POS Tagging with Generative LLMs for Historical Germanic Low-Resource Languages: An Evaluation Against Fine-Tuned BERT

Irene Miani[1], Gregory Darwin[2], Sara Stymne[3]
Department of Linguistics and Philology[1, 3], Department of English[2]
Uppsala University
(irene.miani[1], sara.stymne[3])@lingfil.uu.se, gregory.darwin@engelska.uu.se[2]

Abstract

Part-of-Speech (POS) tagging is a fundamental task in Natural Language Processing, yet its performance on historical low-resource languages is still underexplored, particularly in the context of large generative models. While recent studies have demonstrated strong results for Large Language Models (LLMs) on modern languages and contemporary low-resource settings, their effectiveness for historical varieties remains unclear. Moreover, genre-specific structural variation, which may substantially affect tagging performance, has received limited attention. This study evaluates the zero- and few-shot POS tagging performance of two generative models on four historical Germanic low-resource languages across two literary genres. Their performance is benchmarked against fine-tuned BERT models. To contextualize the performance on historical data, the models are also evaluated on two modern languages. The results show that fine-tuned encoder models consistently outperform generative models across all settings. The performance of the LLMs on historical languages is substantially lower compared to that on modern languages, suggesting limited representation of these varieties in pretraining data. Furthermore, error analysis reveals structural output inconsistencies in LLM predictions that require additional post-processing. These findings highlight the limitations of zero- and few-shot generative models for historical low-resource POS tagging and underline the importance of task-specific fine-tuning.

Keywords: POS tagging, Historical low-resource languages, Large Language Models

1. Introduction

Part-of-Speech (POS) tagging, a fundamental Natural Language Processing (NLP) task, has been extensively investigated over the years. While Large Language Models (LLMs) have demonstrated remarkable abilities across various NLP tasks (Qin et al., 2025), including strong performance on POS tagging for modern languages (Machado and Ruiz, 2024), and promising results for contemporary low-resource languages (Chang et al., 2024), their effectiveness on historical low-resource languages remains unexplored.

Many historical low-resource languages lack reliable NLP tools, or their accuracy is uncertain. This hinders the investigation of several languages that could benefit from the implementation of these tools. Additionally, existing LLM research has primarily concentrated on modern low-resource languages or higher-level tasks, often treating fundamental problems like POS tagging as solved, pushing these languages and their challenges even more into the background. Furthermore, recent studies have highlighted the influence of genre-specific structural patterns on model performance, which has been largely overlooked, despite evidence that low-resource genres can yield degraded results (Miani et al., 2026).

This study evaluates zero- and few-shot POS tagging performance of two generative models—Qwen2.5-3B-Instruct (Qwen Team, 2024), and Apertus-8B-Instruct-2509 (Apertus et al., 2025)—on four historical Germanic low-resource languages. Performance is assessed across both poetry and prose genres and benchmarked against a fine-tuned BERT baseline. To contextualize the performance on historical data, results are further compared with performance on two modern Germanic languages. The results revealed that fine-tuned encoder models consistently outperformed generative models across all languages and genres. The performance of the LLMs was substantially lower for the historical languages compared to modern ones, suggesting limited linguistic representation of these languages in the pretraining data. Additionally, error analysis highlighted structural inconsistencies in the LLM outputs, which required extra post-processing. Overall, the study demonstrated the superior effectiveness of fine-tuned encoder models for tasks such as POS tagging in historical low-resource languages. In contrast, generative models performed poorly on these languages, deviating from their generally strong performance trends observed in modern languages or with other tasks, where they showed comparatively good results.[1]

[1]All code used in the experiments is publicly available at https://github.com/irenemiani/historical-germanic.

Proceedings of LT4HALA 2026 @ LREC 2026, pages 125–138.
11 May 2026.

2. Related Work

The abilities of LLMs on POS tagging have been investigated in several studies (Blevins et al., 2023; Lai et al., 2023; Chang et al., 2023), primarily as part of broader surveys examining the linguistic knowledge of generative models. Other studies focused only on POS tagging, such as Machado and Ruiz (2024), who focused on evaluating three LLMs on the Universal Dependencies POS tagging for Brazilian Portuguese using few-shot prompting with only ten examples. Their work stressed the abilities of multilingual LLMs to leverage prior linguistic knowledge to perform POS tagging in zero- and few-shot settings, outperforming monolingual models.

A smaller number of studies focused on LLMs' abilities with low-resource historical languages. Stüssi and Ströbel (2024) examines the performance of LLMs on POS tagging for 16th-century Latin, which poses particular challenges, as most taggers have been trained on Classical Latin. They benchmark several tagging models and treebanks, evaluating accuracy across different Latin varieties. The authors find that fine-tuned GPT models can achieve competitive accuracies, with their best fine-tuned variant reaching an average accuracy of nearly 89% on curated treebank data, demonstrating that LLMs — when properly adapted — can bridge gaps left by traditional rule-based taggers on historical text.

Schöffel et al. (2025a) studied what factors influence LLMs' performance on POS tagging in low-resource Medieval Romance languages — specifically Occitan, Medieval French, and Medieval Spanish. By conducting a broad set of experiments, the authors showed how each design choice impacts tagging accuracy across corpora. They demonstrate that, despite carefully designed prompts and transfer methods, historical language variation still presents a challenge difficult to overcome for LLM-based POS taggers. In the same year, Schöffel et al. (2025b) further investigated the abilities and limitations of LLMs when applied historical low-resource languages without standardized orthography. In the study, the authors evaluated several open-source LLMs across two distinct Old Occitan corpora using a range of prompting strategies. The results show significant drops in model performance when confronted with diachronic language changes and non-standard orthography, highlighting biases and failure patterns not typically observed in high-resource languages.

Fang et al. (2025) presents a comparative analysis of word segmentation, POS tagging, and named entity recognition (NER) on historical Chinese sources dating from 1900–1950, contrasting LLM–based approaches with traditional NLP toolkits such as Jieba and spaCy. Because of the unique challenges of historical Chinese—including logographic script, the absence of explicit word boundaries, and diachronic linguistic variation—standard NLP pipelines have been proven to be ineffective. The results showed that LLM-based methods consistently outperform conventional tools across segmentation, POS tagging, and NER metrics, albeit at significantly higher computational costs. The study highlights an important performance–efficiency trade-off and suggests that contextual learning in LLMs can better accommodate genre and temporal variation in historical texts.

Despite growing interest in LLM-based tagging, we identified no studies that evaluate such approaches for the four historical low-resource languages investigated in this work; however, this task has been investigated in different works using traditional approaches. An Old English POS tagging tool is available as part of the CLTK library (Johnson et al., 2021), which was trained on the data from the ISWOC Treebank (Bech and Eide, 2014). The accuracy of the tool has not been tested. Recently, Miani et al. (2025) investigated cross-domain transfer learning for Old English poetry and prose, highlighting the impact of in-domain data on the performance of the models. Old High German was rarely investigated directly, but some studies evaluated existing POS taggers for Early Modern German corpora (Scheible et al., 2011; Ferreri Hanberry, 2015). Koleva et al. (2017) developed a POS tagger for Middle Low German. In addition, the four languages have also been the subject of investigation in Miani et al. (2026), who addressed cross-genre and cross-domain transfer learning on POS tagging with traditional methods, highlighting the impact of structural differences on models' performance.

3. Datasets

3.1. Historical Corpora

The Old English (henceforth OE) poetry data were retrieved from the York-Helsinki Parsed Corpus of Old English Poetry (YCOEP)[2]. Part of the same project is the York Toronto Helsinki Parsed Corpus of Old English (YCOE)[3] which includes approximately 1.5 million words of Old English prose annotated for syntax and morphology. Only the YCOE documentation was available; for this reason, it was adopted for both datasets. The texts are segmented into *tokens* representing main verbs with arguments and adjuncts, matrix inflectional

[2] https://www-users.york.ac.uk/~lang18/pcorpus.html

[3] https://penn-historical-corpora.uni-mannheim.de/ycoe/YCOEHomepage.html

phrases, complementizer phrases, or independent non-clausal utterances.

Both Old High German (henceforth OHG) poetry and prose were extracted from the Deutsch Diachron Digital, Referenzkorpus Altdeutsch (Version 1.2) dataset (Zeige Lars Erik, 2025). The texts, amounting to approximately 650.000 words, are structurally and linguistically annotated. The smallest unit is the *token*, which corresponds to both the edited and normalized versions of each word in the text. In addition to the Old High German manuscripts, some Old Saxon poetry and prose texts are also present; these were extracted and used alongside HeliPaD[4], an Old Saxon (henceforth OS) corpus containing 5.968 lines from the C manuscript of the poem *Heliand*. The annotation follows the same guidelines as the Old English corpora YCOEP and YCOE; therefore, the information included, the segmentation, and the POS tags are consistent across the three corpora.

Both Old Norse (henceforth ON) poetry and prose data were extracted from the Menotec collection[5] available on the INESS platform (Rosén et al., 2012). From the collection, five Old Norse manuscripts were selected: *Edda Regius*, containing 3665 lines, was used for poetry data; while the prose data were retrieved from the following texts, amounting to 10.318 lines: *Pamphilus*, *The Old Norwegian Homily Book*, *the Legendary Saga of St Olaf*, and *Strengleikar*. The texts present morphological and syntactical annotations that follow the guidelines for the annotation of Old Norwegian[6] texts by Haugen and Øverland (2014). [7]

From each corpus, the original textual forms and their POS tags were extracted. As the corpora employ different annotation schemes, all tags were mapped to the Universal Dependencies UPOS tag set (de Marneffe et al., 2021) following the mapping scheme proposed by Miani et al. (2026).

3.2. Modern Corpora

To compare the impact of historical languages on the performance of the generative models, two modern Germanic languages, English and German, were included in the experiment.

Modern English (henceforth ME) prose data were retrieved from the English part of the LinES Parallel Treebank[8](Ahrenberg, 2015). The dataset contains segments from nine different sources, totaling 106.305 tokens. Seven sub-corpora comprise literary works ranging from 1902 to 2014. These literary texts were extracted and used to assess LLM performance on Modern English prose.

To the extent of our knowledge, there is no modern German (henceforth MG) prose corpus with UD annotations available; for this reason, the closest dataset that seemed reasonable to be used to evaluate models' performance is UD German LIT[9], a treebank hosting texts from German literary history. It consists mainly of short, aphorism-like texts from the early Romantic period, roughly dated to the end of the 18th century, which is considered linguistically modern German (Besch and Wolf, 2009).

For both modern English and modern German poetry, the data were retrieved from the PoeTee dataset (Plecháč et al., 2025), which comprises approximately 335.000 poems and 90.000.000 tokens in 11 languages in JSON format with complete Universal Dependencies analysis; covering a period of time from the 13th to the 20th century. For English, approximately 40.000 poems are available from the Project Gutenberg, covering roughly 1650 to 1925, while for German, 74.000 from Metricalizer and Deutsches Lyrik Korpus, with a relatively small number of poems from roughly 1200 to 1600 and a much larger concentration from 1600 to 1925. Given the broad time span covered by the poetry data, we randomly sampled the majority of sentences from 1900 onward.

4. Experimental Setup

Fine-tuning BERT-based models has demonstrated strong performance on POS tagging tasks and has proven particularly effective for historical Germanic languages (Miani et al., 2026). This study adopts the same approach as a benchmark to compare against the generative models. In addition, contrastive models were trained for modern English and German to assess LLM performance on modern languages and genres, and to determine whether the observed difficulties persist beyond historical data.

4.1. Fine-Tuned Encoder Model

For each language, two models were trained: one only with poetry data and one only with prose data. The dataset sizes used to train the models were equal between poetry and prose and were the same as those used by Miani et al. (2026), except for Old Saxon. Due to the scarcity of Old Saxon prose data compared to poetry, we chose to use different dataset sizes for each genre to prevent the limited

[4] https://zenodo.org/records/4395040

[5] https://clarino.uib.no/iness

[6] Old Norwegian and Old Icelandic are collectively known as Old Norse in the Scandinavian languages (Haugen and Øverland, 2014).

[7] Documentation for the POS annotation was kindly provided by Paul Meurer and Odd Einar Haugen.

[8] https://github.com/UniversalDependencies/UD_English-LinES

[9] https://github.com/UniversalDependencies/UD_German-LIT

Lang.	Genre	Train	Dev.	Test	Tokens
OHG	poetry	5,327	665	667	63,203
	prose	5,327	665	667	101,830
OE	poetry	5,039	629	631	85,970
	prose	5,039	629	631	93,645
ON	poetry	3,029	378	380	39,779
	prose	3,029	378	380	60,082
OS	poetry	3,108	388	389	56,532
	prose	444	55	57	7,454
ME	poetry	4,080	510	511	44,770
	prose	4,080	510	511	94,627
MG	poetry	1,536	192	192	13,857
	prose	1,536	192	192	40,340

Table 1: Dataset statistics showing the number of sentences (train, development, and test splits) and tokens per genre for each language. All languages except Old Saxon contain equal amounts of prose and poetry data.

prose data from negatively affecting poetry results. All corpora were divided into 80% for training, 10% for development, and 10% for testing. The resulting dataset sizes are listed in Table 1.

Multilingual BERT (henceforth mBERT) was fine-tuned using MaChAmp (van der Goot et al., 2021), a toolkit designed for multi-task learning across various NLP tasks. Although MaChAmp supports multi-task learning, we trained separate models for each dataset, allowing for direct comparison of performance across individual languages and genres. For POS tagging, we used the `seq` task type, which applies a greedy softmax classification layer over the contextualized token embeddings provided by the encoder. All experiments were conducted using MaChAmp's default hyperparameter settings: a learning rate of 1e-4, a batch size of 32, an AdamW optimizer, and a dropout rate of 0.2. All models were trained for 20 epochs using three random seeds.

4.2. Generative Models

Two generative models were compared: Qwen2.5-3B-Instruct (Qwen Team, 2024) (henceforth Qwen), a multilingual model supporting over 29 languages, including English and German, and Apertus-8B-Instruct-2509 (Apertus et al., 2025) (henceforth Apertus), an open multilingual model designed to support more than 1,000 languages. These models were specifically chosen because of their pretraining data which include languages related to the historical ones — English, German, and Swedish — following the findings of Schöffel et al. (2025b), which demonstrated that such relatedness improves LLM performance. Both models were obtained from the HuggingFace model hub and used in their default instruction-tuned configurations without any additional fine-tuning. They were evaluated on the same test sets described in Table 1 that were used for the encoder models.

For both models, zero- and few-shot prompting strategies were adopted. The prompt design was based on the ones proposed in Schöffel et al. (2025b) and Machado and Ruiz (2024). Each prompt contained a single sentence, tokenized according to the structure present in the corpora. A system-level instruction specifying the task was adopted: POS tag the pre-tokenized input and output in valid JSON format. The models were informed of the language and genre of the data. In the few-shot setting, three examples per language and genre were randomly selected from the training sets and included in the prompt. Inference was performed on a single NVIDIA L40 GPU with greedy decoding. The maximum number of generated tokens was set to 2048. This setup ensured that results reflect the models' out-of-the-box instruction-following capabilities, without any fine-tuning or prompt optimization. The zero-shot prompt structure was the following:

```
You are a linguist performing UD
POS tagging on {language_genre}
data.
RULES:
- The sentences are ALREADY TOK-
ENIZED, so PRESERVE THE ORIGINAL
TOKENIZATION and just ADD POS
TAGS.
- NEVER modify, remove, correct,
normalize the text.
- NEVER add additional words or
punctuation.
- NEVER change the order of the
words or punctuation.
- NEVER drop punctuation.  ALWAYS
keep ALL punctuation marks, even
with multiple occurrences.
- Use ONLY these UD tags:  ADJ,
ADP, ADV, AUX, CCONJ, DET, INTJ,
NOUN, NUM, PART, PRON, PROPN,
PUNCT, SCONJ, VERB, X.
- NEVER add or change tags.
- ALL words need a tag.  NEVER
skip a word or a tag.
- Do NOT add or remove text.
- Do NOT add explanations.
- The OUTPUT FORMAT MUST BE JSON:
["word":  TOKEN, "upos":  TAG,
...]
- The JSON MUST BE VALID, with
commas between all key-value
pairs, no extra quotes, and
closed arrays.
- Close the JSON array properly.
Tag these tokens:
```

In the few-shot setting, the instruction `Follow the examples below:` was appended to the zero-shot prompt along with three examples, before

Original Tag(s)	Mapped Tag
WHADJP, ADJP	ADJ
WHADP	ADP
ADVERB, WHADVP, ADVP	ADV
MD	AUX
CONJ, ACONJ, ADCONJ, CC	CCONJ
WH_DET	DET
IN	INTJ
NEG, TO	PART
VERB:Past, VERB:Present, VERB:Infinitive, INF, MORPH, Mood:Ind|VerbForm:Fut	VERB
LOC, AComp, REL, AD, MWE, WDT	WRONG_UPOS
Missing UPOS	MISS_UPOS
Missing word and tag	MISS_WORD_UPOS

Table 2: POS tag mappings and placeholder substitutions applied during post-processing, based on the analysis of the development sets.

the final instruction `Tag these tokens:`.

Based on observations from the development set, a post-processing step was applied to the model outputs before evaluation. To ensure structural correctness, the output JSON files were validated using the Python package jsonschema[10]; all format errors were manually corrected. We also observed that some POS tags predicted by the LLMs were not valid UD tags; therefore, where the mapping was straightforward, the tags were corrected to valid ones, while the others were categorized as invalid. The resulting mappings are presented in Table 2.

After structural validation and annotation mapping, the predicted sentences were aligned with the gold data. Since the LLMs occasionally modified the input data in terms of context and length, a heuristic alignment strategy was employed. When the predicted and gold sentences match in content and length, structural alignment is not applied, preserving the predicted sentence. In all other cases, structural discrepancies are assumed, and alignment is performed. To address tokenization inconsistencies, the method generates multiple alignment candidates, including single-token matches, right-side merges (to combine consecutive predicted tokens), and left-side merges (to revise previous alignment decisions). Each candidate is evaluated using a similarity score against the gold token, and candidates exceeding a predefined threshold are selected. We tested threshold values of 0.7, 0.8, and 0.9, and found that 0.8 produced the most accurate alignments, as lower values introduced spurious matches while higher values caused valid matches to be missed. The UPOS tag assigned to the merged token corresponds to the last UPOS tag in the merged sequence. When no candidate meets the predefined similarity threshold, the algorithm employs bounded lookahead strategies in both the gold and predicted sequences to detect insertions and deletions. Missing tokens or those that can't be aligned through any strategy are marked with the placeholder tag, `MISS_WORD_UPOS`. As a result, all predicted sentences have the same number of tokens as the corresponding gold sentences.

[10] https://pypi.org/project/jsonschema/

5. Results

The results are evaluated using accuracy, macro F1 score, precision, recall, and per-tag F1 score. For mBERT, all metrics are averaged across seeds. Punctuation marks are excluded from all evaluations.

5.1. Overall Analysis

Tables 3 and 4 compare the performance of mBERT, Qwen, and Apertus across the six languages, in two genres: poetry and prose.

mBERT constantly achieves the highest scores on every language and metric in both genres, with accuracy typically exceeding 0.90 and F1 scores ranging from roughly 0.73 to 0.96. Despite being fine-tuned on considerably limited dataset sizes, with both genres, the encoder architecture demonstrates substantially stronger performance than the two generative models.

Apertus generally achieves slightly higher scores than Qwen in most configurations. The differences between poetry and prose are not particularly pronounced, suggesting that both models struggle regardless of genre-specific features. Across languages, however, the variation is much more evident: all four historical languages yield lower scores than the two modern ones. Old English and Old Norse produce the highest scores among the historical languages. In poetry, Old English achieves similar scores with both generative models, around 0.64 for accuracy and around 0.36 for F1 score. In prose, Apertus outperforms Qwen, particularly in the F1 score, which increases by 8% points. For Old Norse in both poetry and prose data, Apertus similarly achieves considerably higher scores than Qwen. In contrast, Old High German and Old Saxon produce the lowest scores in both genres, indicating that the models struggle significantly to accurately tag data in these languages. These significant struggles with historical languages are much more evident compared to the performance on modern languages. Despite the still substantially lower scores compared to the encoder architecture, both generative models achieve considerably

Lang.	Model	Acc.	F1	Prec.	Rec.
OHG	mBERT	**0.927**	**0.831**	**0.850**	**0.820**
	Q (0)	0.371	0.187	0.218	0.215
	Q (F)	0.444	0.268	0.300	0.305
	A (0)	0.382	0.213	0.302	0.281
	A (F)	*0.456*	*0.301*	*0.402*	*0.314*
OE	mBERT	**0.950**	**0.957**	**0.955**	**0.959**
	Q (0)	0.640	0.364	0.399	0.429
	Q (F)	0.673	0.401	0.412	0.458
	A (0)	0.637	0.368	0.413	0.430
	A (F)	*0.683*	*0.447*	*0.539*	*0.509*
ON	mBERT	**0.912**	**0.922**	**0.915**	**0.930**
	Q (0)	0.602	0.321	0.376	0.342
	Q (F)	0.632	0.406	0.437	0.437
	A (0)	0.663	0.422	0.462	0.421
	A (F)	*0.712*	*0.543*	*0.570*	*0.544*
OS	mBERT	**0.961**	**0.893**	**0.897**	**0.890**
	Q (0)	0.425	0.215	0.238	0.219
	Q (F)	0.551	0.352	0.372	0.351
	A (0)	0.446	0.238	0.266	0.251
	A (F)	*0.522*	*0.386*	*0.449*	*0.436*
ME	mBERT	**0.950**	**0.836**	**0.848**	**0.833**
	Q (0)	0.807	0.521	0.569	0.517
	Q (F)	0.817	0.503	0.545	0.501
	A (0)	0.850	0.622	0.673	0.608
	A (F)	*0.870*	*0.639*	*0.674*	*0.633*
MG	mBERT	**0.944**	**0.833**	**0.824**	**0.850**
	Q (0)	0.767	0.472	0.479	0.531
	Q (F)	0.782	0.475	0.475	0.510
	A (0)	0.805	0.536	*0.612*	0.556
	A (F)	*0.837*	*0.560*	0.604	*0.588*

Table 3: Evaluation results for the POS tagging performance of mBERT, Qwen, and Apertus on the **poetry** test sets. **(0)** and **(F)** indicate respectively the zero- and few-shot settings. **Bold** values indicate the best-performing model per language. *Italics* indicates the highest scores for the generative models.

Lang.	Model	Acc.	F1	Prec.	Rec.
OHG	mBERT	**0.931**	**0.845**	**0.849**	**0.842**
	Q (0)	0.397	0.204	0.271	0.216
	Q (F)	*0.484*	*0.287*	0.339	*0.270*
	A (0)	0.369	0.200	0.299	0.215
	A (F)	0.464	0.273	*0.387*	0.269
OE	mBERT	**0.966**	**0.926**	**0.928**	**0.923**
	Q (0)	0.611	0.310	0.366	0.305
	Q (F)	0.714	0.368	0.402	0.368
	A (0)	0.653	0.390	0.450	0.401
	A (F)	*0.746*	*0.498*	*0.547*	*0.513*
ON	mBERT	**0.923**	**0.791**	**0.793**	**0.790**
	Q (0)	0.548	0.264	0.328	0.273
	Q (F)	0.606	0.309	0.354	0.314
	A (0)	0.592	0.355	0.380	0.372
	A (F)	*0.679*	*0.419*	*0.425*	*0.480*
OS	mBERT	**0.875**	**0.730**	**0.744**	**0.722**
	Q (0)	0.367	0.199	0.230	0.226
	Q (F)	0.423	0.233	0.287	0.241
	A (0)	0.377	0.267	*0.381*	*0.297*
	A (F)	*0.466*	*0.286*	0.374	0.277
ME	mBERT	**0.978**	**0.900**	**0.913**	**0.892**
	Q (0)	0.803	0.514	0.570	0.555
	Q (F)	0.808	0.394	0.427	0.420
	A (0)	0.878	*0.645*	*0.649*	*0.683*
	A (F)	*0.891*	0.583	0.577	0.627
MG	mBERT	**0.958**	**0.859**	**0.871**	**0.850**
	Q (0)	0.735	0.434	0.444	0.478
	Q (F)	0.770	0.482	0.509	0.523
	A (0)	0.821	0.567	0.623	0.568
	A (F)	*0.859*	*0.677*	*0.719*	*0.718*

Table 4: Evaluation results for the POS tagging performance of mBERT, Qwen, and Apertus on the **prose** test sets. **(0)** and **(F)** indicate respectively the zero- and few-shot settings. **Bold** values indicate the best-performing model per language. *Italics* indicates the highest scores for the generative models.

higher accuracy scores in both genres, especially Apertus. This significant drop in performance from modern to historical languages underscores the severe lack of linguistic knowledge for the historical Germanic languages in these models. Despite the improvement in modern languages, scores remain lower than the encoder architecture, and the gap between accuracy and F1 scores is still more pronounced than for mBERT, suggesting particularly poor performance on rare tags.

The behaviors of the generative models in the zero-shot setting are consistent with the few-shot one; however, overall, the performance improves considerably. Particularly pronounced are the increments of the F1 score values for poetry in the historical languages. The performance of the models with the modern languages is not really influenced by the different prompting strategies; in fact, in some cases, such as Modern English prose, few-shot prompting even worsens the F1 score by 12 percentage points. In addition, despite the improvement in results with the few-shot setting, the gap between accuracy and F1 scores remains far more pronounced compared to the encoder models, indicating persistent struggles with rare tags.

5.2. Per-Tag Analysis

Tables 5 and 6 present the per-tag F1 scores for poetry and prose, respectively, for each model and language in the few-shot setting. The zero-shot results follow similar patterns and are reported in Appendix A, in Tables 8 and 9. Appendix A also includes Figures 1 and 2, showing the relative frequency distributions for all datasets in poetry and prose, respectively.

The per-tag analysis confirms the poor performance of the generative models compared to the encoder architecture, as suggested by the overall metrics in Tables 3 and 4. With both genres, both

	OHG				OE				ON				OS				ME				MG			
POS	*mBERT*	*Qwen*	*Apertus*	*Support*	*mBERT*	*Qwen*	*Apertus*	*Support*	*mBERT*	*Qwen*	*Apertus*	*Support*	*mBERT*	*Qwen*	*Apertus*	*Support*	*mBERT*	*Qwen*	*Apertus*	*Support*	*mBERT*	*Qwen*	*Apertus*	*Support*
ADJ	.804	.301	.196	250	.888	.526	.565	711	.816	.423	.607	265	.908	.335	.320	368	.910	.785	.810	332	.884	.712	.793	116
ADP	.967	.603	.569	338	.972	.795	.803	673	.930	.707	.782	233	.984	.661	.645	358	.953	.796	.867	405	.985	.777	.911	108
ADV	.912	.430	.380	694	.917	.435	.526	537	.886	.406	.639	283	.967	.409	.450	470	.891	.667	.665	203	.871	.638	.694	84
AUX	.891	.282	.227	236	.935	.531	.246	180	.854	.483	.265	81	.913	.233	.165	125	.954	.571	.879	174	.965	.611	.737	45
CCONJ	.930	.099	.088	149	.980	.863	.447	233	.967	.825	.866	124	.992	.847	.834	101	.989	.888	.977	198	.973	.843	.872	66
DET	.949	.160	.430	713	.964	.254	.566	505	.933	.163	.502	150	.974	.305	.372	400	.986	.895	.920	430	.997	.829	.893	150
INTJ	.743	.027	.024	9	1.000	.353	.387	6	.000	.000	.000	0	1.000	.000	.091	1	.778	.571	.465	14	.000	.000	.000	0
NOUN	.916	.495	.501	873	.957	.785	.800	2116	.900	.710	.814	770	.960	.655	.593	1035	.964	.941	.942	831	.957	.895	.910	267
NUM	.691	.143	.222	13	.900	.468	.462	37	.936	.415	.800	12	.856	.000	.222	16	.929	.615	.571	14	.889	.500	.500	1
PART	.959	.248	.479	106	.985	.000	.102	56	1.000	.000	.000	1	.986	.000	.000	70	.978	.183	.598	84	.894	.118	.250	7
PRON	.969	.556	.646	854	.991	.762	.835	543	.970	.694	.711	347	.985	.622	.614	648	.982	.910	.940	428	.980	.848	.862	162
PROPN	.930	.559	.364	74	.963	.630	.659	180	.942	.846	.883	166	.981	.849	.796	141	.852	.733	.781	75	.750	.471	.432	12
SCONJ	.878	.399	.437	183	.956	.399	.403	117	.935	.513	.508	161	.947	.554	.561	136	.827	.145	.564	97	.915	.000	.788	16
VERB	.931	.525	.546	843	.948	.733	.721	1143	.915	.712	.775	692	.948	.512	.505	733	.943	.850	.891	526	.924	.832	.876	146
X	.000	.000	.000	2	1.000	.476	.075	6	.000	.000	.000	0	.000	.000	.000	0	.444	.000	.000	2	.510	.000	.000	7

Table 5: Per-tag F1 scores and corresponding support values for POS tagging with fine-tuned mBERT and generative models (Qwen, Apertus) in the **few-shot** setting on **poetry** test sets.

	OHG				OE				ON				OS				ME				MG			
POS	*mBERT*	*Qwen*	*Apertus*	*Support*	*mBERT*	*Qwen*	*Apertus*	*Support*	*mBERT*	*Qwen*	*Apertus*	*Support*	*mBERT*	*Qwen*	*Apertus*	*Support*	*mBERT*	*Qwen*	*Apertus*	*Support*	*mBERT*	*Qwen*	*Apertus*	*Support*
ADJ	.818	.258	.235	403	.887	.501	.569	504	.778	.463	.543	279	.644	.130	.241	24	.944	.829	.867	603	.919	.855	.859	345
ADP	.965	.648	.573	702	.980	.796	.826	875	.975	.665	.824	476	.918	.471	.611	63	.981	.816	.898	878	.967	.728	.843	259
ADV	.870	.257	.241	599	.945	.518	.651	634	.928	.549	.673	461	.620	.109	.185	29	.954	.661	.767	483	.921	.748	.793	333
AUX	.921	.383	.493	386	.944	.310	.191	189	.909	.358	.078	114	.891	.478	.576	47	.975	.409	.869	476	.478	.413	.849	231
CCONJ	.961	.602	.402	522	.996	.881	.923	598	.985	.801	.827	410	.968	.400	.211	51	.990	.927	.971	264	.933	.828	.898	162
DET	.956	.298	.514	1165	.976	.486	.632	886	.926	.243	.467	583	.916	.245	.324	75	.998	.938	.938	777	.987	.806	.904	522
INTJ	.814	.082	.083	31	1.000	.000	.105	4	.000	.000	.154	1	.000	.250	.000	3	.884	.533	.323	6	.000	.000	.000	0
NOUN	.930	.570	.520	1593	.958	.802	.841	1436	.895	.713	.774	1024	.846	.504	.542	123	.982	.926	.943	1570	.983	.952	.964	699
NUM	.792	.096	.062	49	.962	.670	.480	94	.000	.000	.000	0	.000	.000	.000	0	.955	.782	.839	59	.909	.556	1.000	6
PART	.950	.385	.074	181	.978	.000	.021	92	.929	.000	.000	35	.899	.000	.000	16	.996	.082	.706	190	.967	.102	.113	100
PRON	.965	.486	.524	837	.991	.848	.886	1040	.958	.667	.736	468	.957	.691	.676	74	.993	.880	.897	1009	.988	.792	.831	283
PROPN	.941	.597	.484	226	.980	.879	.920	346	.852	.611	.632	144	.943	.318	.400	17	.977	.741	.853	239	.774	.658	.716	35
SCONJ	.846	.065	.135	268	.957	.573	.500	295	.954	.289	.587	315	.746	.000	.273	14	.959	.055	.733	139	.931	.254	.776	55
VERB	.944	.559	.572	1418	.969	.786	.777	1270	.925	.725	.740	986	.869	.368	.539	116	.971	.824	.927	1071	.942	.741	.852	311
X	.000	.000	.000	0	.968	.043	.140	84	.848	.098	.098	29	.000	.000	.000	0	.841	.060	.129	4	.711	.250	.429	3

Table 6: Per-tag F1 scores and corresponding support values for POS tagging with fine-tuned mBERT and generative models (Qwen, Apertus) in the **few-shot** setting on **prose** test sets.

Qwen and Apertus struggle more with historical languages compared to modern ones. High-frequency categories, such as NOUN, VERB, PRON, and DET, which achieve the highest scores with mBERT, typically exceeding 0.85 and frequently surpassing 0.95, have the worst scores with the generative models, especially with the historical languages. Rare classes such as NUM, INTJ, and X, are penalized by their low frequency, which leads to low performance with mBERT, but influences the LLMs negatively. Interestingly, also ADJ appears to be a challenging tag: in both genres, but especially in the poetry, mBERT manages to barely surpass 0.91, and the LLMs achieve comparable scores only in the modern languages. This behavior could be justified by the overlaps with other classes, such as nouns, participles, or adverbs. For instance, Old English has the highest number of ADJ in the test set, and the same words can also be tagged as PROPN, NOUN, ADV, VERB, DET, and AUX.

Comparing the performance of the two LLMs reveals that Apertus tends to achieve slightly higher scores compared to Qwen in the poetry data, but their behaviors are similar. In the prose data, Apertus performs better, with higher scores in almost all languages. As highlighted by the relative frequency distributions presented in Figures 1 and 2 in Appendix A, and also by previous studies (Miani et al., 2026), the structural differences of the two genres are consistent and may influence the model's performance. In the poetry data, the label distribution appears unbalanced, with a smaller number of high-frequency classes accounting for large proportions of tokens; in contrast to the prose data, which exhibits a more balanced distribution across labels. Although rare classes are usually more prone to misclassification, the more balanced nature of the prose data can help support more stable performance. Despite the fact that neither of the generative models is able to surpass the encoder architecture, Apertus appears to be able to address this class imbalance better than the other LLM.

5.3. Structural Output Errors and Schema Violations

Table 7 presents the error rates for JSON formatting issues and incorrect additional tags in the test sets with both prompting strategies. The JSON errors column indicates the proportion of malformed outputs generated by Qwen and Apertus. The files were validated with a JSON checker, and the resulting percentage was based on the number of mistakes identified with the tool. The percentage

Lang.	Genre	JSON errors		MAP_UPOS		WRONG_UPOS		MISS_UPOS		MISS_WORD_UPOS	
		Q	A	Q	A	Q	A	Q	A	Q	A
Zero-shot											
OHG	poe	0.75	3.60	0.28	-	0.04	-	-	0.41	0.36	1.14
	pro	6.60	4.50	0.33	-	0.02	-	-	0.89	0.73	1.55
OE	poe	1.43	7.45	1.59	0.07	0.14	-	-	0.17	0.26	1.76
	pro	2.54	8.56	1.57	-	0.32	0.01	0.06	0.84	1.16	2.67
ON	poe	1.05	3.68	0.97	-	0.27	-	-	-	0.15	0.37
	pro	3.16	6.84	1.07	-	0.28	-	-	-	1.24	0.56
OS	poe	5.40	7.71	0.15	-	0.17	-	-	0.78	0.04	2.39
	pro	5.26	10.53	0.77	-	-	-	-	5.83	0.15	9.66
ME	poe	2.15	6.26	2.41	0.03	0.05	-	-	1.29	1.15	1.05
	pro	9.59	34.44	0.89	0.28	0.40	-	-	0.88	0.50	3.36
MG	poe	1.04	3.65	2.78	-	-	-	-	0.51	0.42	0.76
	pro	4.69	30.73	1.08	-	0.12	-	-	0.21	1.41	3.14
Few-shot											
OHG	poe	1.35	1.95	0.15	-	0.02	-	-	0.13	0.58	0.69
	pro	16.04	4.20	0.11	-	0.05	-	-	0.79	0.44	1.19
OE	poe	1.27	2.69	0.94	0.07	0.11	-	-	-	0.18	0.47
	pro	3.33	4.12	0.08	0.04	0.31	-	-	0.49	0.54	0.26
ON	poe	0.79	0.79	0.37	-	-	-	-	-	0.27	0.03
	pro	4.47	3.95	0.30	-	0.56	-	-	-	0.36	0.19
OS	poe	7.71	0.26	-	-	-	-	-	-	0.13	0.02
	pro	15.79	5.26	0.31	-	-	-	-	5.83	0.46	7.82
ME	poe	1.17	2.35	2.18	0.29	0.18	-	-	-	1.15	0.16
	pro	18.98	27.79	0.30	0.05	0.54	0.04	-	0.05	0.57	2.09
MG	poe	3.12	3.12	0.25	-	-	-	-	-	0.93	0.34
	pro	44.27	23.44	0.33	-	0.03	-	-	-	1.73	1.32

Table 7: Percentage of structural and tagging errors produced by Qwen (Q) and Apertus (A) across languages and genres in the zero- and few-shot settings. JSON errors correspond to malformed outputs or schema violations, while tag-related errors include mapped erroneous UPOS tags (MAP_UPOS), invalid UPOS labels (WRONG_UPOS), missing POS tags (MISS_UPOS), and missing tokens introduced during alignment (MISS_WORD_UPOS).

can include different errors, such as structurally invalid JSON (e.g., missing brackets, omitted required schema elements), and incomplete or improperly formatted sentences. The errors tend to decrease from the zero-shot to the few-shot setting, especially for Apertus, suggesting that the inclusion of examples helps the model mitigate structural issues. Qwen shows improvements with Old Saxon, but error rates generally increase across the other languages, indicating that the few-shot examples are less beneficial for this model than for Apertus. In both settings, the models exhibit higher error rates on prose than on poetry, likely due to the longer sentence lengths in prose. Overall, Apertus presents higher percentages of JSON errors, especially with modern English and German prose. These two test sets are also the ones with higher average sentence lengths, respectively 17 and 20 tokens; this increased token count could potentially lead to interruptions in the generation process and consequent output degradation, which would explain the high error rates.

MAP_UPOS reports the percentage of invalid UD tags that were mapped to valid ones following Table 2. Qwen produces more invalid tags than Apertus, with both prompting strategies, particularly with Old English, Old Norse, modern English and modern German. Apertus rarely deviates from the UD tagset, with few instances limited to Old and modern English, suggesting that it adheres more closely to the prompt guidelines and requires less post-processing.

The placeholder WRONG_UPOS was used to mark tags that do not belong to the Universal Dependencies tag set and could not be mapped. The reported percentage includes both the tags listed in Table 2 and additional invalid ones generated in the test data. As for MAP_UPOS, Apertus adheres more closely to the prompt instructions and the UD tag inventory, producing only one extra tag in Old English with the zero-shot setting and in modern English with the few-shot one. In contrast, Qwen introduces extra tags in all test sets with the zero-shot setting, but slightly improves the performance

with the few-shot approach.

The MISS_UPOS placeholder was introduced to mark tokens for which no POS tag was provided. Unlike the MAP_UPOS and WRONG_UPOS errors, Qwen successfully assigned tags to nearly all tokens across languages, with only a few omissions in Old English prose in the zero-shot setting. These errors are absent in the few-shot setting. Apertus, however, failed to tag multiple token constructions in most test sets, except for Old Norse. The highest percentages are observed in the Old Saxon prose dataset, likely due to the relatively small size of the test set that amplifies the impact of the error on the overall rate. For the other languages, the scores do not exceed the 1% threshold. The model improves with the few-shot setting, reducing error rates across most languages and eliminating them completely in modern German.

The MISS_WORD_UPOS placeholder was used during the alignment procedure to represent missing tokens and their corresponding POS tags. With both prompting strategies, both LLMs exhibit errors of this type, although Apertus shows higher rates than Qwen, particularly in Old Saxon and the modern language datasets. With the few-shot prompting strategy, Qwen presents worse performance, slightly increasing the error rates, while Apertus, still presents higher scores than the other LLM, but shows a general improvement compared to the zero-shot setting. The results imply that both models were unable to consistently preserve the required input structure, often modifying it in ways that necessitated manual correction.

6. Discussion

The evaluation of the generative models' performance on zero- and few-shot POS tagging on historical Germanic languages revealed the absence of linguistic knowledge for these languages in the LLMs' pretraining data. Despite improved results in the few-shot setting, neither model was able to outperform the fine-tuned mBERT models, which achieved high scores across all languages, genres, and prompting strategies. This interpretation is further supported by the markedly better performance of the generative models on modern English and German data: although their scores still fell short of mBERT, both models obtained substantially stronger results on the modern datasets, suggesting that they possess sufficient linguistic knowledge for these contemporary languages to perform the task more successfully, but critically lack it for their historical counterparts.

The per-tag analysis reinforces these findings. mBERT clearly outperforms both LLMs across all tags, and both generative models achieve higher scores for modern languages than for historical ones. Apertus generally performs slightly better than Qwen with both poetry and prose. The results also underline the impact of the substantial differences between the genres: the unbalanced nature of the poetry data results in fewer high-frequency classes compared to the prose data, which presents a more balanced distribution. Such distributional characteristics may increase the difficulty of correctly predicting rare classes, leading to lower overall scores. While mBERT appears more robust to these effects, both LLMs are facing issues in raising the scores for low-frequency tags.

The error analysis further reveals a fundamental limitation of generative models: their persistent tendency to alter or restructure the required output format. While the few-shot setting mitigates structural issues for one generative model, it does not eliminate the errors, and substantial post-processing remains necessary — an issue that does not arise with fine-tuned encoder-based architectures. Although post-processing may be acceptable when model performance is competitive, the consistently low scores achieved by the LLMs suggest that the additional effort is not justified, particularly when fine-tuned encoder models deliver substantially superior results without the need for post-processing.

7. Conclusions

The study evaluates zero- and few-shot POS tagging performance of two generative models—Qwen2.5-3B-Instruct and Apertus-8B-Instruct-2509—on four historical Germanic low-resource languages and two literary genres. The results are benchmarked against a fine-tuned Multilingual BERT baseline. Two modern languages are used to investigate the impact of historical data on the models' performance.

The results demonstrate that fine-tuned encoder models consistently outperform the generative models on both historical and modern datasets. While, with both prompting strategies, the generative models achieve relatively weak performance on the historical languages, they obtain noticeably better — though still inferior — results on the modern languages, suggesting limited exposure to or representation of the historical varieties in their training data. Error analysis further highlights the tendency of LLMs to generate malformed outputs requiring manual correction and post-processing, not needed with encoder models.

Future work will extend this investigation to additional NLP tasks and explore the fine-tuning of generative LLMs to assess whether task-specific adaptation can improve performance.

8. Limitations

This study evaluates two generative models against a benchmark on the POS tagging task for four historical low-resource Germanic languages.

All four languages are morphologically rich, and their original POS tag sets were highly fine-grained. Despite the UPOS adopted mapping being the most suitable approach for comparison of model performance, it may lead to linguistic information losses. In addition, original annotation errors may influence the quality of the UPOS labels.

Due to the limited availability of UD-annotated corpora covering both genres for the modern languages, the data spans an extended time period, potentially introducing diachronic variation that may have negatively affected model performance.

The generative models tend to modify the original input through normalization, paraphrasing, tokenization, or lexical substitution, rendering traditional string-matching alignment methods ineffective. The adopted custom alignment function attempts to address these modifications, but it relies on heuristic assumptions derived from the development sets. Consequently, it may fail to account for novel textual characteristics not encountered during development, potentially affecting evaluation accuracy.

To maintain comparability across experiments, dataset sizes were substantially reduced, which may have constrained model performance and obscured potential improvements achievable with larger training sets. Additionally, all models were run with default hyperparameters and greedy decoding to ensure deterministic and reproducible outputs, though different prompts or decoding settings could lead to different results, and fine-tuning the models on the target task might improve performance substantially.

9. Acknowledgments

This work has been supported by the Swedish Graduate School of Digital Philology, funded by the Swedish Research Council (grant 2022-06343). Computations were enabled by resources at the Uppsala Multidisciplinary Center for Advanced Computational Science (UPPMAX).

10. Bibliographical References

Lars Ahrenberg. 2015. Converting an English-Swedish parallel treebank to Universal Dependencies. In *Proceedings of the Third International Conference on Dependency Linguistics (DepLing 2015)*, pages 10–19, Uppsala, Sweden. Association for Computational Linguistics.

Project Apertus, Alejandro Hernández-Cano, Alexander Hägele, Allen Hao Huang, Angelika Romanou, Antoni-Joan Solergibert, Barna Pasztor, Bettina Messmer, Dhia Garbaya, Eduard Frank Ďurech, Ido Hakimi, Juan García Giraldo, Mete Ismayilzada, Negar Foroutan, Skander Moalla, Tiancheng Chen, Vinko Sabolčec, Yixuan Xu, Michael Aerni, Badr AlKhamissi, Inés Altemir Mariñas, Mohammad Hossein Amani, Matin Ansaripour, Ilia Badanin, Harold Benoit, Emanuela Boros, Nicholas Browning, Fabian Bösch, Maximilian Böther, Niklas Canova, Camille Challier, Clement Charmillot, Jonathan Coles, Jan Deriu, Arnout Devos, Lukas Drescher, Daniil Dzenhaliou, Maud Ehrmann, Dongyang Fan, Simin Fan, Silin Gao, Miguel Gila, María Grandury, Diba Hashemi, Alexander Hoyle, Jiaming Jiang, Mark Klein, Andrei Kucharavy, Anastasiia Kucherenko, Frederike Lübeck, Roman Machacek, Theofilos Manitaras, Andreas Marfurt, Kyle Matoba, Simon Matrenok, Henrique Mendonça, Fawzi Roberto Mohamed, Syrielle Montariol, Luca Mouchel, Sven Najem-Meyer, Jingwei Ni, Gennaro Oliva, Matteo Pagliardini, Elia Palme, Andrei Panferov, Léo Paoletti, Marco Passerini, Ivan Pavlov, Auguste Poiroux, Kaustubh Ponkshe, Nathan Ranchin, Javi Rando, Mathieu Sauser, Jakhongir Saydaliev, Muhammad Ali Sayfiddinov, Marian Schneider, Stefano Schuppli, Marco Scialanga, Andrei Semenov, Kumar Shridhar, Raghav Singhal, Anna Sotnikova, Alexander Sternfeld, Ayush Kumar Tarun, Paul Teiletche, Jannis Vamvas, Xiaozhe Yao, Hao Zhao, Alexander Ilic, Ana Klimovic, Andreas Krause, Caglar Gulcehre, David Rosenthal, Elliott Ash, Florian Tramèr, Joost VandeVondele, Livio Veraldi, Martin Rajman, Thomas Schulthess, Torsten Hoefler, Antoine Bosselut, Martin Jaggi, and Imanol Schlag. 2025. Apertus: Democratizing open and compliant llms for global language environments.

Kristin Bech and Kristine Eide. 2014. The ISWOC corpus. Department of Literature, Area Studies and European Languages, University of Oslo.

Werner Besch and Norbert Richard Wolf. 2009. *Geschichte der deutschen Sprache*. Erich Schmidt, Berlin.

Terra Blevins, Hila Gonen, and Luke Zettlemoyer. 2023. Prompting language models for linguistic structure. *arXiv preprint arXiv:2211.07830*.

Chao-Yang Chang, Yan-Ming Lin, Chih-Chung Kuo, Yen-Chun Lai, Chao-Shih Huang, Yuan-Fu Liao, and Tsun-guan Thiann. 2024. A preliminary study on Taiwanese POS taggers: Leveraging Chinese in the absence of Taiwanese POS annotation datasets. In *2024 27th Conference of the Oriental COCOSDA International Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques (O-COCOSDA)*, pages 1–6.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2023. A survey on evaluation of large language models. *arXiv preprint arXiv:2307.03109*.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Zhao Fang, Liang-Chun Wu, Xuening Kong, and Spencer Dean Stewart. 2025. A comparative analysis of word segmentation, part-of-speech tagging, and named entity recognition for historical Chinese sources, 1900–1950. *arXiv preprint arXiv:2503.19844*.

Brendan Ferreri Hanberry. 2015. Application of a POS tagger to a novel chronological division of Early Modern German text. Master's thesis, University of North Carolina at Chapel Hill.

Odd Einar Haugen and Fartein Th. Øverland. 2014. Guidelines for morphological and syntactic annotation of Old Norwegian texts. *Bergen Language and Linguistics Studies (BeLLS)*, 4(2).

Kyle P. Johnson, Patrick J. Burns, John Stewart, Todd Cook, Clément Besnier, and William J. B. Mattingly. 2021. The Classical Language Toolkit: An NLP framework for pre-modern languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 20–29, Online. Association for Computational Linguistics.

Mariya Koleva, Melissa Farasyn, Bart Desmet, Anne Breitbarth, and Veronique Hoste. 2017. An automatic part-of-speech tagger for Middle Low German. *International Journal of Corpus Linguistics*, 22(1):108–141.

Viet Dac Lai, Nghia Trung Ngo, Amir Pouran Ben Veyseh, Hieu Man, Franck Dernoncourt, Trung Bui, and Thien Huu Nguyen. 2023. ChatGPT beyond English: Towards a comprehensive evaluation of large language models in multilingual learning. *arXiv preprint arXiv:2304.05613*.

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 Linguistics.

Irene Miani, Sara Stymne, and Gregory R. Darwin. 2025. Cross-genre learning for Old English poetry POS tagging. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 708–724, Vienna, Austria. Association for Computational Linguistics.

Irene Miani, Sara Stymne, and Gregory R. Darwin. 2026. Cross-lingual and cross-domain transfer learning for POS tagging in historical Germanic low-resource languages. In *Proceedings of the Second Workshop on Language Models for Low-Resource Languages (LoResLM 2026)*, pages 542–558, Rabat, Morocco. Association for Computational Linguistics.

Pavel Plecháč, Andris Šeļa, Héctor Bermúdez Sabel, Katharina Bobenhausen, Soňa Cinková, I. L. Dale, Émilie Delente, Marco De Sisto, Thomas Haider, Benjamin Hammerich, Péter Horváth, Randi Kvinnsland, Niko Kočnik, Radek Kolár, Kirill Korchagin, Anna Martynenko, Alexander Mittmann, Bence Nagy, Begoña Navarro Colorado, and Diana Sitchinava. 2025. PoeTree: Poetry corpora in Czech, English, French, German, Hungarian, Italian, Norwegian, Portuguese, Russian, Slovenian, and Spanish (1.0.0) [data set].

Libo Qin, Qiguang Chen, Xiachong Feng, Yang Wu, Yongheng Zhang, Yinghui Li, Min Li, Wanxiang Che, and Philip S. Yu. 2025. Large language models meet NLP: A survey. *arXiv preprint arXiv:2405.12819*.

Qwen Team. 2024. Qwen2.5: A party of foundation models. Qwen Blog.

Victoria Rosén, Koenraad De Smedt, Paul Meurer, and Helge Dyvik. 2012. An open infrastructure for advanced treebanking. In *META-RESEARCH Workshop on Advanced Treebanking at LREC 2012*, pages 22–29, Istanbul, Turkey.

Silke Scheible, Richard J. Whitt, Martin Durrell, and Paul Bennett. 2011. Evaluating an 'off-the-shelf' POS-tagger on Early Modern German text. In *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 19–23, Portland, OR, USA. Association for Computational Linguistics.

Matthias Schöffel, Esteban Garcés Arias, Marinus Wiedner, Paula Ruppert, Meimingwei Li, Christian Heumann, and Matthias Aßenmacher. 2025a. Unveiling factors for enhanced POS tagging: A study of low-resource medieval romance languages. *arXiv preprint arXiv:2506.17715*.

Matthias Schöffel, Marinus Wiedner, Esteban Garces Arias, Paula Ruppert, Christian Heumann, and Matthias Aßenmacher. 2025b. 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.

Elina Stüssi and Phillip Ströbel. 2024. Part-of-speech tagging of 16th-century Latin with GPT. In *Proceedings of the 8th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature (LaTeCH-CLfL 2024)*, pages 196–206, St. Julians, Malta. Association for Computational Linguistics.

Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021. Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 176–197, Online. Association for Computational Linguistics.

Klotz Martin Donhauser Karin Gippert Jost Lühr Rosemarie Zeige Lars Erik, Schnelle Gohar. 2025. Deutsch Diachron Digital – Referenzkorpus Altdeutsch (Version 1.2). Humboldt-Universität zu Berlin.

A. Appendix

A.1. Relative POS Tag Frequency

The percentages for the relative frequency of POS tags in the two genres are presented in Figures 1 and 2 for poetry and prose, respectively.

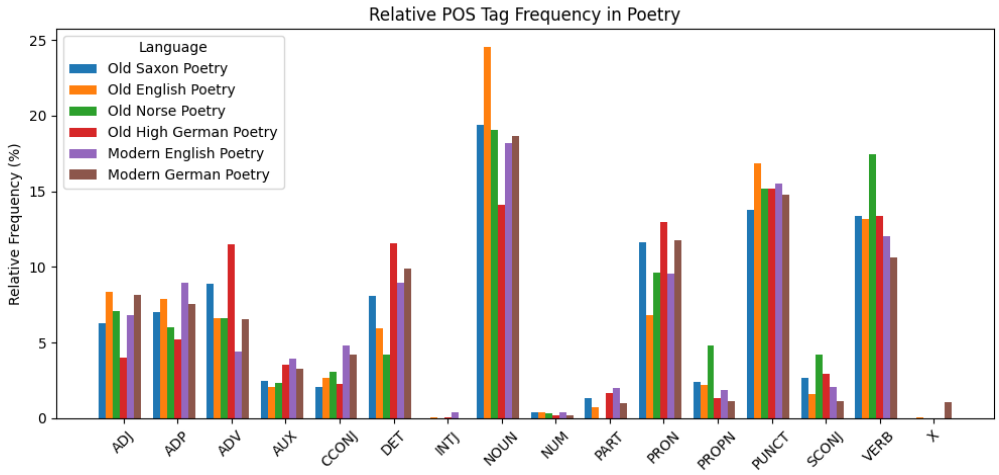

Figure 1: Relative POS tag frequency in the **poetry** datasets.

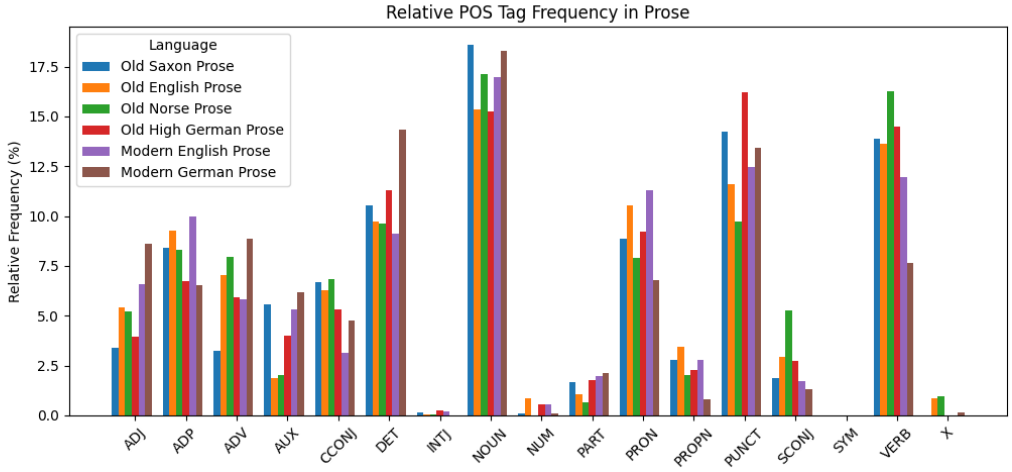

Figure 2: Relative POS tag frequency in the **prose** datasets.

A.2. Additional Results

Tables 8 and 9 present the F1 scores in the zero-shot setting for poetry and prose.

	OHG				OE				ON				OS				ME				MG			
POS	*mBERT*	*Qwen*	*Apertus*	*Support*	*mBERT*	*Qwen*	*Apertus*	*Support*	*mBERT*	*Qwen*	*Apertus*	*Support*	*mBERT*	*Qwen*	*Apertus*	*Support*	*mBERT*	*Qwen*	*Apertus*	*Support*	*mBERT*	*Qwen*	*Apertus*	*Support*
ADJ	.804	.208	.183	250	.888	.468	.524	711	.816	.399	.573	265	.908	.285	.299	368	.910	.778	.784	332	.884	.717	.758	116
ADP	.967	.527	.521	338	.972	.753	.778	673	.930	.674	.689	233	.984	.388	.604	358	.953	.797	.864	405	.985	.766	.889	108
ADV	.912	.268	.271	694	.917	.367	.526	537	.886	.400	.594	283	.967	.271	.458	470	.891	.600	.639	203	.871	.632	.725	84
AUX	.891	.096	.088	236	.935	.271	.155	180	.854	.311	.080	81	.913	.078	.027	125	.954	.414	.830	174	.965	.400	.433	45
CCONJ	.930	.091	.052	149	.980	.886	.407	233	.967	.780	.639	124	.992	.118	.000	101	.989	.946	.969	198	.973	.832	.872	66
DET	.949	.091	.310	713	.964	.315	.561	505	.933	.085	.297	150	.974	.140	.373	400	.986	.899	.906	430	.997	.853	.894	150
INTJ	.743	.042	.048	9	1.000	.308	.375	6	.000	.000	.000	0	1.000	.000	.000	1	.778	.600	.390	14	.000	.000	.000	0
NOUN	.916	.469	.447	873	.957	.765	.748	2116	.900	.698	.757	770	.960	.611	.540	1035	.964	.927	.941	831	.957	.868	.898	267
NUM	.691	.095	.261	13	.900	.480	.440	37	.936	.375	.769	12	.856	.062	.182	16	.929	.667	.727	14	.889	.500	.500	1
PART	.959	.130	.018	106	.985	.064	.000	56	1.000	.000	.000	1	.986	.101	.000	70	.978	.200	.483	84	.894	.194	.250	7
PRON	.969	.549	.601	854	.991	.713	.814	543	.970	.689	.714	347	.985	.453	.509	648	.982	.916	.932	428	.980	.866	.823	162
PROPN	.930	.497	.254	74	.963	.598	.541	180	.942	.870	.855	166	.981	.607	.454	141	.852	.705	.786	75	.750	.383	.421	12
SCONJ	.878	.036	.029	183	.956	.037	.026	117	.935	.126	.485	161	.947	.045	.105	136	.827	.093	.444	97	.915	.000	.720	16
VERB	.931	.461	.521	843	.948	.699	.671	1143	.915	.697	.729	692	.948	.500	.500	733	.943	.829	.872	526	.924	.806	.814	146
X	.000	.000	.008	2	1.000	.190	.053	6	.000	.000	.000	0	.000	.000	.000	0	.444	.000	.000	2	.510	.200	.118	7

Table 8: Per-tag F1 scores and corresponding support values for POS tagging with fine-tuned mBERT and generative models (Qwen, Apertus) in the **zero-shot** setting on **poetry** test sets.

	OHG				OE				ON				OS				ME				MG			
POS	*mBERT*	*Qwen*	*Apertus*	*Support*	*mBERT*	*Qwen*	*Apertus*	*Support*	*mBERT*	*Qwen*	*Apertus*	*Support*	*mBERT*	*Qwen*	*Apertus*	*Support*	*mBERT*	*Qwen*	*Apertus*	*Support*	*mBERT*	*Qwen*	*Apertus*	*Support*
ADJ	.818	.245	.234	403	.887	.434	.483	504	.778	.445	.500	279	.644	.089	.230	24	.944	.827	.857	603	.919	.819	.837	345
ADP	.965	.545	.498	702	.980	.711	.768	875	.975	.610	.695	476	.918	.397	.534	63	.981	.814	.895	878	.967	.749	.841	259
ADV	.870	.239	.266	599	.945	.453	.613	634	.928	.520	.560	461	.620	.172	.271	29	.954	.657	.744	483	.921	.731	.780	333
AUX	.921	.068	.147	386	.944	.290	.237	189	.909	.199	.056	114	.891	.109	.192	47	.975	.320	.853	476	.957	.177	.586	231
CCONJ	.961	.121	.123	522	.996	.431	.613	598	.985	.614	.542	410	.968	.136	.145	51	.990	.965	.975	264	.933	.828	.889	162
DET	.956	.201	.401	1165	.976	.358	.596	886	.926	.211	.334	583	.916	.089	.141	75	.998	.924	.927	777	.987	.759	.887	522
INTJ	.814	.048	.060	31	1.000	.000	.093	4	.000	.000	.000	1	.000	.222	.444	3	.884	.769	.345	6	.000	.000	.000	0
NOUN	.930	.522	.453	1593	.958	.767	.772	1436	.895	.666	.724	1024	.846	.501	.514	123	.982	.925	.936	1570	.983	.934	.946	699
NUM	.792	.156	.033	49	.962	.651	.571	94	.000	.000	.000	0	.000	.000	.000	0	.955	.791	.852	59	.909	.667	.909	6
PART	.950	.040	.000	181	.978	.039	.132	92	.929	.000	.000	35	.899	.186	.000	16	.996	.195	.603	190	.967	.123	.230	100
PRON	.965	.464	.481	837	.991	.736	.842	1040	.958	.646	.696	468	.957	.624	.613	74	.993	.878	.888	1009	.988	.761	.781	283
PROPN	.941	.500	.348	226	.980	.805	.863	346	.852	.543	.596	144	.943	.245	.154	17	.977	.729	.844	239	.774	.571	.691	35
SCONJ	.846	.014	.048	268	.957	.066	.064	295	.954	.100	.564	315	.746	.000	.261	14	.959	.095	.767	139	.931	.171	.804	55
VERB	.944	.502	.507	1418	.969	.760	.713	1270	.925	.691	.703	986	.869	.411	.511	116	.971	.805	.917	1071	.942	.699	.771	311
X	.000	.000	.000	0	.968	.000	.046	84	.848	.040	.071	29	.000	.000	.000	0	.841	.078	.200	4	.711	.267	.250	3

Table 9: Per-tag F1 scores and corresponding support values for POS tagging with fine-tuned mBERT and generative models (Qwen, Apertus) in the **zero-shot** setting on **prose** test sets.