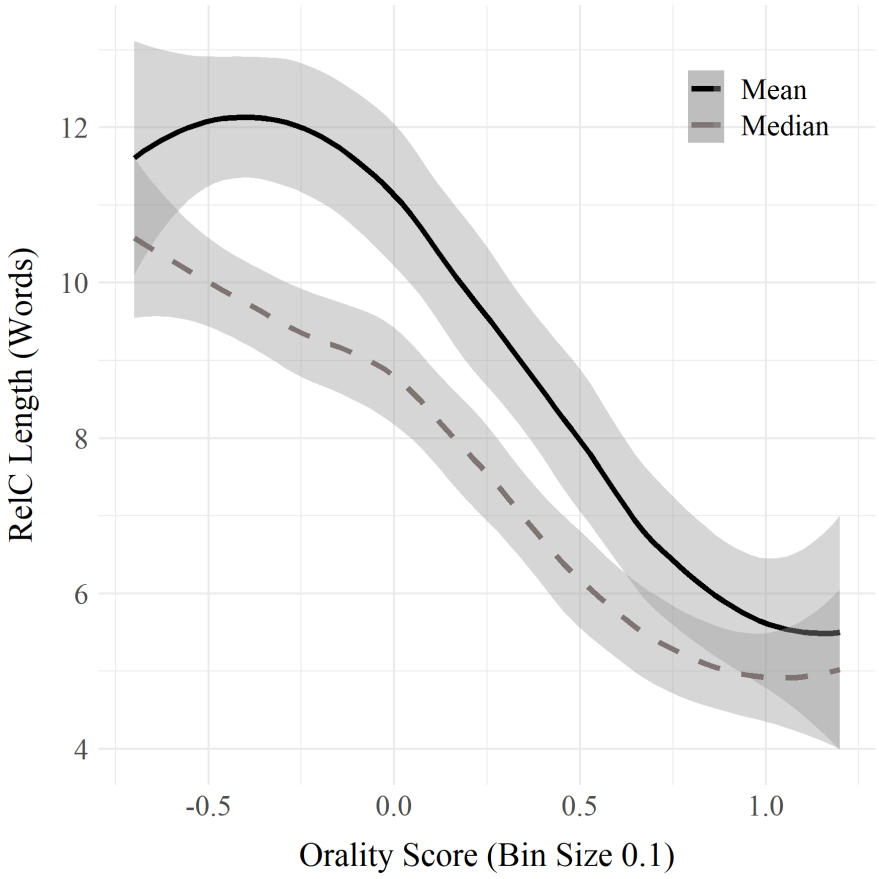


Figure A.6.: Mean and median length of relative clauses by orality score. For visualization, scores are grouped into bins of size 0.1. The plot shows the local regression lines over orality bins (LOESS smoothing) with a confidence interval of 0.95.

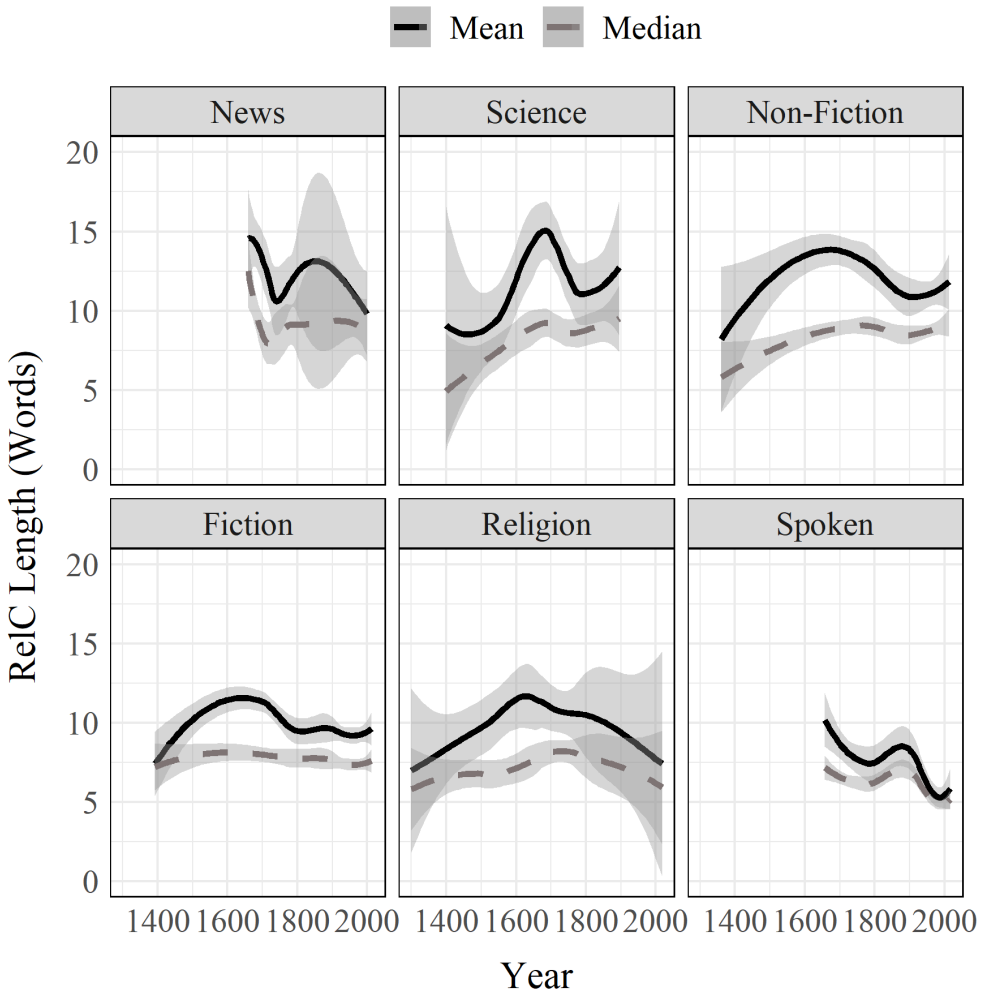


Figure A.7.: Mean and median length of relative clauses per register over time. The plot shows the local regression lines (LOESS smoothing) with a confidence interval of 0.95.

Corpus	$DORM_{diff}$	df	t	p-value		Cohen's d	Effect size	
$Gutenberg_{Fiction}$	-	238	-1.082	0.280		0.07		
$Gutenberg_{Folk\text{-}Tales}$	-	239	-1.210	0.228		0.08		
$Gutenberg_{Non\text{-}Fiction}$	+	235	0.730	0.466		0.05		
$Gutenberg_{Speech}$	+	230	0.502	0.616		0.03		
$OPUS_{Action}$	-	245	-1.371	0.172		0.09		
$OPUS_{Comedy}$	-	240	-2.784	< 0.01	**	0.18		
$OPUS_{Drama}$	-	238	-1.484	0.139		0.10		
SdeWaC	+	220	3.871	< 0.001	***	0.26	small	x
SermonOnline	-	232	-5.143	< 0.001	***	0.34	small	✓
Tiger	+	167	2.001	< 0.05	*	0.15		
TüBa-D/S	-	84	-2.046	< 0.05	*	0.22	small	✓
TüBa-D/W	+	241	2.497	< 0.05	*	0.16		
TüBa-D/Z	+	249	1.514	0.131		0.10		
Anselm	-	187	-7.024	< 0.001	***	0.51	medium	✓
$DTA_{Science}$	-	196	-0.803	0.423		0.06		
$GerManC_{DRAM}$	+	189	1.868	< 0.1	.	0.14		
$GerManC_{HUMA}$	+	203	2.211	< 0.05	*	0.16		
$GerManC_{LEGA}$	-	196	-0.485	0.628		0.04		
$GerManC_{NARR}$	+	202	2.062	< 0.05	*	0.14		
$GerManC_{NEWS}$	-	217	-1.799	< 0.1	.	0.12		
$GerManC_{SCIE}$	-	203	-0.308	0.758		0.02		
$GerManC_{SERM}$	-	212	-1.643	0.102		0.11		
ReF.RUB	-	193	-0.974	0.331		0.07		

Table A.13.: Difference in DORM values between original and variant sentences based on bigram word form surprisal. Since surprisal values are not comparable between language models, only the direction of the difference is given. A negative difference (−) means that the original sentence with extraposition has a lower DORM value (i.e., a smoother information profile) than the variant sentence. If the variant sentence would be smoother, the $DORM_{diff}$ value is positive (+). For each corpus, the table also shows the results of a one-sample t-test and the effect size according to Cohen's d. If $DORM_{diff}$ is negative with a significant p-value and at least a small effect $d \geq 0.2$, the hypothesis is confirmed (✓). If $DORM_{diff}$ is positive with a significant p-value and at least a small effect $d \geq 0.2$, the hypothesis is rejected (x). Otherwise, there is no evidence for or against the hypothesis.

Corpus	$DORM_{diff}$	df	t	p-value		Cohen's d	Effect size	
$Gutenberg_{Fiction}$	-	238	-0.898	0.370		0.06		
$Gutenberg_{Folk\text{-}Tales}$	-	239	-3.204	< 0.01	**	0.21	small	✓
$Gutenberg_{Non\text{-}Fiction}$	-	235	-2.668	< 0.01	**	0.17		
$Gutenberg_{Speech}$	-	230	-3.139	< 0.01	**	0.21	small	✓
$OPUS_{Action}$	+	245	2.209	< 0.05	*	0.14		
$OPUS_{Comedy}$	+	240	5.732	< 0.001	***	0.37	small	x
$OPUS_{Drama}$	+	238	3.016	< 0.01	**	0.20		
SdeWaC	-	220	-8.329	< 0.001	***	0.56	medium	✓
SermonOnline	-	232	-4.323	< 0.001	***	0.28	small	✓
Tiger	-	167	-2.203	< 0.05	*	0.17		
TüBa-D/S	+	84	7.962	< 0.001	***	0.86	large	x
TüBa-D/W	-	241	-7.687	< 0.001	***	0.49	small	✓
TüBa-D/Z	-	249	-1.969	< 0.1	.	0.12		
Anselm	-	187	-1.865	< 0.1	.	0.14		
$DTA_{Science}$	-	196	-3.057	< 0.01	**	0.22	small	✓
$GerManC_{DRAM}$	-	189	-2.414	< 0.05	*	0.18		
$GerManC_{HUMA}$	-	203	-1.553	0.122		0.11		
$GerManC_{LEGA}$	-	196	-1.398	0.164		0.10		
$GerManC_{NARR}$	-	202	-0.403	0.687		0.03		
$GerManC_{NEWS}$	-	217	-1.774	< 0.1	.	0.12		
$GerManC_{SCIE}$	-	203	-0.225	0.822		0.02		
$GerManC_{SERM}$	-	212	-0.358	0.720		0.02		
ReF.RUB	+	193	0.224	0.823		0.02		

Table A.14.: Difference in DORM values between original and variant sentences based on bigram POS surprisal. Since surprisal values are not comparable between language models, only the direction of the difference is given. A negative difference (−) means that the original sentence with extraposition has a lower DORM value (i.e., a smoother information profile) than the variant sentence. If the variant sentence would be smoother, the $DORM_{diff}$ value is positive (+). For each corpus, the table also shows the results of a one-sample t-test and the effect size according to Cohen's d. If $DORM_{diff}$ is negative with a significant p-value and at least a small effect $d \geq 0.2$, the hypothesis is confirmed (✓). If $DORM_{diff}$ is positive with a significant p-value and at least a small effect $d \geq 0.2$, the hypothesis is rejected (x). Otherwise, there is no evidence for or against the hypothesis.

Bibliography

Abney, Steven P. (1991). Parsing by chunks. In *Principle-based parsing*. Ed. by Robert C. Berwick, Steven P. Abney, and Carol Tenny. Vol. 44. Studies in Linguistics and Philosophy. Springer, pp. 257–278.

Ágel, Vilmos and Mathilde Hennig (2006). Grammatik aus Nähe und Distanz: Theorie und Praxis am Beispiel von Nähetexten 1650-2000. Tübingen: Niemeyer.

Ágel, Vilmos and Mathilde Hennig (2008). Kasseler Junktionskorpus. Justus-Liebig-Universität Gießen.

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

Altmann, Hans (1981). Formen der „Herausstellung“ im Deutschen. Rechtsversetzung, Linksversetzung, Freies Thema und verwandte Konstruktionen. Linguistische Arbeiten; 106. Tübingen: Max Niemeyer Verlag.

Anastasiou, Dimitra and Oliver Čulo (2007). Using Topological Information for detecting idiomatic verb phrases in German. In *Proceedings of the Conference on Practical Applications in Language and Computers (PALC)*, pp. 49–58.

Augst, Gerhard, Karl Blüml, Dieter Nerius, and Horst Sitta (1997). Zur Neuregelung der deutschen Orthographie. Begründung und Kritik. Vol. 179. Reihe Germanistische Linguistik. Tübingen: Max Niemeyer Verlag.

Bastings, Joost and Khalil Sima'an (2014). All Fragments Count in Parser Evaluation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. Reykjavik, Iceland, pp. 78–82.

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

Becker, Markus and Anette Frank (2002). A Stochastic Topological Parser for German. In *19th International Conference on Computational Linguistics, COLING 2002*. Taipei, Taiwan.

Becker, Markus and Elsa Pecourt (2002). Anaphora resolution using a topological parser. In *Proceedings of the Fourth Discourse Anaphora and Anaphor Resolution Colloquium*. Lisbon, Portugal.

Behaghel, Otto (1932). Deutsche Syntax: eine geschichtliche Darstellung. Vol. 4: Wortstellung, Periodenbau. Germanische Bibliothek. Heidelberg: Winter.

Benikova, Darina, Chris Biemann, Max Kisselew, and Sebastian Padó (2014). GermEval 2014 Named Entity Recognition Shared Task: Companion Paper. In *Workshop Proceedings of the 12th KONVENS 2014*. Hildesheim, Germany.

Bennett, Paul, Martin Durrell, Astrid Ensslin, Silke Scheible, and Richard Whitt (2007). GerManC (Version 1.0). University of Manchester.

Betten, Anne (1989). Zur Problematik der Abgrenzung von Mündlichkeit und Schriftlichkeit bei mittelalterlichen Texten. In *Neuere Forschungen zur historischen Syntax des Deutschen. Referate der internationalen Fachkonferenz Eichstätt*, pp. 324–335.

Biber, Douglas (1995). Dimensions of register variation: a cross-linguistic comparison. Cambridge University Press.

Bizzoni, Yuri, Stefania Degaetano-Ortlieb, Peter Fankhauser, and Elke Teich (2020). Linguistic Variation and Change in 250 years of English Scientific Writing: A Data-driven Approach. In *Frontiers in Artificial Intelligence, section Language and Computation*.

Black, E., S. P. Abney, D. Flickinger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. P. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski (1991). A procedure for quantitatively comparing the syntactic coverage of English grammars. In *Proceedings DARPA Speech and Natural Language Workshop*. Pacific Grove, CA, pp. 306–311.

Bollmann, Marcel (2013). POS Tagging for Historical Texts with Sparse Training Data. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*. Sofia, Bulgaria, pp. 11–18.

Bollmann, Marcel (2018). Normalization of historical texts with neural network models. In *Bochumer Linguistische Arbeitsberichte (BLA)* 22.

Bollmann, Marcel, Florian Petran, Stefanie Dipper, and Julia Krasselt (2014). CorA: A web-based annotation tool for historical and other non-standard language data. In *Proceedings of the 8th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH)*. Gothenburg, Sweden, pp. 86–90.

Bosch, Antal van den 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*. Philadelphia, PA, pp. 433–440.

Brants, Sabine, 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. In *Research on language and computation* 2.4, pp. 597–620.

Braşoveanu, Adrian M.P., Giuseppe Rizzo, Philipp Kuntschik, Albert Weichselbraun, and Lyndon J.B. Nixon (2018). Framing named entity linking error types. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan, pp. 266–271.

Brill, Eric (1992). A simple rule-based part of speech tagger. In *Proceedings of the third conference on Applied natural language processing*. Trento, Italy, pp. 152–155.

Bybee, Joan (2015). Language Change. Cambridge Textbooks in Linguistics. Cambridge University Press.

Cheung, Jackie Chi Kit and Gerald Penn (2009). Topological Field Parsing of German. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Suntec, Singapore, pp. 64–72.

Chiarcos, Christian, 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.

Cieri, Christopher, Mike Maxwell, Stephanie Strassel, and Jennifer Tracey (2016). Selection criteria for low resource language programs. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož, Slovenia, pp. 4543–4549.

Coavoux, Maximin, Benoît Crabbé, and Shay B. Cohen (2019). Unlexicalized Transition-based Discontinuous Constituency Parsing. In *Transactions of the Association for Computational Linguistics, Volume 7*, pp. 73–89.

Coniglio, Marco, Karin Donhauser, and Eva Schlachter (2014). HIPKON: Historisches Predigtenkorpus zum Nachfeld (Version 1.0).

Coniglio, Marco and Eva Schlachter (2015). Das Nachfeld im Deutschen zwischen Syntax, Informations- und Diskursstruktur. In *Das Nachfeld im Deutschen. Theorie und Empirie*. Ed. by Hélène Vinckel-Roisin. Berlin, München, Boston: De Gruyter, pp. 141–164.

Cuskley, Christine, Rachael Bailes, and Joel Wallenberg (2021). Noise resistance in communication: Quantifying uniformity and optimality. In *Cognition* 214.

Dakota, Daniel and Sandra Kübler (2017). Towards Replicability in Parsing. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*. Varna, Bulgaria, pp. 185–194.

Daum, Michael, 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, pp. 99–106.

De Kok, Daniël (2014). TüBa-D/W: a large dependency treebank for German. In *Proceedings of the Thirteenth International Workshop on Treebanks and Linguistic Theories (TLT13)*. Tübingen, Germany, pp. 271–278.

Degaetano-Ortlieb, Stefania, Hannah Kermes, Ashraf Khamis, and Elke Teich (2019). An Information-Theoretic Approach to Modeling Diachronic Change in Scientific English. In *From Data to Evidence in English Language Research*. Ed. by Carla Suhr, Terttu Nevalainen, and Irma Taavitsainen. Leiden, NL: Brill, pp. 258–281.

Degaetano-Ortlieb, Stefania, Tanja Säily, and Yuri Bizzoni (2021). Registerial Adaptation vs. Innovation Across Situational Contexts: 18th Century Women in Transition. In *Frontiers in Artificial Intelligence, section Language and Computation* 4.

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

Demske, Ulrike (2019). Referenzkorpus Frühneuhochdeutsch: Baumbank.UP. Universität Potsdam.

Dipper, Stefanie, Karin Donhauser, Thomas Klein, Sonja Linde, Stefan Müller, and Klaus-Peter Wegera (2013). HiTS: ein Tagset für historische Sprachstufen des Deutschen. In *Journal for Language Technology and Computational Linguistics, Special Issue* 28.1, pp. 85–137.

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

Dipper, Stefanie and Sandra Kübler (2017). German Treebanks: TIGER and TüBa-D/Z. In *Handbook of linguistic annotation*. Ed. by Nancy Ide and James Pustejovsky. Springer, pp. 595–639.

Dipper, Stefanie and Simone Schultz-Balluff (2013). The Anselm Corpus: Methods and perspectives of a parallel aligned corpus. In *Proceedings of the workshop on computational historical linguistics at NODALIDA 2013. NEALT Proceedings Series 18*. Oslo, Norway, pp. 27–42.

Dipper, Stefanie and Sandra Waldenberger (2017). Investigating diatopic variation in a historical corpus. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*. Valencia, Spain, pp. 36–45.

Ebert, Robert Peter (1980). Social and stylistic variation in Early New High German word order: The sentence frame (›Satzrahmen‹). In *Jahresband* 102, pp. 357–398.

Eckart de Castilho, Richard, Éva Mújdricza-Maydt, Seid Muhie Yimam, Silvana Hartmann, Iryna Gurevych, Anette Frank, and Chris Biemann (2016). A Web-based Tool for the Integrated Annotation of Semantic and Syntactic Structures. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*. Osaka, Japan, pp. 76–84.

Eckhoff, Hanne Martine 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)*. Osaka, Japan: The COLING 2016 Organizing Committee, pp. 62–70.

Faaß, Gertrud and Kerstin Eckart (2013). SdeWaC – A Corpus of Parsable Sentences from the Web. In *Language Processing and Knowledge in the Web. Proceedings of the International Conference of the German Society for Computational Linguistics and Language Technology (GSCL)*.

Fisseni, Bernhard (2017). Das Bonner Frühneuhochdeutschkorpus (FnhdC). korpora.org.

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

Frank, Anette, Markus Becker, Berthold Crysmann, Bernd Kiefer, and Ulrich Schäfer (2003). Integrated shallow and deep parsing: TopP meets HPSG. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*. Sapporo, Japan, pp. 104–111.

Fraser, Alexander, Helmut Schmid, Richárd Farkas, Renjing Wang, and Hinrich Schütze (2013). Knowledge Sources for Constituent Parsing of German, a Morphologically Rich and Less-Configurational Language. In *Computational Linguistics* 39.1, pp. 57–85.

Frey, Werner and Karin Pittner (1998). Zur Positionierung der Adverbiale im deutschen Mittelfeld. In *Linguistische Berichte* 176, pp. 489–534.

Futrell, Richard, Kyle Mahowald, and Edward Gibson (2015). Large-scale evidence of dependency length minimization in 37 languages. In *Proceedings of the National Academy of Sciences of the United States of America* 112.

Geyken, Alexander, Susanne Haaf, and Frank Wiegand (2012). The DTA 'base format': A TEI-Subset for the Compilation of Interoperable Corpora. In *Proceedings of the 11th Conference on Natural Language Processing (KONVENS)*. Vienna, Austria, pp. 383–391.

Gibson, Edward (1998). Linguistic complexity: locality of syntactic dependencies. In *Cognition* 68, pp. 1–76.

Hale, John (2001). A Probabilistic Earley Parser as a Psycholinguistic Model. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language Technologies*. Pittsburgh, PA, pp. 1–8.

Halliday, Michael A. K. (1989). Spoken and written language. Oxford University Press.

Hawkins, John A. (1992). Syntactic Weight Versus Information Structure in Word Order Variation. In *Informationsstruktur und Grammatik*. Springer, pp. 196–219.

Hinrichs, Erhard W, Sandra Kübler, Frank Henrik Müller, and Tylman Ule (2002). A Hybrid Architecture for Robust Parsing of German. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002)*. Las Palmas, Spain, pp. 1505–1512.

Hinrichs, Erhard W., Julia Bartels, Yasuhiro Kawata, Valia Kordoni, and Heike Telljohann (2000). The Tübingen Treebanks for Spoken German, English, and Japanese. In *Verbmobil: Foundations of Speech-to-Speech Translation*. Ed. by Wolfgang Wahlster. Berlin: Springer, pp. 550–574.

Hinrichs, Erhard W., Marie Hinrichs, and Thomas Zastrow (2010). WebLicht: Web-Based LRT Services for German. In *Proceedings of the ACL 2010 System Demonstrations*. Uppsala, Sweden, pp. 25–29.

Hinrichs, Erhard W. and Thomas Zastrow (2012). Automatic Annotation and Manual Evaluation of the Diachronic German Corpus TüBa-D/DC. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey, pp. 1622–1627.

Hirschmann, Hagen and Sonja Linde (2010). Deutsche Diachrone Baumbank (Version 1.0). Humboldt-Universität zu Berlin.

Höhle, Tilman N. (2019). Topologische Felder. In *Beiträge zur deutschen Grammatik: Gesammelte Schriften von Tilman N. Höhle*. Ed. by Stefan Müller, Marga Reis, and Frank Richter. Berlin: Language Science Press, pp. 7–89.

Hsu, Yu-Yin (2010). Comparing Conversions of Discontinuity in PCFG parsing. In *Ninth International Workshop on Treebanks and Linguistic Theories*. Tartu, Estonia, pp. 103–113.

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

IDS (2013). Mannheimer Korpus Historischer Zeitungen und Zeitschriften. Leibniz-Institut für Deutsche Sprache, Mannheim.

Indig, Balázs (2017). Less is More, More or Less... Finding the Optimal Threshold for Lexicalization in Chunking. In *Computación y Sistemas* 21.4, pp. 637–646.

Jaeger, T. Florian (2010). Redundancy and reduction: Speakers manage syntactic information density. In *Cognitive Psychology* 61.1, pp. 23–62.

Jamshid Lou, Paria, Yufei Wang, and Mark Johnson (2019). Neural Constituency Parsing of Speech Transcripts. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Minneapolis, MN, pp. 2756–2765.

Jeffreys, Harold (1946). An Invariant Form for the Prior Probability in Estimation Problems. In *Proceedings of the Royal Society of London Series A: Mathematical and Physical Sciences* 186, pp. 453–461.

Ji, Heng and Joel Nothman (2016). Overview of TAC-KBP2016 tri-lingual EDL and its impact on end-to-end cold-start KBP. In *Proceedings of TAC*.

Jurafsky, Daniel and James H. Martin (2021). Chapter 8: Sequence Labeling for Parts of Speech and Named Entities. In *Speech and Language Processing*. Draft of September 21, 2021.

Kitaev, Nikita, Steven Cao, and Dan Klein (2019). Multilingual Constituency Parsing with Self-Attention and Pre-Training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 3499–3505.

Kitaev, Nikita and Dan Klein (2018). Constituency Parsing with a Self-Attentive Encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. Melbourne, Australia.

Klatt, Stefan (2004). Segmenting Real-Life Sentences into Topological Fields - For Better Parsing and Other NLP Tasks. In *KONVENS 2004. 7. Konferenz zur Verarbeitung natürlicher Sprache*. Vienna, Austria.

Klein, Thomas, Klaus-Peter Wegera, Stefanie Dipper, and Claudia Wich-Reif (2016). Referenzkorpus Mittelhochdeutsch (1050 – 1350) (Version 1.0). https://www.linguistics.ruhr-uni-bochum.de/rem, ISLRN 332-536-136-099-5.

Koch, Peter and Wulf Oesterreicher (1985). Sprache der Nähe – Sprache der Distanz: Mündlichkeit und Schriftlichkeit im Spannungsfeld von Sprachtheorie und Sprachgeschichte. In *Romanistisches Jahrbuch* 36, pp. 15–43.

Koch, Peter and Wulf Oesterreicher (2007). Schriftlichkeit und kommunikative Distanz. In *Zeitschrift für germanistische Linguistik* 35, pp. 246–275.

Kok, Daniël de and Erhard Hinrichs (2016). Transition-based dependency parsing with topological fields. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Berlin, Germany, pp. 1–7.

Krielke, Marie-Pauline, Luigi Talamo, Mahmoud Fawzi, and Jörg Knappen (2022). Tracing Syntactic Change in the Scientific Genre: Two Universal Dependency-parsed Diachronic Corpora of Scientific English and German. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference (LREC)*. Marseille, France, pp. 4808–4816.

Kriz, Milan (2011). Automata Editor v 2.0.

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

Lenerz, Jürgen (1977). Zur Abfolge nominaler Satzglieder im Deutschen. Tübingen: Narr.

Levy, Roger and T. Florian Jaeger (2007). Speakers optimize information density through syntactic reduction. In *Advances in Neural Information Processing Systems 19*. Cambridge, MA: MIT Press, pp. 849–856.

Lison, Pierre and Jörg Tiedemann (2016). Opensubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC)*. Portorož, Slovenia, pp. 923–929.

Lüdeling, Anke, Carolin Odebrecht, Thomas Krause, Gohar Schnelle, and Fischer Catharina (2022). RIDGES Herbology (Version 9.0). Humboldt-Universität zu Berlin.

Lühr, Rosemarie, Vera Faßhauer, Daniela Prutscher, and Henry Seidel (2013). Fürstinnenkorrespondenz (Version 1.1). Universität Jena.

Mahlow, Cerstin and Michael Piotrowski (2010). Noun phrase chunking and categorization for authoring aids. In *10. Konferenz zur Verarbeitung Natürlicher Sprache (KONVENS 2010)*. Saarbrücken, Germany, pp. 57–65.

Manning, Chris (2006). Doing Named Entity Recognition? Don't optimize for F1.

Manning, Christopher D. (2011). Part-of-speech tagging from 97% to 100%: is it time for some linguistics? In *International conference on intelligent text processing and computational linguistics*. Springer, pp. 171–189.

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

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

Müller, Frank Henrik and Tylman Ule (2002). Annotating Topological Fields and Chunks - and Revising POS Tags at the Same Time. In *COLING 2002: The 19th International Conference on Computational Linguistics*. Taipei, Taiwan.

Müller, Karin (1990). "Schreibe wie du sprichst!" Eine Maxime im Spannungsfeld von Mündlichkeit und Schriftlichkeit: Eine historische und systematische Untersuchung. Frankfurt a. M.: Lang.

Neumann, Günter, Christian Braun, and Jakub Piskorski (2000). A Divide-and-Conquer Strategy for Shallow Parsing of German Free Texts. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*. Seattle, Washington, pp. 239–246.

Ortmann, Katrin (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 (LaTeCH-CLfL)*. Barcelona, Spain (online), pp. 10–18.

Ortmann, Katrin (2021a). Chunking Historical German. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*. Reykjavik, Iceland (online), pp. 190–199.

Ortmann, Katrin (2021b). Automatic Phrase Recognition in Historical German. In *Proceedings of the 17th Conference on Natural Language Processing (KONVENS 2021)*. Düsseldorf, Germany, pp. 127–136.

Ortmann, Katrin (2022). Fine-Grained Error Analysis and Fair Evaluation of Labeled Spans. In *Proceedings of the 13th Language Resources and Evaluation Conference (LREC)*. Marseille, France, pp. 1400–1407.

Ortmann, Katrin and Stefanie Dipper (2019). Variation between Different Discourse Types: Literate vs. Oral. In *Proceedings of the NAACL-Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*. Minneapolis, MN, pp. 64–79.

Ortmann, Katrin and Stefanie Dipper (2020). Automatic Orality Identification in Historical Texts. In *Proceedings of the 12th Language Resources and Evaluation Conference (LREC)*. Marseille, France, pp. 1293–1302.

Ortmann, Katrin and Stefanie Dipper (forthcoming). Nähetexte automatisch erkennen: Entwicklung eines linguistischen Scores für konzeptionelle Mündlichkeit in historischen Texten.

Ortmann, Katrin, Adam Roussel, and Stefanie Dipper (2019). Evaluating Off-the-Shelf NLP Tools for German. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS)*. Erlangen, Germany, pp. 212–222.

Ortmann, Katrin, Sophia Voigtmann, Stefanie Dipper, and Augustin Speyer (2022). An Information-Theoretic Account of Constituent Order in the German Middle Field. In *CL Poster Session, DGfS 2022*. Tübingen, Germany.

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

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

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

Paul, Hermann (2007). Mittelhochdeutsche Grammatik. Berlin, Boston: De Gruyter.

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

Petran, Florian (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)*. Lisbon, Portugal, pp. 75–86.

Petrov, Slav, Leon Barrett, Romain Thibaux, and Dan Klein (2006). Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Sydney, Australia, pp. 433–440.

Pinto, Alexandre, 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)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 3:1–3:16.

Poschmann, Claudia and Michael Wagner (2016). Relative clause extraposition and prosody in German. In *Natural Language & Linguistic Theory* 34, pp. 1021–1066.

Potthast, Martin, Benno Stein, Alberto Barrón-Cedeño, and Paolo Rosso (2010). An evaluation framework for plagiarism detection. In *Coling 2010: Poster Volume*. Beijing, China, pp. 997–1005.

Qi, Peng, 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. Online, pp. 101–108.

R Core Team (2018). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria.

Rafferty, Anna N. and Christopher D. Manning (2008). Parsing Three German Treebanks: Lexicalized and Unlexicalized Baselines. In *Proceedings of the ACL-08: HLT Workshop on Parsing German (PaGe-08)*. Columbus, OH, pp. 40–46.

Read, Jonathon, Erik Velldal, Lilja Øvrelid, and Stephan Oepen (2012). UiO1: Constituent-based discriminative ranking for negation resolution. In *First Joint Conference on Lexical and Computational Semantics (SemEval 2012)*. Montreal, Canada, pp. 310–318.

Rehbein, Ines and Josef van Genabith (2007). Treebank Annotation Schemes and Parser Evaluation for German. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Prague, Czech Republic, pp. 630–639.

Richter, Günther (1985). Einige Anmerkungen zur Norm und Struktur des gesprochenen Deutsch. In *Deutsch als Fremdsprache* 22.3, pp. 149–153.

Röder, Michael, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo (2018). Gerbil – Benchmarking Named Entity Recognition and Linking Consistently. In *Semantic Web* 9.5, pp. 605–625.

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

Sahel, Said (2015). Zur Ausklammerung von Relativsätzen und Vergleichsphrasen im frühen Neuhochdeutschen (1650–1800). In *Das Nachfeld im Deutschen. Theorie und Empirie.* Ed. by Hélène Vinckel-Roisin. Berlin/Boston: De Gruyter, pp. 165–183.

Sang, Erik F. Tjong Kim 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*. Lisbon, Portugal, pp. 127–132.

Sapp, Christopher D. (2014). Extraposition in Middle and New High German. In *The Journal of Comparative German Linguistics* 17.2, pp. 129–156.

Schildt, Joachim (1976). „Zur Ausbildung der Satzklammer“. In *Zur Ausbildung der Norm in der deutschen Literatursprache auf der syntaktischen Ebene (1470-1730)*. Ed. by Gerhard Kettmann and Joachim Schildt. Berlin: Akademieverlag, pp. 235–284.

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

Schmidt, Thomas and Kai Wörner (2014). „EXMARaLDA“. In *The Oxford Handbook of Corpus Phonology*. Oxford University Press, pp. 402–419.

Schneider, Gerold, Hans Martin Lehmann, and Peter Schneider (2015). Parsing early and late modern English corpora. In *Literary and Linguistic Computing* 30.3, pp. 423–439.

Shannon, Claude E. (1948). A Mathematical Theory of Communication. In *The Bell System Technical Journal* 27.3, pp. 379–423.

Shao, Yan, 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*. Taipei, Taiwan, pp. 86–90.

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

Speyer, Augustin (2016). Die Entwicklung der Nachfeldbesetzung in verschiedenen deutschen Dialekten: Informationsdichte und strukturelle Verschiedenheit. In *Zeitschrift für Dialektologie und Linguistik* Beiheft 165, pp. 137–157.

Strunk, Jan (2014). A statistical model of competing motivations affecting relative clause extraposition in German. In *Competing Motivations in Grammar & Usage*. Ed. by B. MacWhinney, A. Malchukov, and E. Moravcsik. Oxford: Oxford University Press, pp. 88–106.

Sun, Xu, 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)*. Manchester, UK, pp. 841–848.

Swayamdipta, Swabha, Matthew Peters, Brendan Roof, Chris Dyer, and Noah A. Smith (2019). Shallow Syntax in Deep Water. arXiv preprint arXiv:1908.11047.

Takada, Hiroyuki (1998). Grammatik und Sprachwirklichkeit von 1640-1700. Zur Rolle deutscher Grammatiker im schriftsprachlichen Ausgleichsprozeß. Berlin, New York: Max Niemeyer Verlag.

Tang, Min, Xiaoqiang Luo, and Salim Roukos (2002). Active learning for statistical natural language parsing. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, PA, pp. 120–127.

Telljohann, Heike, 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.

TIGER Project (2003). TIGER Annotationsschema. Universität des Saarlands, Universität Stuttgart, Universität Potsdam.

Tjong Kim Sang, Erik, Marcel Bollmann, Remko Boschker, Francisco Casacuberta, Feike Dietz, Stefanie Dipper, Miguel Domingo, Rob van der Goot, Marjo van Koppen, Nikola Ljubešić, Robert Östling, Florian Petran, Eva Pettersson, Yves Scherrer, Marijn Schraagen, Leen Sevens, Jörg Tiedemann, Tom Vanallemeersch, and Kalliopi Zervanou (2017). The CLIN27 shared task: Translating historical text to contemporary language for improving automatic linguistic annotation. In *Computational Linguistics in the Netherlands Journal* 7, pp. 53–64.

Tjong Kim Sang, Erik F. and Fien De Meulder (2003). Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*. Edmonton, Alberta, Canada, pp. 142–147.

Tomczyk-Popińska, Ewa (1987). Linguistische Merkmale der deutschen gesprochenen Standardsprache. In *Deutsche Sprache: Zeitschrift für Theorie, Praxis, Dokumentation* 15, pp. 336–375.

Uszkoreit, Hans, Thorsten Brants, Denys Duchier, Brigitte Krenn, Lars Konieczny, Stephan Oepen, and Wojciech Skut (1998). Studien zur performanzorientierten Linguistik: Aspekte der Relativsatzextraposition im Deutschen. In *Kognitionswissenschaft* 7, pp. 129–133.

Van Asch, Vincent and Walter Daelemans (2009). Prepositional phrase attachment in shallow parsing. In *Proceedings of the International Conference RANLP-2009*. Borovets, Bulgaria, pp. 12–17.

Veenstra, Jorn, Frank Henrik Müller, and Tylman Ule (2002). Topological Field Chunking for German. In *Proceedings of the 6th Conference on Natural Language Learning*. Taipei, Taiwan, pp. 1–7.

Vilares, David and Carlos Gómez-Rodríguez (2020). Discontinuous Constituent Parsing as Sequence Labeling. arXiv preprint arXiv:2010.00633.

Vinckel, Hélène (2006). Die diskursstrategische Bedeutung des Nachfelds im Deutschen. Wiesbaden: Deutscher Universitäts-Verlag.

Vinckel-Roisin, Hélène (2015). Das Nachfeld im Deutschen. Theorie und Empirie. Berlin, Boston: De Gruyter.

Voigtmann, Sophia and Augustin Speyer (2021a). Information density and the extraposition of German relative clauses. In *Frontiers in Psychology*, pp. 1–18.

Voigtmann, Sophia and Augustin Speyer (2021b). Information density as a factor for syntactic variation in Early New High German. In *Proceedings of Linguistic Evidence 2020*. Tübingen, Germany.

Voigtmann, Sophia and Augustin Speyer (forthcoming). Where to place a phrase? An informational and generative approach to phrasal extraposition. In *JOURNAL of HISTORICAL SYNTAX*.

Waldenberger, Sandra, Stefanie Dipper, and Ilka Lemke (2021). Towards a broad-coverage graphemic analysis of large historical corpora. In *Zeitschrift für Sprachwissenschaft* 40.3, pp. 401–420.

Wasow, Thomas (1997). Remarks on grammatical weight. In *Language variation and change* 9.1, pp. 81–105.

Weber, Sabrina (2019). The acceptability of extraposition of PPs out of NP in German. In *Proceedings of Linguistic Evidence 2018: Experimental Data Drives Linguistic Theory*. Tübingen, Germany, pp. 63–84.

Wegera, Klaus-Peter, Hans-Joachim Solms, Ulrike Demske, and Stefanie Dipper (2021). Referenzkorpus Frühneuhochdeutsch (1350 – 1650), Version 1.0. https://www.linguistics.ruhr-uni-bochum.de/ref/, ISLRN 918-968-828-554-7.

Weiß, Helmut (2005). Von den vier Lebensaltern einer Standardsprache. In *Deutsche Sprache: Zeitschrift für Theorie, Praxis, Dokumentation* 33, pp. 289–307.

Wöllstein, Angelika (2018). Topologisches Satzmodell. In *Syntaxtheorien. Analysen im Vergleich*. Ed. by Jörg Hagemann and Sven Staffeldt. 2., aktualisierte Auflage. Tübingen: Stauffenburg, pp. 145–166.

Yang, Jie, 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)*. Santa Fe, New Mexico, USA, pp. 3879–3889.

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

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

Zifonun, Gisela, Ludger Hoffmann, and Bruno Strecker (1997). Grammatik der deutschen Sprache. Berlin, New York: De Gruyter.

This document was created with LaTeX, based on a template by Arne Köhn (`https://github.com/akoehn/thesis-template`).