Domain-Specific Considerations in the Preparation of Specialized Corpora: A Case Study on a Corpus of German Sermons

Cora Haiber[1,2,3], **Adam Roussel**[1], **Stefanie Dipper**[1]

[1]Ruhr-University Bochum
Universitätsstraße 150
44801 Bochum
Germany
firstname.lastname@rub.de

[2]Saarland University
Campus
66123 Saarbrücken
Germany

[3]Zuse School ELIZA
Hochschulstraße 10
64289 Darmstadt
Germany

Abstract

We present a new corpus of contemporary German sermons and describe the steps taken in its preparation. We apply a semi-automatic approach to sentence segmentation, tokenization, and lemmatization, utilizing annotation guidelines that are specialized to this domain. In the process of preparing these data, we find that state-of-the-art tools for these tasks still make problematic errors, especially with non-standard data, despite apparently very high performance on common benchmarks. We obtain test scores of F_1 = 96.69 % for sentence segmentation, F_1 = 99.99 % for tokenization, and acc = 64.00 % for lemmatization with our domain-adapted models and show that domain-adaptation improves performance over state-of-the-art models for the token and sentence segmentation tasks.

Keywords: corpus preparation, language resources, tokenization, segmentation, lemmatization, semi-automatic annotation

1. Introduction

In this paper we introduce a new specialized corpus of contemporary German sermons, which currently includes a range of metadata specific to the domain, as well as semi-automatically annotated tokenization, sentence boundaries, and lemmatization. Further levels of linguistic annotations are planned, including POS, morphological features, and eventually dependency parses.

In the following, we will outline the process of preparing this corpus and laying the foundations for the annotations to come. The benchmark evaluation of these initial tasks of tokenization, sentence segmentation, and lemmatization usually results in rather high accuracy and F_1 scores, such that one might reasonably have the impression that these are solved tasks, and automatic tools for this purpose can be applied directly with few reservations.

However, when it comes to the preparation of corpora, the comparatively few errors that there are have disproportionate consequences, since tokenization, sentence segmentation, and lemmatization are tasks that are located towards the beginning of several corpus preparation pipelines. Any errors in these early steps will propagate through all subsequent layers of annotation. Furthermore, when it comes to domain-specific corpora and non-standard text types, these errors are not necessarily so few in number, as the requirements of these domains may differ significantly from the newspaper articles that still form the backbone of respective off-the-shelf models.

Thus some of the problems that we have encountered during the process of adding tokenization, sentence segmentation, and lemmatization to this corpus show that there are still improvements to be made, especially as regards the adaptation of corpus preparation tools to specialized domains.

After a review of related work in Section 2, we provide a more detailed description of the corpus and its contents in Section 3. Section 4 provides an overview of the aspects of the annotation guidelines that are more particular to this specialized domain. Section 5 will then give a detailed analysis of the process of training and evaluating algorithms and models for extending our annotations according to these specialized guidelines to the rest of the corpus. Finally, in Section 6, we will discuss some of the consequences and implications of our findings for future work and conclude. Our data, guidelines, scripts, and models are available in a GitLab repository[1].

2. Related Work

2.1. Tokenization and Sentence Segmentation

Other specialized corpora are faced with similar kinds of tokenization and segmentation problems. For instance, Dipper and Laarmann-Quante (2024) discuss some of the issues that they found during the preparation of a parsed corpus of poetry, specifically with regard to the implications of sentence segmentation for dependency parsing of the data.

[1]https://gitlab.ruhr-uni-bochum.de/compchist/slide-2026-specialized-corpora

Proceedings of the Workshop on Structured Linguistic Data and Evaluation, pages 212–223

Beißwenger et al. (2015) present guidelines for tokenization of a domain-specific corpus, namely one concerning computer-mediated communication. Their guidelines must account for a number of phenomena that are quite frequent in such data, such as URLs, hashtags, and emoticons, but not in the more standard datasets that have guided tokenization approaches traditionally, either in the form of training data or as test data for rule-based methods. Similarly to the approach we have adopted for this corpus, Beißwenger et al. (2015) employ a manual correction step, in which annotators work with a one-token-per-line plain text format.

Diewald et al. (2022) evaluate state-of-the-art tokenization tools for German in terms of adaptability for new or different corpora and show how tokenization can be specific to a certain text type. They further emphasize the importance of high-quality tokenization since tokenization errors cascade through all processing steps aligning with our interests of a sensible and reliable tokenization to act as a foundation for higher-level processing steps.

The evaluation in Ortmann et al. (2019) draws attention to the relationship between text types and the performance of off-the-shelf language processing tools. Specifically, they show how tokenization errors often correspond to conventions that are specific to a given text type.

2.2. Lemmatization

In accordance with Toporkov and Agerri (2024), we define lemmatization as producing, from a given inflected word, its lexicon form, which we call lemma. In general, lemma forms correspond to the canonical dictionary form, e.g., for verbs this is often the infinitive, for nouns the nominative singular. However, for morphologically inflected languages such as German, there are many special cases that are handled differently in different guidelines. Table 1 shows some examples from the guidelines of the two standard German corpora, TIGER (Crysmann et al., 2005) and Tüba-D/Z (Telljohann et al., 2017), together with the corresponding forms that we use in the sermons.The table shows that in TIGER, information about capitalization, which distinguishes nouns in German, is not included in the lemma.

Common modeling approaches for lemmatization can be categorized into neural, rule- or pattern-based, and lookup-based models. Lookup-based models such as the WordNet lemmatizer (Miller, 1994) rely on pre-compiled dictionaries or morphological databases and are inherently limited in coverage. Rule-based lemmatizers such as spaCy (Honnibal et al., 2020) lemmatizers apply handcrafted or automatically induced heuristics. They are highly interpretable and efficient, but may lack the flexibility to handle the complexities of morphologically rich languages such as German. Neural models such as Stanza (Qi et al., 2020) lemmatizers or Lematus (Bergmanis and Goldwater, 2018) are often the most flexible option with high coverage since they treat the task as a sequence-to-sequence model or a classification task and leverage contextual information to resolve ambiguity, but might lack reliability and linguistic interpretability (see Section 5.3.2).

Toporkov and Agerri (2024) emphasize the importance of evaluating lemmatization models on out-of-domain test data and find that models using simple UPOS tags outperform those trained with fine-grained morphological features. Dorkin and Sirts (2024) compare current approaches to lemmatization on Estonian and conclude that ensembles of different modeling techniques might be the best possible approach to the task of lemmatization as of today. Hence, we choose to expand GermaLemma (Konrad, 2019), an ensemble method for the lemmatization of part-of-speech (POS) tagged German sentences, which combines rule- and lookup based methods with a large lemma dictionary based on the TIGER corpus (Brants et al., 2004), functions from the CLiPS "Pattern" package (De Smedt and Daelemans, 2012), and an algorithm to split compounds.

3. Corpus description

The Corpus of Contemporary German-language Protestant Sermons consists of a collection of 1 068 German-language sermons published by the Zentrum für evangelische Gottesdienst- und Predigtkultur ('Center for Protestant Mass and Sermon Culture')[2] between 2011 and 2023. The sermons in our corpus are typically based on a specific Bible passage, which is explained and commented on, and some include literal citations of the respective passage. The corpus includes sermons from 109 authors who generously granted us permission to include their works, contains approximately 1.93 M tokens in total, and is available under a CC BY-NC-SA 4.0 license[3]: https://linguistics.rub.de/predigtenkorpus.

The texts are collected in their original HTML format and automatically converted to an internal JSON-based standoff format. In the process, we preserve as much information as possible from the original documents. This includes metadata pertaining to the author, the date published, and the relevant Bible passage, including a URL that points to a digital version of that Bible passage, among other things. It also includes text-structural elements, such as headings and line breaks, when these are included in the markup, and emphasized

[2] https://predigten.evangelisch.de/

[3] https://creativecommons.org/licenses/by-nc-sa/4.0/

Word class	Word forms	TIGER	Tüba-D/Z	Sermons
Articles	der / die / das 'the'	der	der / die / das	der
	ein / eine 'a'	ein	ein / eine	ein
Pronouns	manchen / manches 'some'	mancher	mancher / manches	mancher
	wer / wem / was 'who, what'	wer / wem / was	wer / wer / was	wer / wer / was
Deadj. nouns	(ein) Junger 'a youth'	junge	Junger	Junge
	Nettes '(something) nice'	nette	Nettes	Nette
Deverbal nouns	(das) Lesen 'reading'	lesen	Lesen	Lesen
	(der) Angeklagte 'the accused'	angeklagter	Angeklagter	Angeklagte
	(die/der) Vorsitzende 'the presiding (person)'	vorsitzend	Vorsitzender	Vorsitzende
Fused prep.	im 'in the'	in	in	im

Table 1: Example lemmas from the German corpora TIGER and Tüba-D/Z.

passages (using `<em>` tags), which are usually quotations from the Bible. Since the Bible passages themselves use a very different form of language than the sermons themselves, it is of particular interest for the annotation process to be able to distinguish these passages from the surrounding text, and preserving this markup makes this possible with a fair degree of reliability.

4. Annotation

All 1068 texts were initially tokenized and split into sentences using SoMaJo with its `de_CMC` ruleset (Proisl and Uhrig, 2016). Of these automatically tokenized documents, 155 randomly selected sermons were then manually corrected according to the domain-specific tokenization guidelines presented below. Later, in Section 5, these corrected data will be used in the training of segmentation and tokenization models, with which we plan to process the remaining corpus documents.

Note that the following sections don't contain the full contents of the guidelines used for the given tasks, but rather serve to give an impression of some of their finer points, i.e. the aspects that apply to the domain of sermons in particular. The full guidelines are available as part of the documentation, which can be found in the corpus repository.

4.1. Tokenization and Sentence Segmentation

In the following examples, tokens are separated by space characters, and sentences are each presented on a new line, indicated by the line numbers given at the left. Where sentences are too long to fit in a column, they are wrapped and this is indicated by ↩. Continuations of logical lines are indented to reflect that these line breaks do not correspond to sentence boundaries as usual.[4] English translations of the relevant passages are provided in italics.

Biblical citations. Citations from the Bible usually provide an abbreviation of the name of the book, a chapter number, and a verse number, i.e. `2. Kor 13,11` refers to the second letter to the Corinthians, chapter 13, verse 11. Each citation represents a complex reference to a Bible passage and should therefore be treated as a single token. One simple way to achieve this is to replace space characters (i.e., U+0020) within these reference by underscores (_, U+005F), to indicate visual spacing and also the instance's acting as a single token, as in `2._Kor_13,11`. This is similar to the format sometimes used for multi-word expressions so that they can be treated as single tokens, e.g. by `word2phrase` (Mikolov et al., 2013), the `mwetoolkit`[5] (Ramisch, 2015), and `stanza` (Qi et al., 2020).

Multiple citations should be multiple tokens. These are usually separated by punctuation marks such as the semicolon `;`. Citations and sequences of citations, when they come at the end of an otherwise complete sentence, should be considered a separate sentence:

```
1 Freuet euch !
   'Rejoice !'
2 Kap_3,1 ; 2._Kor_13,11 ; ↩
      1._Thess_5,16
```

If a citation acts as an argument, i.e. it is integrated syntactically into the surrounding text, or it is part of the title, it is not considered to be a separate sentence.

```
1 Predigt zu Lukas_17,11-19
   'Sermon on Luke_17,11-19'
```

Quoted material. Quotations (in this corpus, these are usually Biblical in origin) are segmented internally if they contain multiple sentences:

```
2
```

[4] Note that this differs from the format that was used during the annotation process, in which, for easier editing, each line contained one token, and sentence boundaries were represented by an empty line.

[5] Though note that `mwetoolkit` also supports many other formats. https://gitlab.com/mwetoolkit/mwetoolkit3

```
1 5
3 Eure Güte laßt kund sein allen ↩
     Menschen !
  'Let your kindness be known to ↩
     all people !'
4 Der Herr ist nahe !
  'The Lord is near !'
5 Tit_3,2
```

Quoted material may be preceded by introductory comments. These should be annotated as a single sentence whenever the quoted material is considered part of the previous sentence syntactically, which is usually the case.

```
1 Dem Briefpapier vertraut er an ↩
     : Paulus , berufen zum ↩
     Apostel Christi Jesu durch ↩
     ...
  'To the letterhead he entrusts ↩
     : Paul , called to be an ↩
     apostle of Christ Jesus ↩
     through ...'
```

Headers and other metadata. If the sermon contains a title, subtitle, author, etc, they are all segmented separately.

```
1 Versöhnung : Christliches ↩
     Alleinstellungsmerkmal !?
  'Reconciliation: A uniquely ↩
     Christian trait !?'
2 Hans Meyer
  'Hans Meyer'
3 Liebe Gemeinde !
  'Dear congregation !'
```

4.2. Lemmatization

Our guidelines largely follow those of TIGER. The most important differences are (see Table 1):

- Deadjectival and deverbal nouns: We follow the original upper/lower case spelling and adopt this for the lemma (see TüBa-D/Z), regardless of whether it is a lexicalized form.
- Deadjectival nouns: The lemma adopts the ending of the weakly inflected nominative singular, which is *-e* in all three genders (see TIGER). This ending is also used for nominalized participles.
- Personal pronouns: Word forms that are generally ambiguous are lemmatized with an ambiguous form: *ihm* → *er_es; ihr* → *sie_ihr*.
- Proper nouns: Unusual capitalization is normalized in the lemma, e.g. *der Heilige Geist* → *heilig; HERR* → *Herr*
- Fused prepositions: To encode the difference from normal prepositions, the lemma form corresponds to the word form: *im* → *im*.

5. Models

To scale the applicability of our guidelines to the size of the corpus, we develop models for sentence segmentation, tokenization, and lemmatization. Our overarching goals are to robustly and efficiently model the domain-specific particularities of our texts according to our guidelines and to investigate the transferability of off-the-shelf tools to a specialized domain. While we primarily deal with Biblical citations and quoted Bible passages, other particularities will emerge in other domains. What they have in common is that expert knowledge and modeling adaptations might be required to adequately process them. We ascertain that deviating from standard modeling approaches is appropriate for various domain-specific problems and that quantitative evaluation scores of state-of-the-art models are not necessarily informative of their true generalization abilities.

5.1. Segmentation

Common segmentation algorithms, such as Punkt (Kiss and Strunk, 2006), largely rely on punctuation to detect sentence boundaries, which is unsuitable for our sermons due to three fallacies. Firstly, though the texts were published in written form, many of them are more adjacent to transcripts of spoken language – manifesting in the fact that the punctuation often reflects rhetorical indications, such as pauses in speech, rather than sentence boundaries. Secondly, the texts contain numerous interjections of Biblical citations with parentheses, additional punctuation, and abbreviations of Bible chapters (e.g. *Lk.*, *Mk.*) as well as direct speech, where a punctuation mark does not necessarily reflect a sentence boundary in the matrix clause. Thirdly, the sermons frequently contain line numbers, which we choose to segment separately but which may not be separated by any punctuation mark.

5.1.1. Methods

To ensure robustness to the above particularities, we follow the approach of Frohmann et al. (2024), who present a collection of segmentation models (SaT) with less reliance on punctuation and promising experiments with Low-Rank-Adaptation (LoRA; Hu et al., 2021) for specialized domains. They train subword-based multilingual encoder language models (LMs) in a self-supervised manner on 85 languages sampled from the mC4 corpus

Split	%	#Text	#Sent	#Tok
train	70	106	12 103	208 589
val	15	23	2 668	45 287
test	15	26	2 829	47 791

Table 2: Split of manually corrected data for segmentation model development reporting percentage of data (%), number of texts (#Text), number of sentences (#Sent), and number of tokens (#Tok) per subset.

Base	F1	R	P
sat-1l	0.71	0.94	0.57
sat-3l	0.67	0.96	0.52
sat-6l	0.76	0.95	0.63
sat-9l	0.80	0.94	0.69
sat-12l	**0.81**	**0.93**	**0.71**
sat-1l-sm	0.90	0.87	0.92
sat-3l-sm	0.93	0.91	0.95
sat-6l-sm	0.93	0.94	0.93
sat-12l-sm	**0.95**	**0.94**	**0.96**

Table 3: Evaluation of pretrained SaT models on the validation set reporting base model, F1 score, recall (R), and precision (P).

(Raffel et al., 2019) and subsequently train the self-supervised models on a supervised mixture of already segmented sentences (`sm-models`). We split the 155 manually corrected sermons as presented in Table 2 and first evaluate the pretrained models on our validation set (see Table 3). Model performance is measured in terms of F1 score as in the original paper[6].

For both classes, unsupervised and supervised, we find that the largest models *sat-12l* and *sat-12l-sm* perform best and that the *sm*-versions perform better than the completely unsupervised models. We train LoRA adapters for each of the two highest-performing models on our training set, of which the size approximately compares to the 10 000 sentences that Frohmann et al. (2024) use for LoRA training.

5.1.2. Results

Matching the findings of Frohmann et al. (2024), we find that the *sm*-versions do not benefit from LoRA as much as the unsupervised models and perform slightly worse with adapters even though the base *sm*-models had significantly higher F1 scores. Furthermore, model scores only increase for a given number of epochs *k* as demonstrated in Table 4. The overall best model on the validation set consists of *sat-12l* with a LoRA head trained for 6 epochs and reaches F_1 = 96.69 % on the validation set. After epoch 6, precision improves further but at the cost of recall. A final evaluation on the test set yields F_1 = 96.86 % with 94.97 % recall and 98.82 % precision. Compared to the validation set, we obtain slightly lower recall and slightly higher precision, but overall good generalization abilities.

A qualitative analysis of the model outputs reveal that Biblical citations are still a major source of errors and are not segmented reliably.

[6] Frohmann et al. (2024) suggest character-level F1 scores for the positive labels to compare ground truth and predicted sentence boundaries. I.e. for each predicted sentence, we check whether the right sentence border is in the correct position.

Base	#Ep	F1	R	P
sat-12l	5	0.96	0.94	0.98
sat-12l	6	**0.97**	**0.95**	0.98
sat-12l	7	0.96	0.95	0.98
sat-12l	10	0.96	0.93	0.99
sat-12l	20	0.25	0.14	**0.99**
sat-12l-sm	4	0.95	0.95	0.96
sat-12l-sm	5	0.96	0.95	0.97
sat-12l-sm	6	0.96	0.94	0.98

Table 4: Evaluation of LoRA adapters with different hyperparameter configurations on the validation set, reporting base model, number of epochs trained (#Ep), F1 score, recall (R), and precision (P).

```
1 Ja, Gesetzlichkeit ist ↩
    eigentlich nichts als ↩
    Ärgernis an der " ↩
    herrlichen Freiheit der ↩
    Kinder Gottes " , wie ↩
    Paulus sagt ( Röm._8,21 ) .
  'Yes , the law is really ↩
    nothing but a hindrance to ↩
    the ‘ wonderful freedom of ↩
    the children of God ' , as ↩
    Paul says ( Röm._8,21 )'
2 Ich lese den gesamten ↩
    Textabschnitt und stelle ↩
    uns damit die Szene ↩
    bildlich vor Augen ↩
    Mi_6,1-8.
  'I read the entire passage and ↩
    use it to vividly picture ↩
    the scene in our minds ↩
    Mi_6-,18.'
3 Wir haben also heute ↩
```

```
      interessanterweise einen ↩
      alttestamentlichen Text zum ↩
      Osterfest !
   'So , interestingly enough , ↩
      today we have an Old ↩
      Testament passage about ↩
      Easter !'
4  Jesaja_25,8-9
5  Es ist Matthäus , der sie ↩
      schildert : " Und Petrus ↩
      ging hinaus und weinte ↩
      bitterlich . " .
   'It is Matthew who describes ↩
      it : "And Peter went out ↩
      and wept bitterly. "'
6  ( Mt_26,75 )
```

According to the guidelines, each of the examples above should be segmented into two sentences, but the segmentation model only splits the sentences in 3–4 and 5–6 correctly but not in 1 and 2. There is no clear tendency as to which cases are segmented incorrectly other than punctuation: Citations are usually segmented correctly when there is no subsequent punctuation mark and usually segmented incorrectly when there is.

5.2. Tokenization

Sensible tokens are fundamental to semantically motivated tasks such as metaphor detection, where annotators as well as classifiers rely on a reliable lexicon of candidate words. The assumption that tokenization is a solved task based on very high quantitative evaluation metrics is challenged by the intricacies of domain-specific text.

5.2.1. Methods

We apply the neural tokenizer model of Qi et al. (2018), who propose the combination of a bidirectional LSTM and a 1-layer CNN, and provide a training interface for custom models within the Stanza framework (Qi et al., 2020). The default model for German is trained on a dataset sampled from the GSD (Nivre et al., 2016) and the HDT (Borges Völker et al., 2019) treebanks and reaches F_1 = 99.62 % on a GSD test set and F_1 = 100.00 % on a HDT test set. We evaluate the off-the-shelf model on the same validation set of sermons that was used for the development of the segmentation model and obtain a comparatively low accuracy of 95.83 %, even when citations from the Bible are already joined by underscores. A qualitative analysis reveals that the model tends to split these tokens specifically at underscores and other punctuation. Since the models distributed with Stanza are not inherently re-trainable, we default to training a new model with the same architecture on a mixture of the GSD training data and our training set of sermons.

5.2.2. Results

The best model with a mixing ratio of approximately 2:1 of sermons and GSD sentences illustrated in Table 6 yields F_1 = 99.99 % on the test set[7]. We evaluate qualitatively on our test set in two settings: *simple*, where Biblical citations have already been preprocessed and joined by underscores, and *hard*, where this is not the case and subtokens of the citations are split by spaces. We utilize our manually corrected sermons for this, but for future utility this preprocessing step could be done with either a rule-based script or a specialized model for named entity recognition. In the *easy* setting, the tokenizer learns to not split citations at underscores, validating our approach to adapt the default model to our domain. However in the *hard* setting, citations are split into subtokens and the model does not learn to recognize them as constructs adjacent to multiword expressions. We present a few representative examples in Table 5.

5.3. Lemmatization

As Wartena (2019) states, state-of-the-art lemmatizers for German are often rule-based, which we can attribute to the fact that lemmatization is an analytical step heavily depending on morphological analysis. To map a word form onto a lemma is to make a claim about how that word form is related to others on a linguistic level, and one must highlight the contrast between lemmatization and stemming here. For stemming, it is not necessary for the string to correspond to a word one can find in the lexicon, since it primarily aims at mapping related words to a common string to counteract data sparsity. However in lemmatization, we claim to map words to their lexicon form, which is why a non-existent lemma is always wrong whereas a stem that does not exist as a word can serve its purpose regardless. This is why linguistically sound lemmatization is foundational for tasks like identifying candidate words acting as metaphors, where we depend on true lexicon forms with semantic meaning.

This contrast made, we turn our attention to statistical lemmatization models and find that they are often not as reliable as their quantitative performance metrics suggest. While Stanza (Qi et al., 2018) reports F_1 = 97.23 % to 98.04 %[8] for their default

[7] Here, we follow Stanza's internal evaluation approach yielding character-level F1 scores.

[8] combined_charlm: https://stanfordnlp.github.io/stanza/performance.html#system-performance-on-ud-treebanks

Gold	Easy	Hard
Jer_7,18	Jer_7,18	Jer ; 7,18
Lukas_15,11-24	Lukas_15,11-24	Lukas ; 15,11-24
2._Buch_Mose	2._Buch_Mose	2. ; Buch ; Mose

Table 5: Examples of tokenization of Biblical citations in *easy* vs. *hard* scenarios as produced by our custom tokenization model. ";" represents the beginning of a new token.

Sermons		GSD		
#Sent	#Char	#Sent	#Char	F1-val
12,103	927,614	13,813	1,388,381	57.11
12,103	927,614	6,906	656,466	**99.99**

Table 6: Training mixture for Stanza tokenizer model with varying proportions of GSD support reporting number of sentences (#Sent) and characters (#Char) per dataset and F1 score on the validation set of sermons.

Form	Stanza	spaCy
Jesu	Jesu	Jesu
Jesus	Jesus	jesus
Gottes	Gott	gott
Bergpredigt	Bergpredigt	bergpredigen
Anfällen	Anfall	anfällen
reifen	reif	reifen
trete	treten	tren
5a	zeichnen	5a
Geist	Geist	isen
dass	dass	daß
dass	dass	dass
am	am	an
ihm	er	ihm
mir	ich	mir
aller	alle	aller
allem	alle	alle
was	was	wer

Table 7: Illustrating examples of unreliable performance of neural SOTA models on German lemmatization reporting token form, POS tag according to STTS, Stanza output, and spaCy output.

lemmatization model for German and spaCy (Honnibal et al., 2020) 98.00 % accuracy[9], Table 7 illustrates how high performing models fail at cases that are not inherently difficult or "just" domain-specific. We observe four classes of common mistakes:

1. Nouns specific to the religious domain such as *Jesus* or *Bergpredigt* 'Sermon on the Mount', which are either just lowercased or lemmatized as verbs by spaCy.
2. Non-sensical lemmas for various classes such as *5a* → *zeichnen* 'draw' or *Geist* 'spirit' → *isen*.
3. Unreliable predictions, where the same model produces different outputs for arguably the same input. For example, spaCy predicts *daß* for *dass* in some cases and *dass* correctly in others.
4. Classes where outputs differing between models can be attributed to different guidelines, e.g. how to lemmatize preposition–article contractions ("Schmelzformen") and pronouns.

5.3.1. Methods

We target each of the above types of errors with our implementation of GermaLemma++ (GL++), an ensemble of rule-based, lookup-based, and neural lemmatization methods and follow in the footsteps of Ortmann (2019), who adapts GermaLemma (Konrad, 2019) with additional rules for pronoun handling and introduces spaCy as a large-scale fallback to the original rule-based model. For GL++, we propose extensive handling of proper nouns, allow for domain adaptation, add additional rules for pronouns, adjectives, adverbs, articles, as well as conjunctions, and optimize the model choice for the fallback. The new GL++ architecture is illustrated in Figure 1 in the appendix.

Both preceding versions propose the original word form as the lemma for proper nouns, i.e. words that receive the POS tag *NE* from the Stuttgart-Tübingen-Tagset (STTS; Schiller et al., 1999). Cases such as *Jesu* or *Gottes* suggest that this solution is not always suitable. We utilize Kaikki (Ylonen, 2022), a digital archive providing 13 631 German proper nouns, and provide GL++ with a look-up table based on this resource. Only when the word is out-of-vocabulary do we assign the original word form as the lemma.

Since GermaLemma already has a dictionary-

[9] de_core_news_sm: https://spacy.io/models/de

based component based on the TIGER corpus, we exploit this method further for domain adaptation and provide a Python interface to add lists of custom mappings to the dictionary. Since we are working with sermons, we additionally provide a dictionary already adapted to the religious domain.

We extend the rule-based pronoun handling of Ortmann (2019) with additional rules for pronouns, corresponding to the STTS tags *PIAT*, *PIS*, *PDAT*, *PWS*, and *PRELS* according to the guidelines presented in Section 4.2. Furthermore, we adopt a rule-based and non-invasive approach to the preposition–article contractions as well as conjunctions to leave them unchanged and map conventionalized adjective constructions such as *desweiteren* and *imfolgenden* to *weiter* and *folgend* respectively.

Ortmann (2019) proposes spaCy as a rule-based fallback model for cases where neither the original GermaLemma nor the additional rules could assign a lemma. However qualitative observations as in Table 7 suggest that these edge cases might be exactly what spaCy often fails at. We conduct a quantitative analysis contrasting GL++ with spaCy and with Stanza as a fallback. Manually curating an evaluation set of 19 145 tokens shows that the latter provides the correct lemma in 34 and the former in only 11 out of 51 cases where the models differed in their predictions. Since the number of differing predictions is not high enough to make a definite judgement, we provide an option for choosing between the two fallback models in our GL++ implementation, but proceed with Stanza as the fallback for our corpus of sermons.

5.3.2. Results

For the final evaluation, we manually curate a set of 200 words from unseen sermons and contrast the performance in lemmatization of the default models provided by Stanza and spaCy vs. GL++. The 200 words are randomly selected content words whose lemmatization differs between models from 20 different sermons each providing 10 randomly chosen words.

We provide two gold standard versions: (i) “strict”: the lemma form that complies with our guidelines; (ii) “flexible”: other lemma forms that comply with the other guidelines. Table 9 shows that spaCy performs by far the worst in both conditions. Stanza clearly delivers the best results, even in the “strict” condition, for which GL++ is specifically designed. If we turn to accuracy separated by part of speech (see Table 10), we obtain that all models perform similarly well with adjectives. spaCy’s problems lie mainly with nouns and verbs. GL++ achieves exactly the same results as Stanza with adjectives and verbs, but performs significantly worse with nouns.

In order to assess the severity of the deviations in the lemmas, we classify errors into four types, see Table 8. For types 1–3, the corresponding lemmas can still be guessed (e.g., by a human annotator), whereas this is no longer possible for type 4. The critical errors are therefore the “strong” errors, as these typically involve the assignment of completely incorrect lemmas. Table 11 shows how often each error type occurs in the different models. The “strong” type occurs most frequently with spaCy, but also with the other models. Table 12 shows some examples of the different models. spaCy (and also GL++) quite often produce non-words, while Stanza more often assigns incorrect existing lemmas.

6. Conclusion

In this paper, we introduced a new corpus of contemporary German sermons and described some of the fundamental corpus preparation steps that we have taken, for which we adopt a semi-automatic annotation approach. We described the annotation guidelines that we applied in this corpus, which reflect the specific properties of this text type. Our analysis of established tools for tokenization, sentence segmentation, and lemmatization shows that these tasks are not yet as solved as they often seem and that it is worthwhile to approach them in a domain-sensitive way.

Our evaluation of domain-adaptable neural models for tokenization and sentence segmentation show that such models can be successfully adapted to our sermon-specific guidelines, but only to a certain extent. In both tasks, there are certain error classes that remain persistent. With respect to lemmatization, Stanza appears to provide good overall lemmatization performance, though the error rates are still quite high. A frequent error is the generation of non-existent lemmas: Lemmatizers should do more to avoid non-words, ideally linking to specific entries in a lexicon as opposed to providing strings alone, which are often ambiguous. If there is no clear solution, a model should not predict any lemma, and this would do less damage than making a misleading prediction.

In general, we wish to emphasize the importance of a domain-specific approach for more correct annotations, since different domains have different properties and requirements, and different corpora are structured in different ways. The pursuit of high-quality annotation is particularly important in these early stages of corpus preparation, since errors here will propagate through all subsequent stages of processing, which can then have unpredictable effects on derived automatic annotations as well as on any statistical analysis one carries out on the basis of such corpus data.

Error Type	Description	Example
marginal	Incorrect capitalization Or correct stem (vowel + consonants) with incorrect inflection but nom. sg.	*Striche* → **strich* /✓ *Strich* *rechte* → **rechter* /✓ *recht*
small	Correct stem, incorrect inflection	*Abschieden* → **Abschiede* /✓ *Abschied*
medium	Incorrect stem but lemma can be guessed	*blies* → **blies* /✓ *blasen*
strong	Lemma distorted, e.g., other, unrelated lemma	*Zelt* → **zeln* /✓ *Zelt*

Table 8: Error types of incorrect lemma forms.

Model	Strict	Flex
spaCy	0.26	0.38
Stanza	0.72	0.81
GL++	0.64	0.69

Table 9: Accuracies (strict & flexible) per model.

Model	NOUN	ADJ	VERB
spaCy	0.29	0.76	0.23
Stanza	0.86	0.73	0.82
GL++	0.65	0.73	0.82

Table 10: Accuracies (flexible) per model and main parts of speech.

Word form	✓Lemma	* Lemma
spaCy:		
abzugleiten	abgleiten	Abzugleit
mitgeprägt	mitprägen	mitgepragen
begehrenswert	begehrenswert	begehrenswern
Zelt	Zelt	zeln
Stanza:		
http://wiki.vol[...]	(same)	Gewinntomaniet
betrüge	betrügen	betragen
Wollen	Wollen	Wolle
GL++:		
bitter	bitter	bitt
abzugleiten	abgleiten	abzugleien
Wüsst	wissen	wüsen

Table 12: Example of strong deviations of lemmas.

7. Acknowledgements

This research is funded by the Deutsche Forschungsgemeinschaft (DFG) SFB1475 – project number 441126958. Cora Haiber is supported by the Konrad Zuse School of Excellence in Learning and Intelligent Systems (ELIZA) through the DAAD programme Konrad Zuse Schools of Excellence in Artificial Intelligence, sponsored by the Federal Ministry of Education and Research.

Furthermore, we thank Tamara Abdul Majeed and Sophie Kratz for their efforts in manual annotation and correction.

Model	marg.	small	medium	strong
spaCy	10	20	78	17
Stanza	2	10	12	10
GL++	3	37	15	7

Table 11: Frequencies of error types per model.

8. Bibliographical Referencess

Michael Beißwenger, Sabine Bartsch, Stefan Evert, and Kay-Michael Würzner. 2015. Richtlinie für die manuelle Tokenisierung von Sprachdaten aus Genres internetbasierter Kommunikation: Guideline document from the Empirikom shared task on automatic linguistic annotation of internet-based communication (EmpiriST 2015).

Toms Bergmanis and Sharon Goldwater. 2018. Context sensitive neural lemmatization with Lematus. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1391–1400, New Orleans, Louisiana. Association for Computational Linguistics.

Emanuel Borges Völker, Maximilian Wendt, Felix Hennig, and Arne Köhn. 2019. HDT-UD: A very large Universal Dependencies treebank for German. In *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, pages 46–57, Paris, France. Association for Computational Linguistics.

Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen, Esther König, Wolfgang Lez-

ius, Christian Rohrer, George Smith, and Hans Uszkoreit. 2004. TIGER: Linguistic interpretation of a German corpus. *Journal of Language and Computation*, 2(4):597–620.

Berthold Crysmann, Silvia Hansen-Schirra, George Smith, and Dorothea Ziegler-Eisele. 2005. *TIGER Morphologie-Annotationsschema*. Projekt TIGER, Universität des Saarlandes, Universität Stuttgart, Universität Potsdam.

Tom De Smedt and Walter Daelemans. 2012. Pattern for Python. *Journal of Machine Learning Research*, 13(66):2031–2035.

Nils Diewald, Marc Kupietz, and Harald Lüngen. 2022. Tokenizing on scale: Preprocessing large text corpora on the lexical and sentence level. In *Dictionaries and Society. Proceedings of the XX EURALEX International Congress*, pages 208–221, Mannheim. IDS-Verlag.

Stefanie Dipper and Ronja Laarmann-Quante. 2024. UD for German poetry. In *Proceedings of the 4th International Conference on Natural Language Processing for Digital Humanities*, pages 177–188, Miami, USA. Association for Computational Linguistics.

Aleksei Dorkin and Kairit Sirts. 2024. Comparison of current approaches to lemmatization: A case study in Estonian. _eprint: 2404.15003.

Markus Frohmann, Igor Sterner, Ivan Vulić, Benjamin Minixhofer, and Markus Schedl. 2024. Segment Any Text: A universal approach for robust, efficient and adaptable sentence segmentation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11908–11941, Miami, Florida, USA. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength natural language processing in Python.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. LoRA: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685.

Tibor Kiss and Jan Strunk. 2006. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525.

Markus Konrad. 2019. GermaLemma: A lemmatizer for German language text. Publication Title: GitHub repository.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 27th International Conference on Neural Information Processing Systems – Volume 2*, NIPS'13, pages 3111–3119. Curran Associates Inc.

George A. Miller. 1994. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8–11, 1994*.

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

Katrin Ortmann. 2019. GermaLemma++. Publication Title: GitHub repository.

Katrin Ortmann, 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 2019): Long Papers*, pages 212–222, Erlangen, Germany. German Society for Computational Linguistics & Language Technology.

Thomas Proisl and Peter Uhrig. 2016. SoMaJo: State-of-the-art tokenization for German web and social media texts. In *Proceedings of the 10th Web as Corpus Workshop (WAC-X) and the EmpiriST Shared Task*, pages 57–62, Berlin. Association for Computational Linguistics.

Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D. Manning. 2018. Universal Dependency parsing from scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 160–170, Brussels, Belgium. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. _eprint: 2003.07082.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Carlos Ramisch. 2015. *Multiword Expressions Acquisition: A Generic and Open Framework*. Springer International Publishing.

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

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

Olia Toporkov and Rodrigo Agerri. 2024. On the role of morphological information for contextual lemmatization. *Computational Linguistics*, 50(1):157–191.

Christian Wartena. 2019. A probabilistic morphology model for German lemmatization. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019)*, pages 40–49. Fakultät III – Medien, Information und Design.

Tatu Ylonen. 2022. Wiktextract: Wiktionary as machine-readable structured data. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1317–1325, Marseille, France. European Language Resources Association.

A. Appendix

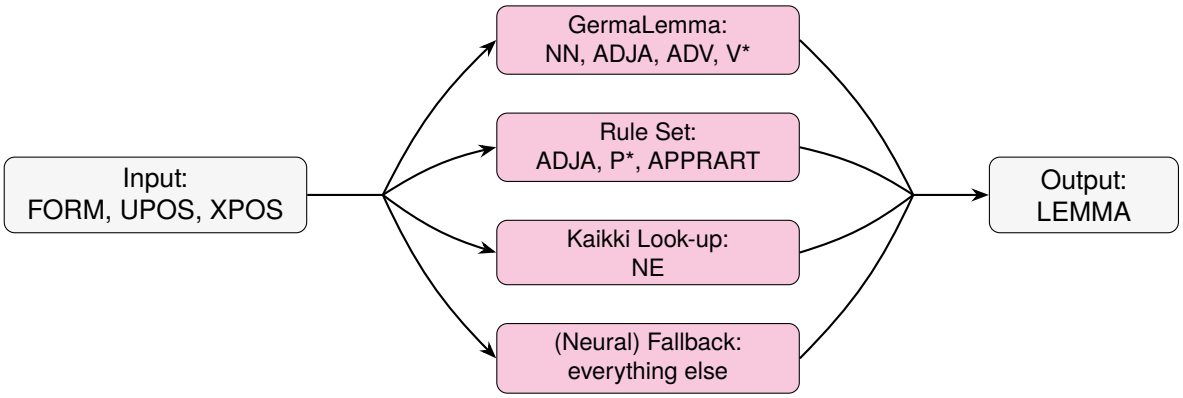

Figure 1: GL++ ensemble combining rules, lookup, and a neural model.