Automatic Error Correction for PoS in Historical Text

Ines Rehbein **Laura Duve** **Antje Dammel**
Institut für Germanistik, Abteilung Sprachwissenschaft
Universität Münster
{irehbein|laura.duve|dammel}@uni-muenster.de

Abstract

Building resources for historical text is a time-consuming process and requires a substantial amount of manual effort, given the sparsity of tools and training data for historical time periods. In the paper, we focus on the problem of part-of-speech tagging of Early New High German documents and evaluate methods to reduce human effort. Specifically, we present a pipeline and annotation interface that supports manual post-correction of automatically predicted tags and show that using open-weight LLMs for post-correction of PoS tags is feasible at least for some error types, when breaking down the task into well-defined problems.

1 Introduction

The creation of language ressources for historical text is a time-consuming process and requires a substantial amount of manual effort. Typically, the transcripts need to be digitized before they can be processed by NLP tools. Furthermore, we cannot expect the same quality from NLP tools as we get for contemporary standard German, given that historical documents often include considerable orthografic variation. This results in a sparsity of high-quality annotated training data for historical time periods even for basic tasks like part-of-speech (PoS) tagging.

Several resources for diachronic German have been built, e.g., the Reference corpora of Old German (ReA) (Donhauser, 2015), Middle High German (ReM) (Petran et al., 2016) and Early New High German (ReF),[1] the historical corpora in the German Text Archive (DTA) and the RIDGES corpus (Odebrecht et al., 2017). The corpora, however, do not use the same tagsets for analysing parts-of-speech. The DTA corpora and RIDGES include automatically predicted PoS tags based on the Stuttgart-Tübingen TagSet (STTS) (Schiller et al., 1999) which encodes 54 different PoS categories. The reference corpora, on the other hand, were annotated partly manually but mostly automatically using the HiTS tag set (Dipper et al., 2013), a PoS tag set developed for diachronic German that includes 84 fine-grained part-of-speech categories. This detailed classification results in valuable resources for linguistic research, however, the high number of fine-grained tag distinctions poses a challenge for using this tagset in research projects that cannot devote much time and resources for the creation of a high-quality PoS layer.

This is the scenario we address in our work. We aim to support research projects with limited resources for linguistic annotation. Our main goal, therefore, is not to develop new algorithms for detecting and correcting PoS errors, but rather to apply existing methods and test their potential for saving time and improving data quality under realistic conditions. Our main findings show that significant improvements can be achieved by fully automatic rule-based identification and correction of PoS errors. Applying LLMs for error detection can yield some further improvements, however, only when targeting specific error types.

The contributions of our work are threefold. First, we describe a workflow for efficient PoS error correction that can be used to improve PoS quality in newly created historical corpora when only a limited amount of time and human ressources are available. Second, we provide a PoS correction interface, implemented as a Streamlit app, that can be used by other projects for PoS correction and similar tasks. Finally, we present an evaluation of LLM-based PoS error correction for historical documents.[2]

[1] ReF is based on the "Bonner Frühneuhochdeutschkorpus" (FnhdC) (Fisseni, 2017; Herbers et al., 2021)

[2] Our code and annotation interface can be found here: https://github.com/irehbein/HisPos.

2 Related Work

We now review related work on PoS tagging for historical German (§2.1), on PoS error detection (§2.2) and on using LLMs as PoS taggers (§2.3).

2.1 PoS tagging of historical texts

Many studies have developed NLP pipelines for processing historical text, including normalisation (see, e.g., Bollmann et al. (2011); Jurish (2012); Bollmann et al. (2019)) and PoS tagging (Dipper, 2011; Jurish, 2003; Bollmann, 2013).

Jurish (2003) combines a rule-based morphological analyser with Hidden Markov Models (HMM) in a hybrid approach while Bollmann (2013) shows the importance of normalisation to improve PoS tagging of historical documents from the 15-18th century. Schulz and Kuhn (2016) address the problem of data sparsity, using unsupervised and semi-supervised methods. Finally, Dipper et al. (2013) have adapted the STTS, the quasi-standard for German PoS tagging, for the analysis of historical language, resulting in a fine-grained PoS tagset with 84 label categories that have been used to analyse the reference corpora for Old, Middle, and Early New High German (Donhauser, 2015; Petran et al., 2016; Fisseni, 2017).

2.2 Automatic correction of PoS errors

A lot of work has tried to identify parts-of-speech errors in manual annotations, typically based on the variation in manually assigned tags (see, e.g., Eskin (2000); van Halteren (2000); Květoň and Oliva (2002); Dickinson and Meurers (2003); Loftsson (2009); Helgadóttir et al. (2014); Klie et al. (2023). Less work has focused on errors in automatically predicted annotations, which is a much harder problem, given that those errors are systematic and thus difficult to detect. Rehbein (2014) tried to identify PoS errors in tagger predictions for informal dialogues of youth language, showing that results can be either optimised for precision or for recall, but not both. In follow-up work, Rehbein and Ruppenhofer (2017) present an approach based on an unsupervised generative model guided by human feedback, to create high-quality annotations at affordable cost. Ménard and Mougeot (2019) combine active learning with outlier detection to search for errors in silver standard annotations.

Klie et al. (2023) present a systematic overview of different methods for error detection. They distinguish between *Annotation Error Detection* (AED) and *Annotation Error Correction* (AEC) and criticise that most work only evaluates their method on one dataset and/or one task, making it hard to generalise results and determine the best method for a given setting.

To address this issue, they re-implement and evaluate several methods for error detection and test them on different corpora (some of them, however, created by inserting artificial errors by randomly swapping token labels, which might not always be a realistic approximation of real errors made by humans). One of their findings is that applying tagger ensembles and looking for variance in their predictions to indicate potential errors worked well across the board. We therefore use this approach in our first set of experiments.

While we acknowledge the importance of comparing the performance of new methods on several datasets to achieve generalisable results, in this work we do not develop new techniques for AED and AEC but present an error detection framework tailored to one specific project and task, to assess the benefits of fully automatic AEC in a realistic setting.

2.3 LLMs as PoS taggers

Finally, we overview studies that explore the use of LLMs for PoS tagging, focussing on low-resource scenarios. Machado and Ruiz (2024) evaluate three LLMs for PoS tagging of Brazilian Portuguese in the context of UD, showing that results heavily depend on model size, with promising results for the larger models. Absar (2025) explore the use of LLMs for PoS tagging of code-switched text (Indian-English). Vidal-Gorène et al. (2026) test different LLMs for lemmatisation and PoS tagging of Ancient Greek, Classical Armenian, Old Georgian, and Syriac, finding that foundation models can achieve comparable or even superior performance in low-resource scenarios. Finally, Schöffel et al. (2025) apply seven small open-weight models with parameter sizes of 7B to 14B for PoS tagging of Old Occitan. They find severe limitations especially in the context of high orthographic and syntactic variation.

3 Automatic processing of historical texts

The data we work with are diachronic documents from the 15th to 17th century (Early New High German). Our corpus includes transcriptions of medical texts, more specifically balneological treat-

ises and plague treatises, with more than 150,000 tokens. The corpus also comprises texts from other registers and text types and has been collected for the study of practices of pronoun references, with a focus on the impersonal pronoun "man" (*one*). In this paper, we only focus on the medical subpart of the corpus. The data selection is motivated by the frequent use of instructions in the documents which represents an important domain for studying the pragmatic uses of the pronoun "man".

Data collection The original documents were digitised images from the database "German Technical and Scientific Language until 1700" (Klein et al., 2017), available in pdf format. The database also includes metadata for the documents. The digitised materials are available under the Public Domain Mark 1.0 Universal license.[3] Following the links provided by the database, we downloaded the digitised versions of the prints from the respective libraries.[4]

Transcription The Transkribus platform[5], an AI platform specifically designed for processing historical documents, was used to transcribe the data. The platform offers Optical Character Recognition (OCR) services specialised for historical documents and also provides tools for manual postprocessing. To preserve information on ligatures, diacritical marks, and abbreviations, the Transkribus Print M1 model was applied, with a postprocessing step where system errors have been manually corrected. The system then outputs the corrected files in text format.

Preprocessing All documents have been preprocessed, using the workflow described below.

In a first step, the documents are split into sentences, tokenised and augmented with parts of speech (PoS), using the web service provided by the German Text Archive (DTA) for linguistic analysis of historical texts. Specifically, we use the *Cascaded Analysis Broker (CAB)* by Jurish (2012) which not only predicts PoS tags, but also outputs lemmas and orthografic normalisation on the token level.[6]

[3]See https://creativecommons.org/publicdomain/mark/1.0 (last accessed 2026-08-01).

[4]All documents are freely available as digitised pdf.

[5]Transkribus by READ-COOP, available from https://www.transkribus.org/ and app.transkribus.org (last accessed 2026-08-01.

[6]We used the EXPAND.ALL model of the CAB, available from https://deutschestextarchiv.de/public/cab/ (last accessed 2026-08-01).

While the web service is a valuable resource for research projects working with historical data, the analyses are not perfect (as is arguably true for any automatic analysis). We manually corrected lemmas and normalised word forms, following the guidelines developed in our project. As the manual correction of the PoS layer would take a substantial amount of time, we explore ways to make this process more efficient.

4 Experiments

In a first step, we train and evaluate different PoS taggers to assess the difficulty of the task. Given the lack of high-quality training data with manually assigned (or manually corrected) PoS tags, we use the PoS layer of the RIDGES Herbology corpus, version 9.0 (Odebrecht et al., 2017) as training data. The selection is motivated by the similarity between the texts, both regarding the register and the topic.

RIDGES includes 73 documents, dating from 1482 to 1914. We divided the documents into bins, sorted by centuries, with a varying number of tokens for each time period (see Table 1). While the PoS tags have been automatically predicted, using the TreeTagger (Schmidt, 1994), early versions of the corpus have been subject to a post-correction step, based on DECCA (Dickinson and Meurers, 2003). Thus, the data can be considered to represent a silver standard where the most common errors have been fixed.

4.1 PoS taggers

To identify potentially erroneous PoS tags, we augment the data with additional PoS predictions, using different tagging models trained on subsets of the RIDGES corpus. The decision to use a tagger ensemble rather than training a model on the whole RIDGES data is motivated as follows. First, it is well known that ensembles that combine several individual models produce more accurate predictions than the single best model alone (Zhou, 2012). Second, we hope that using an ensemble will help reduce errors, since most of the annotations in the training data were predicted automatically and contain a certain amount of noise. We also refrain from training models specialised for the various time periods, since our test data do not contain sufficiently large samples for the different time periods and, for each time period, we only have texts by a single author at our disposal, which raises doubts as

Name	Model	15th	16th	17th	18th	19th	20th
RIDGES corpus size:		23,061	87,577	64,443	49,194	63,724	3,831
RIDGES 1	gbert-large	20,000	50,000	35,000	-	-	-
RIDGES 2	gbert-large	15,000	75,000	35,000	10,000	-	-
RIDGES 3	gbert-large	10,000	35,000	55,000	35,000	10,000	-
RIDGES 4	gbert-large	-	15,000	35,000	50,000	35,000	-
RIDGES 5	TIGER	20,000	35,000	35,000	35,000	35,000	-
TIGER	gbert-large	*trained on 888,238 tokens from TiGer for 1 epoch*					

Table 1: Data sources used for PoS tagger training. The first row shows the no. of available tokens in RIDGES for each century. Model: the model used to initialise the tagger.

to the representativeness of the time-period-specific subsamples. As input, we use the normalised word forms provided by DTA CAB and validated by our student annotators.

Our first PoS layer includes the tags predicted by DTA CAB. We add additional layers of automatically predicted PoS tags, using a BERT token classification model (Devlin et al., 2019) trained on different subsets of the RIDGES corpus and on contemporary German. The subsets differ regarding the proportion of tokens taken from the different sources and time periods. We hypothesize that this will result in taggers specialised for different time periods. To increase the diversity of the ensemble tags, we add a model that has been initialised with the weights of a tagger trained on contemporary German (TIGER) and is further finetuned on text from the RIDGES corpus (RIDGES-5, see below).

TIGER The TIGER model has been trained on contemporary standard German newspaper articles from the TiGer treebank (Brants et al., 2004). As the corpus is quite large and we do not want to overfit the tagger on this data, we train the model for one epoch only.

RIDGES 1-5 These models have been trained on randomly selected tokens from different centuries (see Table 1). We sample new sentences from each of the different bins until the token count for the sample exceeds the specified number of tokens (e.g., for RIDGES 1: 20,00 tokens from the 15th century, 50,000 tokens from the 16th century and 35,000 tokens from the 17th century). We shifted the token balance so that the majority of tokens for each model comes from a different time period.

The RIDGES-5 model has been initialised with the TIGER model described above. We continued finetuning the model using RIDGES data that is relatively balanced across the various time periods.

DocID	Time	TP	FP	FN	Missed
1	1494	71.0	6.4	7.8	12.5
2	1550	46.8	28.9	8.8	14.3
3	1608	41.3	35.0	13.5	9.5
4	1690	64.5	19.1	8.9	7.0
Avg.		46.9	25.9	11.9	12.9

Table 2: Percentage of instances where (i) the ensemble tag was correct (TP); (ii) the DTA tag was correct (FP); (iii) both predicted tags were incorrect but different (FN); (iv) both tags agreed but were incorrect (Missed).

Tagging model and parameters Our tagger is a BERT token classifier, initialised with the `deepset/gbert-large` model[7] (Chan et al., 2020) and trained for 6 epochs, using a batch size of 16 and a learning rate of 6.35e-05. The parameters have been determined through hyperparameter tuning, based on Optuna.[8] For continued finetuning of the TIGER model (RIDGES-5), we only train for four epochs.

We use the ensemble of pos tagging models to preprocess our target data. We keep the tags assigned by the DTA CAB service as a reference and, based on the newly predicted tags, we determine a new PoS prediction using a majority vote over RIDGES 1-5. We hypothesize that DTA-assigned PoS tags that deviate from the new tag based on the majority vote might be incorrect.

4.2 Manual error correction

We create a test set for our target data where the automatic predictions have been revised by two expert annotators (both trained historical linguists). For PoS correction, they used a graphical user in-

[7] https://huggingface.co/deepset/gbert-large (last accessed 2026-08-01)

[8] https://github.com/optuna/optuna (last accessed 2026-08-01)

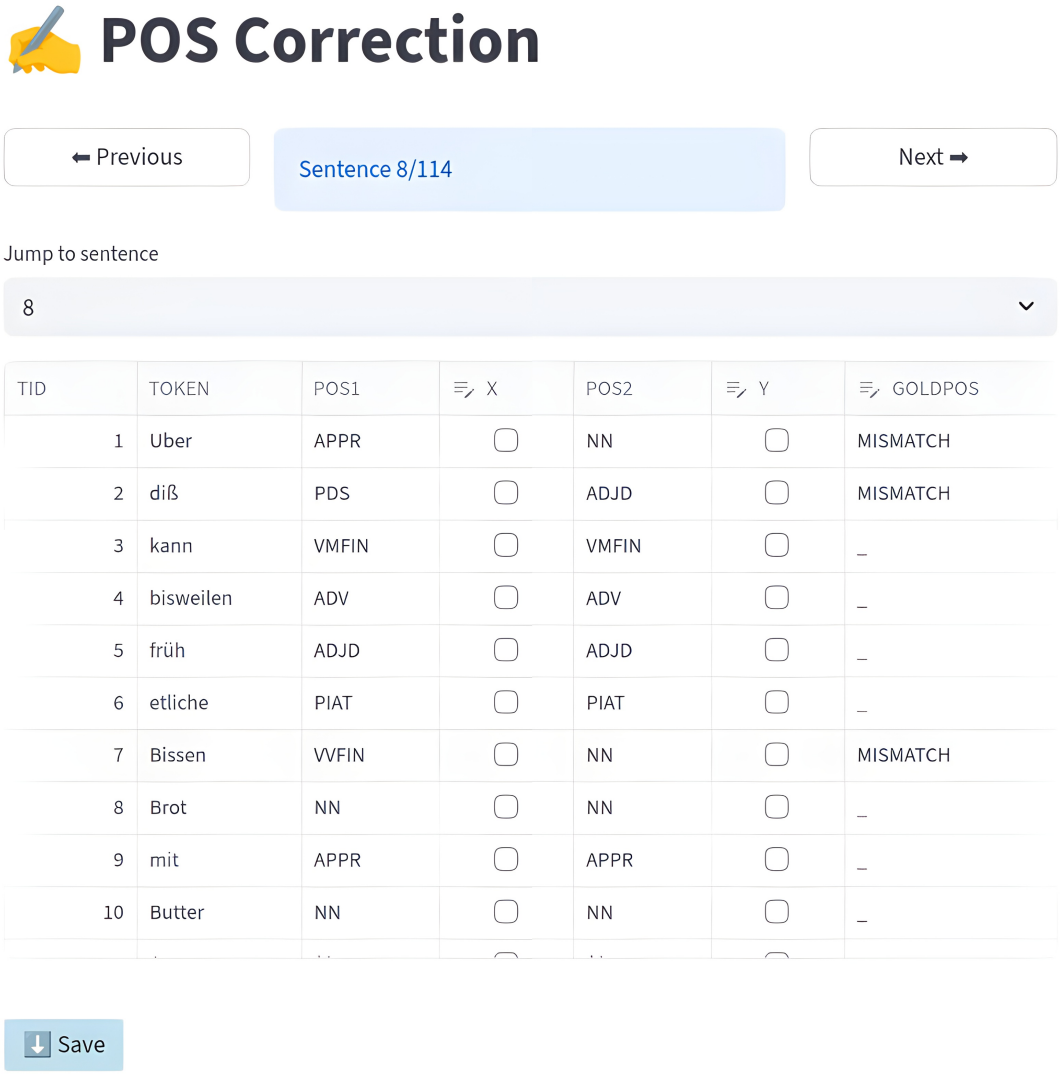

Figure 1: User interface for PoS correction. Potential errors are marked with a MISMATCH tag which enables annotators to process the texts in an efficient manner. The annotators can either select one of two predicted PoS by clicking on the corresponding checkbox or, if both are incorrect, they can enter the correct PoS label in the GOLDPOS column.

terface where mismatches between the DTA tag and the majority tag have been marked (see Figure 1). The correct tag can be selected by clicking at the check box next to the tag or, if none of the two tags are correct, has to be inserted in the right-most column. In addition to the marked tags, the annotators were instructed to also correct errors where the automatic tags agreed but were both wrong.

The resulting **gold standard** includes four documents from different centuries (1494, 1550, 1608, 1690) with 20,799 tokens. Each of the files has been corrected by one trained annotator.

Around 20% of the predicted tokens were highlighted for review. However, the numbers vary considerably across texts and time periods. In the earliest text, 32% of the instances have been marked while in the newest document, only 13% of the tokens had to be checked (also see Table 2).

Overall, for roughly 47% of the tokens that have been corrected by our experts, the ensemble tags provided the correct label. In 26% of all cases, the DTA-assigned tag was correct and in the remaining 25%, both tags were incorrect. Critical cases are the 13% of missed instances where both taggers predicted the same, but incorrect, tag, meaning that those were not flagged by our framework. Note that this number might be an optimistic estimate, given that we had only one annotator per document and that there might be unmarked errors that have not been found during the correction phase.

The numbers also tell us that a manual revision of around 20% of the data succeeds in identifying and correcting around 84% of errors in the data (ignoring the missed cases that were not flagged and might be missed when correcting the data).[9] It also means that, while simply replacing the DTA-based PoS tags with the ones from the tagger ensemble would also introduce new errors, overall we could decrease the PoS error rate in the data from 16% to 11.3% without *any* manual work. The PoS tags predicted by the DTA tagger had an error rate of 16.0% (3,336 out of 20,799 tokens) while the error rate for the tagger ensemble was substantially lower with 11.3% (2,361 out of 20,799 tokens). Overall, this seems like an efficient approach to correcting PoS errors in historical text.

4.3 POS error analysis

We start with an error analysis to identify common error types that we can then address. We take the mismatches between the DTA and ensemble predictions as our starting point to identify frequent error patterns (Table 3). Please note that our analysis should not be seen as a comparison between the DTA tagger and the RIDGES-trained ensemble as the two models predicted PoS tags based on different inputs. The DTA made predictions based on the original text while the ensemble was provided with the corrected normalised tokens.

FM-OTHER We identify five frequent error patterns, based on the predictions of the DTA tagger and the tagger ensemble trained on RIDGES. The first one concerns foreign language material (mostly Latin) in our data. Here we see that the DTA tagger does a good job and correctly labels 706 of the 780 instances where we observe a label mismatch. Thus a strong baseline for this error type would be to simply assign the DTA tag.

Most errors made by the RIDGES-trained model concern proper names (NE). This is not surprising, giving that NE is often selected by the TreeTagger as a residual class when other parts-of-speech seem improbable. Here a clean-up of the RIDGES corpus that focusses on NE errors might be a promising way to improve tags.

NE-OTHER For instances tagged as NE, the picture is reversed. Here the DTA tagger mostly predicted an incorrect label while the RIDGES-trained model assigned the right label in over 70% of cases (291 out of 410). Here our baseline is to always keep the ensemble label.

VVFIN-OTHER The next frequent error type concerns instances that are predicted as finite verb (VVFIN) by the DTA tagger. Here we mostly see past participles (VVPP) and infinite verbs (VVINF) but also some nouns (NN) and adjectives (ADJD) that have been misclassified. Our baseline here is to always keep the ensemble tag, which would result in correct predictions for 85% of the mismatches (232 out of 272).

PTKZU-OTHER This error type mostly includes prepositions that have been tagged as PTKZU (particle "zu" before infinite verb) by the DTA tagger. For this error pattern, keeping the tags assigned by the RIDGES-trained model would be correct in 117 out of the 127 cases (92%).

ADV-OTHER Finally, we look at errors where the DTA tool predicted an adverb (ADV) while the ensemble tagger analysed the token as an adverbial or predicative adjective (ADJD). In the majority of all cases, the DTA-assigned tag would be correct, resulting in a baseline of roughly 77% (53 out of 69).

In conclusion, when focussing on the most frequent error types in the data and using the majority baselines described above, we can cover nearly 40% of the mismatches identified by our error correction approach, and more than 35% of the real error in the data (including the ones that have not been flagged).

This gives us an **upper bound** of what we could achieve if we had a method that would correct PoS tags without errors. It also means that any error correction method that we apply at this stage needs to do better than correcting 35% of the errors that we can fix in a simple rule-based approach, based on our analysis of frequent error patterns in the data.

Even though the margin of what we can expect to gain is quite small, it might be still worthwhile to explore how well we can do, given that we could also apply our methods to the larger RIDGES corpus to correct PoS errors in the data and obtain more reliable training data for our corpus.

5 Using LLMs for AEC

We now want to find out whether we can improve results for AEC over the pattern-based approach

[9] Optimistically assuming that our annotators found all errors in the gold standard dataset.

Error type	DTA	RIDGES Ensemble	Freq.	GOLD	Total	Percentage
FM-OTHER	FM	NE	442	FM		
		NE	9	NE		
		NN	136	FM		
		NN	13	NN		
		ADJA	100	FM		
		ADJA	18	ADJA		
		APPR	18	FM		
		APPR	15	APPR		
		ADV	10	FM		
		ADV	8	ADV		
		ART	10	ART		
		ART	1	PDS	*780 errors*	18.7 %
NE-OTHER	NE	NN	291	NN		
		NN	89	FM		
		NN	21	NE		
		NN	9	ADJA, TRUNC, VVFIN, CARD, ADV, ADJD	*410 errors*	9.9%
VVFIN-OTHER	VVFIN	VVPP	73	VVPP		
		VVINF	65	VVINF		
		NN	64	NN		
		ADJD	23	ADJD		
		VVPP	17	VVFIN		
		VVINF	13	VVFIN		
		VMFIN	7	VMFIN		
		ADJD	10	VVFIN, VVIMP, VVPP, ADJD	*272 errors*	6.5%
PTKZU-OTHER	PTKZU	APPR	98	APPR		
		APPR	3	PTKA		
		APPR	1	PTKVZ		
		APPR	1	PTKZU		
		APPR	1	PTKVZ		
		PTKA	14	PTKA		
		PTKA	3	APPR		
		PTKA	1	PTKZU		
		PTKVZ	5	PTKVZ	*127 errors*	3.0%
ADV-OTHER	ADV	ADJD	53	ADV		
		ADJD	14	ADJD		
		ADJD	2	APPR, TRUNC	*69 errors*	1.7%

Table 3: Most frequent error patterns in our data (FM: foreign language material; NE: proper name; VVFIN: finite verb (full); PTKZU: particle "zu" before infinite verb; ADV: adverb; NN: noun; ADJA: attributive adjective; APPR: preposition; ART: determiner; VVPP: past participle; ADJD: predicative adjective; VVINF: infinite verb (full); PTKA: answer particle; PDS: substitutive definite pronoun; CARD: cardinal number; TRUNC: truncated component; VVIMP: imperative verb (full); PTKVZ: separable verb prefix).

ID		DTA	RIDGES	$RIDGES_{RULE}$	GEMMA all	GEMMA zero	GEMMA few	OSS (all) all	OSS zero	OSS few
1	ALL TAGS	83.8	88.9	91.9	88.7	92.1	92.2	90.8	92.8	**92.9**
2	ADV	-	-	-	-	89.1	89.1	-	89.2	89.2
3	FM	-	-	-	-	92.0	-	-	92.3	-
4	NE_NN	-	-	-	-	88.9	89.1	-	89.2	89.3
5	PTKZU	-	-	-	-	88.8	88.9	-	88.9	88.9
6	VVFIN	-	-	-	-	88.2	88.8	-	89.0	89.0

Table 4: Accuracies for the different LLM-based AEC approaches on the test set (for FM, we do not provide few-shot examples).

that addresses frequent error types in our data (see §4.3 above). Specifically, we investigate whether using generative AI can help to automatically correct at least some of the errors in our data, in order to reduce the number of instances that need to be revised by human experts.

5.1 Models

We experiment with different open-weight models in zero-shot and few-shot settings.[10] All models can fit on a single GPU.[11]

Mistral-Small-24B-Instruct-2501 is a fast and efficient general-purpose model from MistralAI with 24 billion parameters.

Gemma-3-27b-it from Google is trained for text generation and image understanding tasks and, with 27 billion parameters, is only slightly larger than the small Mistral model.

Gpt-oss-120B is a general purpose model from OpenAI and, with 120 billion parameters, significantly larger.

5.2 Baseline

As a baseline, we include an experiment where we prompt an LLM to predict PoS tags for each token in the documents (as compared to identifying errors in the output of our tagger ensemble). For that, we use the largest model, `Gpt-oss-120B`, in a zero-shot setup. The prompt is shown in the Appendix, Figure 2.

5.3 Correcting FM errors

For AEC of FM related errors (i.e., instances where the DTA tagger predicted FM while the RIDGES-trained model decided on another label), we reformulate the PoS tagging task as language identification. Specifically, we prompt the model to determine whether a given token in a sequence of tokens is German or not. We prompt the model to answer with either "DE" (for German) or "FM" (for foreign language material). The prompt is included in the Appendix, Figure 4.

[10] Details for the few-shot settings can be found in §A.1.

[11] For example, an Nvidia H100 or similar.

Text	Date	BASE	DTA	RID	# S.	none
o.A.	1494	77.6	65.7	90.0	48	4
Willich	1550	68.5	80.4	85.4	114	3
Brendel	1608	67.7	86.0	87.4	289	9
Heer	1690	70.9	88.8	95.1	114	5
Total		**69.2**	**83.8**	**88.9**	**565**	**21**

Table 5: Results for individual texts for the LLM baseline and for the AEC approaches (BASE: Baseline model, DTA: DTA CAB, RID: RIDGES ensemble w/o rules, # S.: no. of sentences, none: no. of sentences where the number of tags predicted by the baseline did not match the number of tokens).

5.4 Correcting other error types

For the correction of the remaining error types (ADV-OTHER, NE-OTHER, PTKZU-OTHER, VVFIN-OTHER), we use the same prompt template in the zero-shot setting. Here we provide the model with a token and its respective sentence as context, where the token of interest is highlighted. This token is then again presented to the model, together with two alternative PoS tags to chose from (the tags predicted by the DTA and RIDGES-trained taggers). We instruct the model to chose the correct tag or, in the case that both are incorrect, to present the correct tag from the STTS tagset.

5.5 LLM-based correction of generic errors

In our final setting, we do not target specific error types but prompt the LLMs to resolve *all* POS mismatches between the tags predicted by the DTA tagger and the tagger ensemble. Here we want to see whether we can increase recall for AEC without loosing precision (see Appendix, Figure 6). We refer to this setting as POS-ALL.

6 Results

6.1 Baseline

Table 5 shows results for the baseline model for individual texts. Please note that, due to the limited scope of the gold standard, we only have texts by a single author available for the various time periods, and it is unclear whether differences in accuracy are attributable to the publication date, the author's style, or specific characteristics of the text.

While most of the tags predicted by the baseline model are valid STTS tags, for some sentences the number of predicted tags did not match the number of tokens in the input. We exclude those sentences in the evaluation and report accuracy for the sentences where token and PoS numbers are the same. Not surprisingly, the model struggles to predict the correct tags for the historical texts, with an average score that is around 15-20% lower than the ones for the smaller models, DTA and RIDGES.

6.2 Results for AEC

Below we present the results for the AEC approach. The first observation we make is that MISTRAL-SMALL often fails to follow the instructions and, even though we prompted the model to only output the correct PoS tag, tended to provide longer answers and justification for its choice. Some of the tags could, in theory, be extracted by using regular expressions. However, we did not spend much time on parsing the answers but excluded Mistral from the evaluation.

GEMMA-3-27B-IT and GPT-OSS-120B, on the other hand, always follow the instruction and provid well-formed answers. Table 4 shows results for the different error types. The first row reports results on the whole goldstandard, below we list results for individual error types. The effects we observe are rather small, due to the fact that we measure the impact of correcting individual sets of labels against the total number of tokens in the data. Considering only the set of tokens flagged as potential errors would show larger effect sizes for AEC, however, from our perspective the accuracy on the overall data is more informative for judging whether a particular method is worth the effort.

We can see that the largest improvements over the DTA baseline are obtained by the tagger ensemble, with an increase of around 5%. We can further improve accuracy by another 3 percentage points when applying our rule-based correction to the RIDGES-trained tagger output. Applying GEMMA-3-27B-IT and GPT-OSS-120B to all PoS tags marked as errors fails to improve results. Here the accuracies for GEMMA-3-27B-IT are below the ones for the RIDGES-trained tags and, for GPT-OSS-120B, slightly higher than the ones for RIDGES but lower than the accuracy for RIDGES$_{RULE}$.

A more promising approach is to apply the LLMs in a setting tailored to correct specific error types, with prompts that give concrete instructions for each PoS error. This results in results for Gemma that are on par with the rule-based approach (RIDGES$_{RULE}$) and slightly better for GPT-OSS-120B with another increase of around 1%. Surprisingly, few-shot prompting does not outperform zero-shot prompting.

To find out where we achieved the highest improvements using LLM-based AEC, we now look at the results for the individual error types (Table 4, rows 2-6). For both models, we see the largest increase for the FM errors. We assume that reformulating the PoS tagging task as a language identification task might have helped. We should thus try to think of other ways to rephrase AEC for specific error types, to make the best use of generative large language models.

In summary, our approach provides an efficient workflow for research projects where there is not much time to devote to manual annotation. Although the error categories that automatic correction should focus on are project-specific and must be identified within the context of each project, we have shown that this can be accomplished automatically and with minimal human supervision.

7 Conclusion

In this paper, we have presented a fully automated pipeline for correcting PoS errors in historical texts. We have shown that significant improvements can be achieved by using a tagger ensemble trained on noisy in-domain data and by comparing the annotations with the predictions of a tagger trained on a different corpus. Identifying the most common error types and correcting them using a rule-based approach led to improvements of around 7 percentage points (accuracy). We were then able to demonstrate that the use of LLMs with open weights can further boost performance, resulting in an overall improvement of around 9 percentage points compared with the DTA baseline.

Limitations

We acknowledge the following limitations of our work. First, the size of our gold standard is not large with around 20,800 tokens. Although we have included texts from various eras to cover a wide range of diachronic features, it is clear that we cannot claim to be representative. Second, the data was corrected by one person only, meaning that some errors in the data might have been overlooked, even though our annotators are both trained experts in historical linguistics. Regarding LLMs, we only included three different open-weight models in our study. We thus do not make any claims regarding the best current model for PoS tagging of historical German. We also did not test larger closed-source LLMs as our focus is on improving PoS quality with limited resources, not on comparing the newest models for this task. As a final limitation, we would like to mention the training splits of the RIDGES data into time slices. Splitting the data up by century was an arbitrary choice that might not adequately reflect the historical facts of language change. Our main motivation here was to obtain a tagger ensemble that shows some variety regarding its prediction. It might be interesting to compare this to a more linguistically motivated approach to splitting up the data. We leave this for future work.

Acknowledgments

The work presented in this paper is partly funded by the German Research Foundation (DFG) under the grant "Referential Practice in Transition: The Pronoun 'man' in the Diachrony of German" within the DFG Research Group "FOR 5317 Practices of Personal Reference: Personal, Indefinite, and Demonstrative Pronouns in Use" (Project No. 457855466). We would also like to thank our annotators, Liv Büchler, Lena Christoffer, Nele Zohren and Maya Jeken.

References

Shayaan Absar. 2025. Fine-tuning cross-lingual LLMs for POS tagging in code-switched contexts. In *Proceedings of the Third Workshop on Resources and Representations for Under-Resourced Languages and Domains (RESOURCEFUL-2025)*, pages 7–12, Tallinn, Estonia. University of Tartu Library, Estonia.

Marcel Bollmann. 2013. POS tagging for historical texts with sparse training data. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 11–18, Sofia, Bulgaria. Association for Computational Linguistics.

Marcel Bollmann, Natalia Korchagina, and Anders Søgaard. 2019. Few-shot and zero-shot learning for historical text normalization. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 104–114, Hong Kong, China. Association for Computational Linguistics.

Marcel Bollmann, Florian Petran, and Stefanie Dipper. 2011. Rule-based normalization of historical texts. In *Proceedings of the Workshop on Language Technologies for Digital Humanities and Cultural Heritage*, pages 34–42, Hissar, Bulgaria. Association for Computational Linguistics.

Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. 2004. TIGER: Linguistic Interpretation of a German Corpus. *Research on Language and Computation*, 2:597–620.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Markus Dickinson and W. Detmar Meurers. 2003. Detecting errors in part-of-speech annotation. In *10th Conference of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary. Association for Computational Linguistics.

Stefanie Dipper. 2011. Morphological and Part-of-Speech Tagging of Historical Language Data: A Comparison. *Journal for Language Technology and Computational Linguistics*, 26(2):25–37.

Stefanie Dipper, Karin Donhauser, Thomas Klein, Sonja Linde, Stefan Müller, and Klaus-Peter Wegera. 2013. HiTS: ein Tagset für historische Sprachstufen des Deutschen. *Journal for Language Technology and Computational Linguistics, Special Issue*, 28(1):85–137.

Karin Donhauser. 2015. Das Referenzkorpus Altdeutsch: Das Konzept, die Realisierung und die neuen Möglichkeiten. In Jost Gippert and Ralf Gehrke, editors, *Historical Corpora. Challenges and Perspectives*, CLIP 5, pages 35–49. Narr Francke Attempto Verlag GmbH + Co. KG, Tübingen.

DTA. 2026. Deutsches Textarchiv. Grundlage für ein Referenzkorpus der neuhochdeutschen Sprache.

Eleazar Eskin. 2000. Detecting errors within a corpus using anomaly detection. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*.

Bernhard Fisseni. 2017. Das Bonner Frühneuhochdeutschkorpus (FnhdC), 2017. Dokumentation zur aktuellen Fassung.

Sigrún Helgadóttir, Hrafn Loftsson, and Eiríkur Rögnvaldsson. 2014. Correcting errors in a new gold standard for tagging Icelandic text. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2944–2948, Reykjavik, Iceland. European Language Resources Association (ELRA).

Birgit Herbers, Sylwia Kösser, Ilka Lemke, Ulrich Wenner, Juliane Berger, Sarah Kwekkeboom, and Frauke Thielert. 2021. Dokumentation zum Referenzkorpus Frühneuhochdeutsch und Referenzkorpus Deutsche Inschriften. *Bochumer Linguistische Arbeitsberichte*, 24.

Bryan Jurish. 2003. A hybrid approach to part-of-speech tagging. Technical report, Berlin-Brandenburgische Akademie der Wissenschaften.

Bryan Jurish. 2012. *Finite-state canonicalization techniques for historical German. Endliche Techniken zur Kanonikalisierung historischen deutschen Textes*. Ph.D. thesis, Universität Potsdam.

Wolf P. Klein, Michael Breyl, Solveig Ahlers, Isabel Kromer, and Saskia Schmid. 2017. Projektbeschreibung.

Jan-Christoph Klie, Bonnie Webber, and Iryna Gurevych. 2023. Annotation error detection: Analyzing the past and present for a more coherent future. *Computational Linguistics*, 49(1):157–198.

Pavel Květoň and Karel Oliva. 2002. (Semi-)Automatic Detection of Errors in PoS-Tagged Corpora. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Hrafn Loftsson. 2009. Correcting a POS-tagged corpus using three complementary methods. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 523–531, Athens, Greece. Association for Computational Linguistics.

Mateus Machado and Evandro Ruiz. 2024. Evaluating large language models for the tasks of PoS tagging within the Universal Dependency framework. In *Proceedings of the 16th International Conference on Computational Processing of Portuguese - Vol. 1*, pages 454–460, Santiago de Compostela, Galicia/Spain. Association for Computational Lingustics.

Pierre André Ménard and Antoine Mougeot. 2019. Turning silver into gold: error-focused corpus reannotation with active learning. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 758–767, Varna, Bulgaria. INCOMA Ltd.

Carolin Odebrecht, Malte Belz, Amir Zeldes, Anke Lüdeling, and Thomas Krause. 2017. Ridges herbology: designing a diachronic multi-layer corpus. *Language Resources & Evaluation*, 3(51):695–725.

Florian. Petran, Marcel Bollmann, Stefanie Dipper, and Klein Thomas. 2016. ReM: A reference corpus of Middle High German – corpus compilation, annotation, and access. *Journal for Language Technology and Computational Linguistics*, 31(2):1–15.

Ines Rehbein. 2014. POS error detection in automatically annotated corpora. In *Proceedings of LAW VIII - The 8th Linguistic Annotation Workshop*, pages 20–28, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Ines Rehbein and Josef Ruppenhofer. 2017. Detecting annotation noise in automatically labelled data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1160–1170, Vancouver, Canada. Association for Computational Linguistics.

Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. 1999. Guidelines für das Tagging deutscher Textcorpora. Technical report, University of Stuttgart and University of Tübingen.

Helmut Schmidt. 1994. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*.

Matthias Schöffel, Marinus Wiedner, Esteban Garces Arias, Paula Ruppert, Christian Heumann, and Matthias Aßenmacher. 2025. Modern models, medieval texts: A POS tagging study of old Occitan. In *Proceedings of the 5th International Conference on Natural Language Processing for Digital Humanities*, pages 334–349, Albuquerque, USA. Association for Computational Linguistics.

Sarah Schulz and Jonas Kuhn. 2016. Learning from Within? Comparing PoS Tagging Approaches for Historical Text. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4316–4322, Portorož, Slovenia. European Language Resources Association (ELRA).

Hans van Halteren. 2000. The detection of inconsistency in manually tagged text. In *Proceedings of the COLING-2000 Workshop on Linguistically Interpreted Corpora*, pages 48–55, Centre Universitaire, Luxembourg. International Committee on Computational Linguistics.

Chahan Vidal-Gorène, Bastien Kindt, and Florian Cafiero. 2026. Under-resourced studies of under-resourced languages: lemmatization and POS-tagging with LLM annotators for historical Armenian, Georgian, Greek and Syriac. In *Proceedings of the Second Workshop on Language Models for Low-Resource Languages (LoResLM 2026)*, pages 324–334, Rabat, Morocco. Association for Computational Linguistics.

Zhi-Hua Zhou. 2012. *Ensemble Methods: Foundations and Algorithms*. CRC Press.

A Appendix

A.1 Few-shot settings

We compare the zero-shot setting to a few-shot scenario where we provide the model with examples targeting the different error types. The examples are appended to the user prompt (see §A.2 below). All examples have been extracted from the RIDGES corpus.

ADV-OTHER For the error type ADV-OTHER, we use a five-shot setting with the examples below.

Beispiel 1: als have ich / <POS>kürzlich<POS> hiervon meine Meinung niemands vorgegriffen /
Wortart zuweisen: ('kürzlich', 'ADV', 'ADJD')
Antwort: ADV

Beispiel 2: oder einem Schattenhut <POS>gleich<POS> /
Wortart zuweisen: ('gleich', 'ADV', 'ADJD')
Antwort: ADJD

Beispiel 3: und Saft von der Zeitlosen Wurzel jegliches <POS>gleich<POS> viel
Wortart zuweisen: ('vermischt', 'ADJD', 'ADV')
Antwort: ADJD

Beispiel 4: und <POS>erstlich<POS> folgt das Kräutlein Abbiss /
Wortart zuweisen: ('erstlich', 'ADV', 'ADJD')
Antwort: ADV

Beispiel 5: danach mische <POS>ferner<POS> guten Zucker dazu /
Wortart zuweisen: ('ferner', 'ADV', 'ADJD')
Antwort: ADV

FM-OTHER For the language identification task, we do not provide few-shot examples.

NE-OTHER We use a six-shot setting for the NE-OTHER error type, using the examples below.

Beispiel 1: Der eine appolexia heißt und das andere <POS>Eppilencia<POS> .
Wortart zuweisen: ('Eppilencia', 'NE', 'NN')
Antwort: NE

Beispiel 2: Auqa stillatitia ex <POS>herba<POS> .
Wortart zuweisen: ('herba', 'NE', 'NN')
Antwort: FM

Beispiel 3: Der ägyptische Schlottendorn / in lateinischer Sprache <POS>Acacia<POS> genannt /
Wortart zuweisen: ('Acacia', 'NN', 'NE')
Antwort: NE

Beispiel 4: Dem wohlgeborenen Herrn / <POS>Doctor<POS> Paulo Khevenhüller von Aichelberg / Freiherr in LandtsCron /
Wortart zuweisen: ('Doctor', 'NE', 'NN')
Antwort: NN

Beispiel 5: Dem wohlgeborenen Herrn / Doctor Paulo Khevenhüller von Aichelberg / Freiherr in <POS>LandtsCron<POS> /
Wortart zuweisen: ('LandtsCron', 'NN', 'NE')
Antwort: NE

Beispiel 6: so wenig können auch ehrliche und erfahrene <POS>Medici<POS> solchen ihren Frevel approbieren
Wortart zuweisen: ('Medici', 'NE', 'NN')
Antwort: NN

PTKZU-OTHER For the PTKZU-OTHER error type, we provide three examples in the few-shot setting.

Beispiel 1: ist eine goldene Arznei <POS>zu<POS> unzähligen Krankheiten /
Wortart zuweisen: ('zu', 'PKTZU', 'APPR')
Antwort: PPER

Beispiel 2: / unten aber auf der Seite gegen der Erden <POS>zu<POS> ganz weiß
Wortart zuweisen: ('zu', 'PTKZU', 'APPR')
Antwort: APZR

Beispiel 3: Also hat es auch seine Bedeutung / Nutzen und Brauch <POS>zu<POS> der Seelen Arznei .
Wortart zuweisen: ('zu', 'PTKZU', 'APPR')
Antwort: APPR

VVFIN-OTHER For this error type, we use three examples.
Beispiel 1: tu <POS>es<POS> auf die Feigwarzen.
Wortart zuweisen: ('es', 'VVFIN', 'PPER')

Antwort: PPER

Beispiel 1: Ist gut <POS>gebraucht<POS> dem entzündeten Podagra
Wortart zuweisen: ('gebraucht', 'VVFIN', 'VVPP')
Antwort: VVPP

Beispiel 1: Und wird <POS>vermischt<POS> den Arzneien
Wortart zuweisen: ('vermischt', 'VVPP', 'VVFIN')
Antwort: VVPP

A.2 Prompt templates

Figure 2 shows the prompt used for the LLM baseline where we task the model to predict PoS tags for each input token in a zero-shot setting.

Figures 4-7 illustrate the prompts used in the zero-shot settings. The system prompt provides the model with a role description (expert for MHG and EMHG) and a high-level task description. The user prompt then adds detailed instructions for the task. In the few-shot setting, we append the few-shot examples (see §A.1) to the user prompt.

Prompt: `BASELINE (zero-shot)`

Rolle und Ziel:
Du bist Experte für Neuhochdeutsche und Frühneuhochdeutsche Sprache. Deine Aufgabe ist es, Wortarten in historischen Texten zu taggen.
Kontext der Aufgabe:
Du erhältst einen Textabschnitt aus einem neuhochdeutschen Text. Bestimme für jedes Wort die korrekte Wortart nach dem STTS-Tagset.
Anweisungen:

1. Lies den bereitgestellten **Textabschnitt** sorgfältig durch. Der Text ist als Liste in einem JSON-Objekt formatiert.
2. Analysiere alle Worte in ihrem grammatikalischen Kontext und weise jedem Wort sein korrektes PoS-Tag zu. Die PoS-Tags sollen als Liste "predictions" im JSON-Objekt eingefügt werden.
3. Die Antwort soll nur das JSON-Objekt enthalten. Achte darauf, dass die Wortliste die gleiche Anzahl an Tokens hat wie die PoS-Liste.

Textabschnitt: `<input tokens>`

Figure 2: Prompt template for the LLM BASELINE. For the English translation, see Figure 3 below.

Prompt: `BASELINE (zero-shot)`

Role and Objective:
You are an expert for Modern High German and Early Modern High German. Your task is to predict parts of speech in historical text.
Task Context:
You will be given a text passage from a Modern High German text. For each word, identify its correct part of speech according to the STTS tag set.
Instructions:

1. Carefully read the provided **text passage**. The text is formatted as a list within a JSON object.
2. Analyse all words in their grammatical context and assign the correct PoS tag to each word. The PoS tags should be inserted as a list with key “predictions” in the JSON object.
3. The answer should contain only the JSON object. Make sure that the word list has the same number of tokens as the PoS list.

Text passage: `<input tokens>`

Figure 3: Prompt template for the LLM BASELINE (English translation)

Prompt: `FM-OTHER (zero-shot)`

Rolle und Ziel:
Du bist Experte für Neuhochdeutsche und Frühneuhochdeutsche Sprache. Deine Aufgabe ist es, fremdsprachliches Material zu identifizieren.
Aufgabe:
Du erhältst den folgenden Input:
Text: eine kurze Textspanne aus einem frühneuhochdeutschen Text.
Wort: ein Wort in dieser Textspanne.
Entscheide, ob das Wort im Kontext deutsch oder fremdsprachlich (meist Latein) ist. Antworte mit 'FM' für fremdsprachlich oder mit 'DE' für deutsch.

Figure 4: Prompt template for the FM-OTHER error type (zero-shot setting). For the English translation, see Figure 5 below.

Prompt: `FM-OTHER (zero-shot)`

Role and Objective:
You are an expert for Modern High German and Early Modern High German. Your task is to identify foreign-language material.
Task:
You will receive the following input:
Text: a short text excerpt from an Early Modern High German text.
Word: a word within this text segment.
Determine whether the word is German or a foreign language (usually Latin) in context. Respond with 'FM' for foreign language or 'DE' for German.

Figure 5: English translation of the prompt template for the FM-OTHER error type (zero-shot setting).

Prompt: `POS-ALL (zero-shot)`

Rolle und Ziel:
Du bist Experte für Neuhochdeutsche und Frühneuhochdeutsche Sprache. Deine Aufgabe ist es, Fehler in der Zuweisung von PoS-Tags zu korrigieren.
Kontext der Aufgabe:
Du erhältst einen Textabschnitt aus einem neuhochdeutschen Text. Im Textabschnitt ist ein Wort mit '<POS>'-Tags markiert. In der nächsten Zeile wird das Wort wiederholt, zusammen mit zwei möglichen PoS-Tags. Entscheide, welches der beiden vorgeschlagenen Tags das richtige ist. Wenn beide Tags falsch sind, dann nenne das korrekte Tag aus dem STTS-Tagset.
Anweisungen:

1. Lies den bereitgestellten **Textabschnitt** sorgfältig durch.
2. Identifiziere das markierte **Wort**.
3. Analysiere das Wort in seinem grammatikalischen Kontext.
4. Lies die beiden vorgeschlagenen PoS-Tags und weise das korrekte Tag zu. Die Antwort soll nur das korrekte PoS-Tag enthalten.

Figure 6: Prompt template for correcting all potential errors, marked by tag mismatches between the DTA tagger und the ensemble model (POS-ALL, zero-shot setting). For the English translation, see Figure 7 below.

Prompt: `POS-ALL (zero-shot)`

Role and Objective:
You are an expert for Modern High German and Early Modern High German. Your task is to correct errors in the assignment of PoS tags.
Task Context:
You are given a text excerpt from a Modern High German text. In the excerpt, a word is marked with '<POS>' tags. In the next line, the word is repeated, along with two possible PoS tags. Decide which of the two suggested tags is correct. If both tags are incorrect, specify the correct tag from the STTS tag set.
Instructions:

1. Read the provided **text segment** carefully.
2. Identify the highlighted **word**.
3. Analyse the word in its grammatical context.
4. Read the two suggested PoS tags and assign the correct tag. Your answer should contain only the correct PoS tag.

Figure 7: Prompt template for correcting all potential errors, marked by tag mismatches between the DTA tagger und the ensemble model (POS-ALL, zero-shot setting).