

Sentence Segmentation of Early New High German via Standard German Translation: Perplexity, Burstiness and Linguistic Analysis

Fritz Kliche
Akademie der Wissenschaften
zu Göttingen
fritz.kliche@adwgoe.de

Meryem Karalioglu
Herzog August Bibliothek
Wolfenbüttel
karalioglu@hab.de

Kevin Rößler
Herzog August Bibliothek
Wolfenbüttel
kroessler@hab.de

Abstract

We report on prompting-based sentence segmentation in a corpus of 16th century “Karlstadt” texts by using a translation into Standard German as an intermediate representation. We evaluate four prompting settings against a manually created gold standard. The results show that translation-based prompting yields better segmentation performance than prompting on the original texts alone, especially in terms of precision. At the same time, our analysis indicates that the task cannot be reduced to a simple correct-versus-incorrect distinction. In cases of disagreement, annotators and model outputs often produce alternative but internally coherent sentence spans and corpus-level boundary-distance patterns suggest differences in segmentation granularity and rhythmic scale. We discuss these differences in terms of perplexity and burstiness. We complement our study with a linguistic analysis showing that many of the model’s errors are connected with forms of ambiguity which often can be fixed using the Standard German translation.

1 Introduction

Sentence segmentation in Early New High German (ENHG) is difficult because punctuation, syntax, and semantic structure do not consistently converge on clear sentence boundaries. With the medium-term objective of an analysis of such texts, we address this problem in a collection of 16th century texts from the theologian and Protestant reformer Andreas Bodenstein von Karlstadt ([Kaufmann et al., 2017](#)).

Punctuation in these texts differs from modern usage. Instead of full stops and commas, “virgules” are frequently used, which are slashes separating coherent semantic units. The “periodus” resembles a modern-day full stop, but has a different meaning. Sentences can be extremely long, corresponding to the courtly language style of that time. Uppercase letters can be used at the beginning of clauses, and

do not necessarily indicate a sentence boundary. Some sentence boundaries are already indicated by editorial <.> markers in the source edition.

The objective of our work is to assess alternatives for introducing sentence segmentation in such texts by using a large language model and in-context learning, either directly in the historical texts or via a translation into contemporary German. We use a benchmarking approach for comparing four prompting strategies.

Splitting texts into sentences improves several downstream NLP tasks. E.g., sentences are an appropriate context for disambiguating reading variants of words or for identifying collocations.¹ Sentence segmentation is relevant for other historical language variants (e.g. [Li et al. \(2024\)](#) for Classical Chinese or [Bouma and Adesam \(2013\)](#) for Old Swedish), and it can support preprocessing of user-generated texts from comment sections, forums, etc., which often do not use standard punctuation.

Sentence segmentation and sentence punctuation often only achieve medium results. In a shared task on Classical Chinese, the highest reported F1 scores for both tasks were 88.47% and 75.29%, respectively ([Li et al., 2024](#)). Their setting was more complex than ours, as it jointly evaluated sentence boundaries and 11 punctuation types across four genres, and as the training data could not be varied.

The difficulty of the task is certainly due to different concepts of what a sentence is in old language variants. Accordingly, we do not generally assume that the insertion of sentence boundary markers is a binary decision which can be “correct” or “false”. Often, several interpretations are plausible, and we are aware that inserted boundary markers are modern editorial decisions. This is reflected in our annotation scheme for evaluating the AI-inserted sentence boundary markers against a gold standard which includes optional sentence boundaries.

¹This is planned in the scope of our project.

1.1 Perplexity

We take a look at the AI annotations from a broader perspective. We adapt two metrics which have previously been used for the detection of AI-generated texts for a stylistic comparison of the AI-based and human sentence segmentations. Texts can often be recognized as AI-written because AI tends to use more unsurprising, expected words when compared to humans. This can be measured with the “perplexity” score which is based on the probabilities with which words occur given their preceding context. Perplexity is often lower for AI texts than for human texts (Gehrmann et al., 2019; Gutiérrez Megías et al., 2024). We adapt this score for the evaluation of our sentence segmentation. As detailed later in this article, we use four prompts and there are three human annotators, resulting in seven strands of annotations. We compare the *distances* between inserted sentence boundary markers; in other words: we assess sentence lengths. We hypothesize that prompting-based sentence boundary markers are more evenly distributed within the annotated data than the human annotations, and, accordingly, have a lower perplexity.

The translations into Standard German change the number of tokens in the corpus. But we keep an exact alignment of the translated texts with the original lines in the edition. We decide to measure sentence lengths not in terms of tokens, but of the number of lines from a current sentence boundary to the previous boundary. Thus, we adapt token-level perplexity to a measurement of perplexity of sequence lengths.

We follow Jurafsky and Martin (2026, p. 9) and measure the perplexity of a set of sentence boundary distances D as given in equation 1, where each d_i indicates a number of lines. N is the number of all sentence boundary markers in a strand.

$$\text{perplexity}(D) = P(d_1, \dots, d_N)^{-\frac{1}{N}} \quad (1)$$

$P(d_1, \dots, d_N)$ is the product of the probabilities with which each distance d_i occurs in the strand.² The exponent $-\frac{1}{N}$ results in a normalization of the overall probability P to the range $[1, \infty]$. The negation “ $-$ ” in the exponent inverts the values: a high value for P results in a low perplexity score, a low P corresponds to a high perplexity.

² P is thus derived from the empirical distribution of sentence boundary distances within the corpus sample analyzed in our study. For example, for the strand of boundary distances 1-2-2-4-1-2, distance 2 has probability $p(2) = 3/6 = 0.5$.

1.2 Burstiness

We use a second measurement to take a distant view on the inserted sentence boundary markers. In our heterogeneous data, passages can differ quite strongly in their average sentence lengths. Some parts of the corpus consist of extremely long sentences, e.g., in letters to the court, while other sections mostly consist of short, consecutive sentences.

Such deviations from the average sentence length can be measured with the “burstiness score” (Qayyum et al., 2024, p. 5, and Goh and Barabási, 2008), which is distinct from perplexity. E.g., a strand of sentence lengths 2-2-2-2-14-14-2-2 has a low perplexity, because a sentence here spans either over 2 or over 14 lines, making another 14-line sentence little surprising. But this strand has a high burstiness, as the 14-line sentences strongly deviate from the average sentence length.

We measure the average sentence length μ of a strand, again by counting numbers of lines from a boundary marker to the previous marker. Next, we measure for each sentence its deviation from μ . σ is then the standard deviation from μ over all sentences (=the square root of the average squared deviations from μ). For normalizing this deviation in the range $[-1, 1]$, we finally use equation 2:

$$\text{burstiness}(D) = \frac{\sigma - \mu}{\sigma + \mu} \quad (2)$$

Burstiness has also been applied to detect AI-generated texts, using it to quantify how often words are repeated in short subsets of a corpus (Yadagiri et al., 2024). Our hypothesis is that in our setting, the lengths between AI-generated sentence boundary markers might yield a lower burstiness score than those of human annotations.

1.3 Research Questions

Our experiments are driven by the following research questions:

- (1) Does a translation into Standard German improve prompt-based sentence segmentation?
- (2) Which differences by means of perplexity, burstiness and linguistic features can we distinguish when the task is solved with or without an intermediate translation and when compared to human annotations?

2 Related Work

Sentence segmentation of historical or non-standard texts is an active field of study. E.g.,

MONAPipe (Dönicke et al., 2022) is a collection of NLP tools for German texts including a “Normalizer” for older German language variants, and a “Sentencizer”. Similarly, “Transnormer” is a collection of models for normalizing historical German texts (Bracke et al., 2024). Frohmann et al. (2024) train their system “Segment any Text” (SaT) on e.g. Tweets and forum texts which often lack reliable punctuation. Their goal is a model which is less dependent on punctuation marks as overt cues for sentence boundaries, for improving the model’s performance in contexts without such indicators.

Ye et al. (2025) compare translations of Large Reasoning Models (LRMs) and traditional LLMs. They find that LRMs achieve better results than traditional LLMs when tasks involve document-level coherence, but they perform worse in structurally rigid and terminology-intensive tasks. They interpret this finding as a tendency of LRMs to “prioritize semantic plausibility over strict adherence to domain-specific terminology”. This finding makes LRMs more promising than traditional LLMs for our approach which is based on a translation for finding larger coherent structures.

Liu et al. (2026) show that the insertion of sentence boundary markers improves LLM-based performances: as LLMs learn by inferring implicit structures from masses of written human language, explicit sentence markers are meaningful information. In their experiments, they vary and test the insertion of standard punctuation (such as “.”), common words (such as *and*), or structural tags (“<seg>”) as indicators for sentence boundaries, and find that the latter achieve the best performance in subsequent NLP tasks. We adapt this finding by also using structural tags, such as [SENTENCE].

Wang et al. (2024) fine-tune a generative model for sentence segmentation and punctuation of classical Chinese texts. They observe that AI output can slightly differ from the original texts. They use a character correction method to ensure that both the original and generated text versions are identical except for the inserted punctuation marks.

The approach of translating texts for better performance in NLP tasks has also been proven successful. E.g., both Etxaniz et al. (2024) and Huang et al. (2023) show that LLMs generate more precise answers for tasks such as common sense reasoning or mathematical reasoning when the LLM first translates its prompt into English. Both groups test this claim for 27 languages, including low-resource languages.

3 Corpus and Manual Annotations

The Karlstadt edition is available online.³ Scans of the 16th century printings and handwritings are given, along with a line-by-line aligned text version, such that readers can follow the original texts. The edition is provided in XML files conforming to the guidelines of the Text Encoding Initiative (TEI), which is a widely used format for digital editions and in the Digital Humanities in general. We extract line-aligned plain text from these files and apply sentence segmentation to it.

We retain the line-by-line alignment with the original texts, as it facilitates the mapping of the inserted sentence boundary markers onto the original TEI edition. We use layout information given in the TEI files, e.g. on paragraph and headline boundaries, as the ends of such elements mark in most (but not all) cases the end of a sentence. As a rule, we insert a sentence-end marker at the end of each layout block.

Our main task is the insertion of the additional sentence boundary markers in the paragraphs. Two annotators A1 and A2 manually labeled sentence boundaries in ~2600 lines (~21,000 tokens) of the Karlstadt edition. This sample from the corpus comprises 19 documents: 16 letters (1,032 lines), including a printed letter from Martin Luther to Karlstadt (71 lines), one announcement regarding the territorial affiliation of a village (34 lines), a manuscript containing a personal legal opinion of Karlstadt (43 lines), and an excerpt from one of Karlstadt’s theological treatises (1,249 lines). We compile a unified gold standard after a third annotator A3 decided on divergent annotations between A1 and A2. A1–A3 are native German speakers, two have specialized training in historical linguistics. We use the following labels.

- s: End of a layout block
- x: Sentence boundary within a layout block
- ; Context where a sentence boundary can either be assumed or rejected, e.g., positions followed by a conjunction such as *and* or adverbs such as *therefore* which can be interpreted as onset of a subordinate clause or as onset of an independent sentence
- : Karlstadt’s writing style suggests a colon in many contexts, especially for quoted biblical passages, e.g. “As Jeremiah says: ...”

³<https://karlstadt-edition.org>

The corpus contains 2,575 lines in total, including 2,429 text lines and 146 structure lines corresponding to layout blocks. Of the text lines, 946 received a gold-standard label (or, in rare cases, more than one), and 1,483 remained unlabeled.

Table 1 shows eight lines of our corpus, aligning the original ENHG text, the Standard German translation and, for illustration, an English translation. The following three columns contain the labels of A1 and A2, as well as the final gold standard label given by A3. Line 6 shows an example where both an interpretation as the end of a sentence or its continuation are plausible; the reader might even find more such examples in the texts given in the table.

Our annotations are publicly available at <https://github.com/karalioglu/konvens2026-enhg-segmentation>

The vagueness of the task is reflected in differences between the two annotators. A1 more likely treats a text passage as a continuous sentence when it has a coherent meaning, even if two sentences could be assumed in grammatical terms, and uses label “;” more often. A2 prioritizes grammatical structures over semantic decisions.

We measure $\kappa=0.871$ as inter-annotator agreement between A1 and A2 for the labels [sx;:], achieved after a revision of the first annotation guidelines. Most interesting for our experiments are presumably labels “x” and “;” which are the labels we see as sentence boundaries, but excluding the layout-based label “s”. Only considering [x;], we achieve $\kappa=0.861$.

4 Prompt-based Annotations

Translation-based sentence segmentation involves both understanding of the texts and alignment with the original text structure which motivates our use of a reasoning model. We use the GPT 5.5 Thinking model⁴. The experiments were conducted via the ChatGPT interface in June 2026; no explicit reasoning-effort setting was selected and the temperature could not be configured. The model processed the corpus in chunks of approximately 500 lines per call, without splitting layout blocks across chunks. The prompts are given in the appendix.

Currently, generative AI often does not perform sufficiently for many text classification tasks, and “scores decrease rapidly as the number of labels increases” (Leitner and Rehm, 2025, p. 6). Accordingly, we do not prompt the model to insert

the full range of our manual labels, but only the tag [GPT-SENTENCE] for sentence boundaries and [GPT-SENTENCE?]⁵ in contexts when a sentence boundary is likely. Note that both the LLM-based and manual annotations include optional markers: [GPT-SENTENCE?] and [;]. Additionally, the texts include the layout-based marker [SENTENCE] indicating ends of layout elements, corresponding to the label “s” in the manual annotations.

4.1 First Run Prompts

We first test with two prompts. (1) The task described with the first prompt is the insertion of [GPT-SENTENCE] and [GPT-SENTENCE?] tags into the ENHG texts based on modern punctuation rules, which we refer to as “insert in ENHG”. (2) Second, we use a prompt “insert in translation” to translate the texts into Standard German and to insert the tags then, while preservation of the line-by-line alignment has the highest priority.

4.2 Second Run Prompts

While the above two prompts lead to results with high precision scores (see below, sec. 5), recall may be further improved with two additional prompts, where we give both line-by-line-aligned results from prompts 1 and 2 as input to the model, i.e., the original ENHG texts and, tab-separated, Standard German translations, both including the inserted tags. The additional prompts (3) and (4) are as follows: (3) The model is prompted to add [GPT2] into the ENHG texts while considering the tags in the Standard German translations. We call this prompt “improve in ENHG”. (4) Vice versa, we instruct the model to include [GPT2] into the Standard German texts, by considering the aligned ENHG texts (“improve in translation”). Both prompts do not allow the model to remove or alter existing tags.

5 Results and Discussion

5.1 Aggregate Results

We evaluate the results of our experiments using different variants of tags and labels. Table 2 shows interesting results.

The prompt used in each evaluation variant (=a line in the table) is given in the first column.

In column 2, the GPT tags considered in the respective evaluation variant are given. First, and

⁴<https://openai.com/index/introducing-gpt-5-5>

⁵We used the German tags [SATZ], [GPT-SATZ] and [GPT-SATZ?].

	Early New High German	Standard German	English translation	A1	A2	A3
1	Ich bewege das/ das ordentlich und gemeinen	Ich erwäge das, was ordentlich und gemeinen	I consider that which is proper and in accordance with common			
2	gerechten gemes ist/ das verpindten	Rechten gemäß ist, dass Binden	law, namely that binding			
3	und endtpindten/ excommuniceren	und Entbinden, Exkommunizieren	and unbinding, excommunicating			
4	und absolviren/ Erkennen	und Absolvieren, Erkennen	and absolving, recognizing			
5	und urteilen/ gleich eyner person	und Urteilen einer Person	and judging are equally due to one person,			
6	geburett [GPT-SATZ?] und wer eyneß mechtig	gleich gebühren, und wer eines mächtig	[GPT-SENTENCE?] and whoever has authority over one	;	x	;
7	ist/ der hait das andere zethun	ist, der hat das andere zu tun,	also has the other to perform,			
8	also haben gemein rechten . [SATZ]	also haben gemeine Rechte. [SATZ]	as is held by common law. [SENTENCE]	s	s	s

Table 1: Example of one coherent layout block from the Karlstadt corpus with aligned translations and sentence-boundary markers. [GPT-SATZ?] is model-generated, [SATZ] is a layout-based sentence boundary marker at the end of the block. The English translation is included for illustration only.

Prompt	Tags	Labels	Precision	Recall	F1	TP	#
Insert in ENHG	[SENT]+[GPT]	sx	0.794	0.922	0.853	735	797
Insert in translation	[SENT]+[GPT]	sx	0.930	0.917	0.924	731	797
(1) Insert in ENHG	[SENT]+[GPT]	sx;	0.828	0.887	0.857	767	865
(2) Insert in translation	[SENT]+[GPT]	sx;	0.966	0.877	0.919	759	865
(3) Improve in ENHG	[SENT]+[GPT]	sx;	0.831	0.941	0.883	814	865
	+ [GPT2]						
(4) Improve in translation	[SENT]+[GPT]	sx;	0.839	0.945	0.889	817	865
	+ [GPT2]						
Insert in translation: only GPT tags	[GPT]	x;	0.921	0.846	0.882	593	701

Table 2: Precision, recall and F1 score for different evaluation variants.

unexpectedly, distinguishing [GPT-SENTENCE] from [GPT-SENTENCE?] never improved any evaluation variant. Especially, we did not achieve convincing results when equating [GPT-SENTENCE?] with the gold standard “optional-ity” label “;”. We decided to combine [GPT-SENTENCE] and [GPT-SENTENCE?] in Table 2 and refer to both as [GPT]. Second, column 2 shows that prompts (3) and (4) (cf. section 4.2) contain the additional tag [GPT2].

For the translation-based prompts, the sentence-boundary tags inserted in the Standard German text were reliably mapped back onto the corresponding positions in the ENHG text using a separate prompt. We do not discuss these mapping results further in this paper.

The gold standard distinguishes the four labels [sx;:]. For our evaluation variants, we take different

subsets from these labels, and evaluate the GPT results against the subsets [sx], [sx;] and [x;]. The labels used for an evaluation variant are given in column 3. Variants containing [:] are not given in the table as it turned out that these subsets did not improve any results.

For comparison of our approach with other methods, the last line in the table shows the results when only the GPT-generated tags [GPT] are evaluated against [x;], skipping the layout-based tag [SENTENCE].

The next columns give precision, recall and F1 for the respective evaluation variant. Column 7 “TP” lists the total number of True Positives of GPT’s tag insertions, and column 8 “#” is the number of labels in the gold standard for the subset of labels used in the respective evaluation variant ([sx], [sx;] or [x;]).

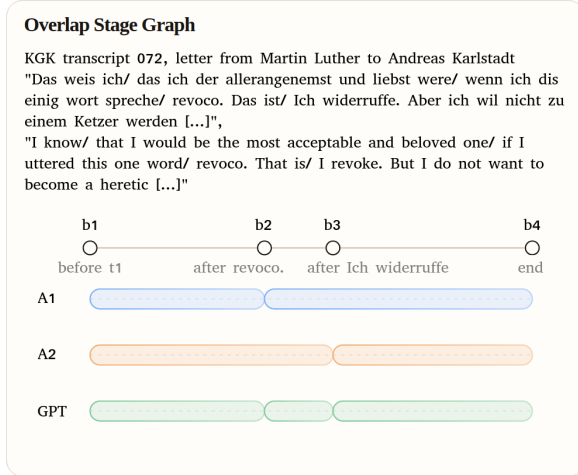


Figure 1: One overlapping disagreement case rendered as an overlap stage graph. The same local boundary points are present in all three annotations, but they result in different sentence divisions.

Results we find noticeable are highlighted in boldface. For the first run, it is interesting that the prompt “insert in translation” reaches a high precision (0.930), although “insert in ENHG” has slightly better results for recall (0.922).

The results for both prompts of the additional second run (prompts 3 and 4) score high for recall (0.941 and 0.945), while precision is lower than for the prompt “insert in translation”.

We additionally ran MONAPipe (Dönicke et al., 2022) without fine-tuning and post-processing. We see this experiment only as a reference rather than a baseline for our system. MONAPipe yielded precision=0.712, recall=0.622 and F1=0.664 when compared to the labels [sx;].

5.2 Annotation Patterns beyond Aggregate Scores: Overlap and Regularity

Beyond aggregate precision and recall, the annotation data show that sentence segmentation in this material cannot be reduced to a simple correct-versus-incorrect distinction. In a number of passages, the two human annotators (A1, A2) and the prompts assign different but internally coherent sentence spans to the same textual sequence. This is not only a matter of isolated boundary errors. Rather, the same boundary candidates can be grouped into larger or smaller sentence units, so that disagreement often takes the form of competing segmentations of one shared local structure.

Figure 1 illustrates this point with a compact disagreement example. The analytical relevance of such cases lies in the fact that disagreement is struc-

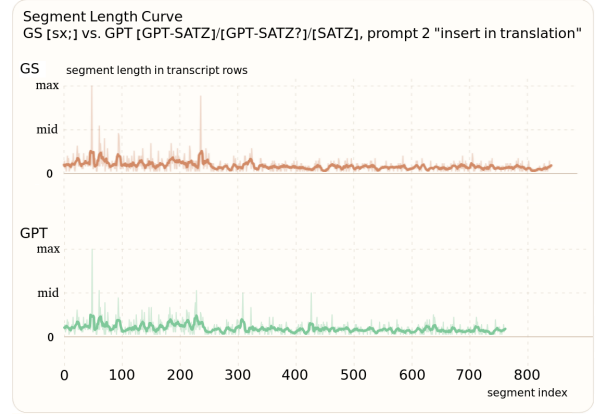


Figure 2: Sequence of distances between successive sentence (“segment”) boundaries [sx;] in the gold-standard column (GS) compared with boundary markers of GPT (prompt 2). GPT sentences are less densely distributed and have a larger variety than GS sentences.

tured rather than arbitrary. The annotators do not simply place unrelated boundaries; instead, they frequently share some local anchor points while differing in how broadly or narrowly they close a sentence. This suggests that part of the task’s difficulty lies in the interaction between local cues and larger interpretive decisions about sentence scope.

At the corpus level, a similar picture emerges when sentence boundaries from the gold standard (GS) are compared with GPT boundaries for prompt 2 “insert in translation”. The resulting segment-length curves are given in Figure 2. The graphs show that GS contains more segments and therefore extends farther along the segment-index axis. The average distance between successive GS boundaries is shorter, whereas GPT leaves longer intervals between sentence boundaries. The GPT strand shows a quite large variety, and the curve does not indicate a generally balanced distribution.

This pattern supports a cautious interpretation in which GS and GPT do not operate at the same segmentation scale. If one compares only GS boundaries with GPT sentence boundaries, the GS stream appears rhythmically denser and more fine-grained, while GPT holds larger units together before placing a boundary. The relevant contrast is therefore less one of regular versus irregular annotation than one of coarse versus fine-grained segmentation rhythm. One might initially expect an AI model to annotate in a more regular way, perhaps even in an almost harmonic way, but the present data do not support such a simple contrast. The data do not show that GPT fluctuates less than GS. In

absolute terms, the GPT sequence even has the higher dispersion, so a stronger harmonic regularity cannot be inferred from mean segment length alone. We argue that sentence segmentation should not be reduced prematurely to a single normalized output, since competing boundary structures may carry information that is relevant to downstream lexical, editorial and semantic analysis. We therefore examine these differences more closely in the following section.

5.3 Results for Perplexity and Burstiness

We measure perplexity and burstiness in our seven annotation strands of four prompts and three human annotators. The results are given in Table 3, along with the number of annotations of the respective strand in the last column. For prompts 1, 3 and 4, the AI annotation strands have a lower perplexity than human annotators A1 and A2, which is consistent with our initial hypothesis, but prompt 2 yields a higher perplexity. Perplexity is strongly associated with the number of boundaries: prompt 2 with its high perplexity has the fewest boundary markers, whereas prompts 3 and 4 have the most boundary markers and the lowest perplexity scores. With fewer boundary markers, longer segments seem to remain intact and a broader range of segment lengths can occur, which contributes to a higher perplexity.

Also recall that prompt 2 “insert in translation” had the highest precision score in Table 2. At least within our four AI annotation strands, both the results for perplexity and precision suggest that denser AI segmentation is associated with a more regular boundary rhythm, indicating a smoothing effect that comes at the cost of missing unusual boundary patterns.

Interestingly, prompt 2 also has the lowest burstiness given in the table (-0.118). Both values combined show that this prompt (1) uses heterogeneous patterns for sentence lengths (=perplexity), (2) but these lengths do not deviate strongly from the average sentence length (=burstiness). All four prompts yield lower burstiness scores than all three human annotators, consistent with our hypothesis.

This dichotomy of high perplexity and low burstiness calls for an explanation. We interpret these findings as follows. The translation into Standard German generates diversified texts. But on the other hand, the Standard German translations (given the low recall and low burstiness) do not find outliers, spurious hits and unusual sentence bound-

aries. This has benefits, e.g. the high precision, but also drawbacks, such as the low recall.

Generally, perplexity for the “second run” of our experiments (“improve in ENHG” and “improve in translation”) is lower than for the first run (“insert in ENHG” and “insert in translation”). The second run seems to increase recall at the cost of a less natural language.

The perplexity scores also show differences between the three human annotators. A3 decided on the final gold standard annotations in cases where A1 and A2 disagreed, and yields a lower perplexity. We argue that this is similar to the aforementioned finding for the second run of the AI annotations: When an annotation strand is supposed to combine two previously existing annotation strands, there is a tendency to harmonize them, resulting in a lower perplexity.

6 Qualitative Analysis

By comparing the GPT-suggested sentence boundaries with the unified gold standard data in the ENHG text, we find interesting diverging cases between the first run (prompts “insert in ENHG” and “insert in translation”) and the two prompts of the second run. Although some of these differences can be seen as actual mistakes, most of them seem to lack grammatical and semantic justification and require another explanation. We divide our discussion into the two previously introduced categories of “False Positives” (FP), where the model places a sentence boundary not determined by the gold standard, and “False Negatives” (FN), where the gold standard determines a sentence boundary that GPT does not confirm. E.g., prompt 4 yields more False Positives than prompt 2 but fewer False Negatives.

Analyzing the sentence structure of Karlstadt’s writing in these cases, some patterns can be distinguished that may be used to determine the issues the model encounters when working with ENHG texts. The interpretations are speculative, however, since insights into the decision process of GPT and its familiarity with the ENHG language in general are beyond the scope of this paper.

By far the most prominent case of False Positive assignment concerns Karlstadt’s use of coordinating conjunctions. As words like *und* (*and*) or *denn* (*because*) in Standard German introduce another main clause that may or may not be linked to the first, in a lot of instances a case can be made for both the placement and the absence of punc-

Annotation	Perplexity	Burstiness	Labels
(1) Insert in ENHG	6.492	-0.098	926
(2) Insert in translation	7.284	-0.118	786
(3) Improve in ENHG	6.114	-0.100	979
(4) Improve in translation	6.105	-0.102	974
Human annotator 1	6.887	-0.067	822
Human annotator 2	6.871	-0.082	826
Human annotator 3	6.468	-0.062	865

Table 3: Perplexity and burstiness of the annotation variants, using the label subset [sx;].

tuation of almost any kind. The high number of False Positive cases containing ENHG words such as *und* or *dan* (*then*) is not surprising. However, in most of these occurrences, the output of GPT does not only deviate from all three human annotators, but also from its own output of run 2. This fact corroborates the idea that GPT struggles in recognizing ENHG grammatical connectors as a whole, like conjunctions (co- or subordinating), relative pronouns, or pronominal adverbs. For example, as Karlstadt does not use the modern spelling *dass* but writes *das* instead, a lot of cases suggest that GPT often fails to distinguish the homographic terms *das* (relative pronoun or article), from *das* (conjunction), thus entirely changing the structure of the sentences.

A few of the aforementioned occurrences may also be explained not by the model’s insufficient understanding of the texts, but by Karlstadt’s style of writing. For instance, he seems to capitalize connectors in arbitrary cases, leading GPT to determine them as the beginning of a new sentence. Similarly, GPT inserts sentence boundary markers when Karlstadt uses the ENHG “periodus” that, while being indistinguishable from a full stop, can also be translated as a colon and is sometimes even placed for emphasis and little else. Both of these cases in particular are not or barely present in the results of prompt 2, meaning that integrating the translation to Standard German seems to largely fix these issues. This is very likely the result of the LLM sourcing from the larger datapool of Standard German, and of forcing the model to understand the ENHG texts as a prerequisite of the translation.

In general, the False Positive and False Negative results change at a certain point in the annotated corpus. This is, of course, the direct result of Karlstadt’s style of writing in this second part. With that part being a theological text which is much more paratactic than the texts of the first part of the

corpus (mostly letters), the segmentations of the gold standard become much more frequent. That is, however, not the case for the GPT annotations. Going through the results from top to bottom, it is evident that the False Positive assignments dominate during the first part, but are vastly outweighed by False Negatives in the second part. This suggests that GPT does not sufficiently adapt its segmentation behavior to the change in textual structure and writing style, and less clearly than the human annotators (cf. again Figure 2).

Most categories we discussed in the qualitative analysis include a form of ambiguity which can only be specified with a coherent understanding of the larger context: *das* vs. *das(s)*; coordinate conjunctions such as *und* connecting sentences vs. constituents of a phrase; the modern full stop vs. the ENHG “periodus”. Similarly, the change in the writing style in the corpus shows a certain non-uniformity.

This paper is concerned with the performance of generative AI when working on historical sentence segmentation. The findings of the linguistic analysis highlight the general difficulties of the model, and serve readers as food for thought in similar projects.

7 Conclusion

We worked on prompt-based sentence segmentation of Early New High German (ENHG) texts using Standard German as an intermediate representation.

Translating the texts into Standard German indeed achieved better results for precision compared to prompting the model to insert sentence boundary tags in the ENHG texts directly.

Using a second run and prompting the model to consider the tags inserted by both previous prompts (“insert in ENHG” and “insert in translation”) improved recall, but lowered precision. This two-step

approach can be used to generate sentence boundary candidates which would require a subsequent refinement by e.g. candidate classification.

We do not stick to an evaluation in terms of true or wrong annotations. Instead, we argue that in our historic text corpus sentence boundaries vary between AI and human annotators due to different conceptions of what a sentence is. Accordingly, we include perplexity and burstiness as measurements for discussing stylistic variants. The prompt-based translation into Standard German leads to high precision and perplexity scores, but lower results for recall and burstiness. We interpret this finding by stating that this approach of using Standard German as a “pivot language” generates valuable and readable texts, but it misses several sentence boundaries which are outliers, spurious hits and unusual constructions.

In a qualitative linguistic analysis, we find that most of the model’s False Positives and False Negatives include a form of ambiguity. They include coordinate conjunctions such as *und* and the distinction of *dass* and *das*, both written as *das* by Karlstadt. Importantly, several of these problems are largely reduced in the translation-based setting, suggesting that Standard German helps the model resolve ENHG-specific structural ambiguities.

We detect the model’s problems in recognizing a change of the writing style in the corpus, resulting in frequent False Positives in the first part and a high number of False Negatives after this change, showing that the model apparently tries to balance its annotations to a certain extent.

Our research design entails several limitations. We only evaluate one closed model and focus our comparisons on differences between prompting strategies. Future work should test whether these effects generalize across models. As we do not perform repeated runs in combination with significance testing, small differences between prompting variants should be interpreted carefully.

In future work, we can also examine whether different reasoning settings affect the segmentation. Likewise, it would be interesting to inspect the reasoning processes of the models during the translation and during the insertion of sentence boundary tags. An analysis of the translations’ quality would also be valuable, as the measurements in our current setting (perplexity and burstiness) are solely based on sentence lengths.

Acknowledgments

This research was conducted as part of the project EDIKILEX, which is funded by the Ministry of Science and the Arts of the German federal state of Lower Saxony.

References

- Gerlof Bouma and Yvonne Adesam. 2013. Experiments on sentence segmentation in Old Swedish editions. In *NEALT Proceedings Series*, volume 18, pages 11–26.
- Yannic Bracke, Gregor Middell, and Alexander Geyken. 2024. [Transnormer: Ein Transformer-basierter Ansatz für die Normalisierung historischer Texte](#). Poster presented at the 3rd Text+ Plenary, Mannheim, Germany. October 9–11, 2024.
- Tillmann Dönicke, Florian Barth, Hanna Varachkina, and Caroline Sporleder. 2022. [MONAPipe: Modes of narration and attribution pipeline for German computational literary studies and language analysis in spaCy](#). In *Proceedings of KONVENS 2022*, Potsdam, Germany.
- Julen Etxaniz, Gorka Azkune, Aitor Soroa, Oier Lopez de Lacalle, and Mikel Artetxe. 2024. [Do multilingual language models think better in English?](#) In *Proceedings of NAACL*, Mexico City, Mexico.
- Markus Frohmann, Igor Sterner, Ivan Vulić, Benjamin Minixhofer, and Markus Schedl. 2024. [Segment any Text: A universal approach for robust, efficient and adaptable sentence segmentation](#). In *Proceedings of EMNLP*.
- Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. [GLTR: Statistical detection and visualization of generated text](#). In *Proceedings of ACL: System Demonstrations*, Florence, Italy.
- Kwang-Il Goh and Albert-László Barabási. 2008. [Burstiness and memory in complex systems](#). *Europhysics Letters*, 81(4):48002.
- Alberto José Gutiérrez Megías, L. Alfonso Ureña-López, and Eugenio Martínez Cámara. 2024. [The influence of the perplexity score in the detection of machine-generated texts](#). In *Proceedings of NLPAICS 2024*, Lancaster, UK.
- Haoyang Huang, Tianyi Tang, Dongdong Zhang, Wayne Xin Zhao, Ting Song, Yan Xia, and Furu Wei. 2023. [Not all languages are created equal in LLMs: Improving multilingual capability by cross-lingual-thought prompting](#). *Preprint*, arXiv:2305.07004.
- Daniel Jurafsky and James H. Martin. 2026. [Speech and language processing](#). Draft of January 6, 2026.

- Thomas Kaufmann, Harald Bollbuck, Ulrich Bubenheimer, Martin Kessler, Stefania Salvadori, Christian Speer, and Alejandro Zorzin. 2017. *Kritische Gesamtausgabe der Schriften und Briefe Andreas Bodensteins von Karlstadt*. Gütersloher Verlagshaus, Gütersloh.
- Elena Leitner and Georg Rehm. 2025. [Exploring the limits of LLMs for German text classification: Prompting and fine-tuning strategies across small and medium-sized datasets](#). *Journal for Language Technology and Computational Linguistics*, 38(2):1–12.
- Bin Li, Bolin Chang, Zhixing Xu, Minxuan Feng, Chao Xu, Weiguang Qu, Si Shen, and Dongbo Wang. 2024. [Overview of EvaHan2024: The first international evaluation on Ancient Chinese sentence segmentation and punctuation](#). In *Proceedings of the Third Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA) at LREC-COLING-2024*, Turin, Italy.
- Zhichen Liu, Yongyuan Li, and Yang Xu. 2026. [Think in sentences: Explicit sentence boundaries enhance language model’s capabilities](#). *Preprint*, arXiv:2604.10135.
- Hina Qayyum, Muhammad Ikram, Benjamin Zi Hao Zhao, Ian D. Wood, Nicolas Kourtellis, and Mohamed Ali Kaafar. 2024. [Exploring the distinctive tweeting patterns of toxic Twitter users](#). *Preprint*, arXiv:2401.14141.
- Shiquan Wang, Weiwei Fu, Mengxiang Li, Zhongjiang He, Yongxiang Li, Ruiyu Fang, Li Guan, and Shuangyong Song. 2024. [Sentence segmentation and punctuation for ancient books based on supervised in-context training](#). In *Proceedings of the Third Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA) at LREC-COLING-2024*, Turin, Italy.
- Annepaka Yadagiri, Lavanya Shree, Suraiya Parween, Anushka Raj, Shreya Maurya, and Partha Pakray. 2024. [Detecting AI-generated text with pre-trained models using linguistic features](#). In *Proceedings of ICON*, Chennai, India.
- Yongshi Ye, Biao Fu, Chongxuan Huang, Yidong Chen, and Xiaodong Shi. 2025. [How well do large reasoning models translate? A comprehensive evaluation for multi-domain machine translation](#). *Preprint*, arXiv:2505.19987.

A Appendix

A.1 Prompt for »Insert in Original«

Du bist ein philologisch geschulter NLP-Assistent für frühneuhochdeutsche Editionstexte.

Aufgabe:

Segmentiere den folgenden frühneuhochdeutschen Text in Sätze, indem du Satzende-Kandidaten mit [GPT-SATZ] oder [GPT-SATZ?] markierst.

Bereits vorhandene Strukturmarker:

- Zeilen mit #####ID: sind Strukturmarker. Behalte sie unverändert bei; sie gehören nicht zum Text.
- <empty> markiert technische Leerstellen. Behalte <empty> unverändert bei; es ist keine Satzgrenze.
- Setze keine Satzgrenze über eine #####ID:-Grenze hinweg.
- Vorhandene [SATZ]-Marker markieren sichere Satzenden. Sie bleiben unverändert erhalten.

Marker:

- [GPT-SATZ] = klare oder sehr plausible Satzgrenze.
- [GPT-SATZ?] = unsichere, aber plausible Satzgrenze.
- Ziel ist hoher Recall: Markiere auch unsichere, aber plausible Satzende-Kandidaten.

Beispiel:

Ohne GPT-Marker:

#####ID: 1

Durchleuchtichster hochgebornner Churfurst Eweren Churfurstlichen gnaden
sindt unßer gebet gegen got/ und <empty> underthenige dinst in vleis
alzzeit zcu vor(.) Gnedigster herre wir fugen Eweren Churfurstlichen gnaden
undertheniglich wissen/ wie der Schosser den doctor Andream
umb eynen jherlichen hauszcinß fur uns verklagt
und wir darauff eyn urteil gesprochen/ do von sich doctor Andreas
an Bebstliche Heilikeit beruffen hat/ als wirs achten unbillich

Mit GPT-Markern:

#####ID: 1

Durchleuchtichster hochgebornner Churfurst Eweren Churfurstlichen gnaden
sindt unßer gebet gegen got/ und <empty> underthenige dinst in vleis
alzzeit zcu vor(.) [GPT-SATZ] Gnedigster herre wir fugen Eweren Churfurstlichen gnaden
undertheniglich wissen/ wie der Schosser den doctor Andream
umb eynen jherlichen hauszcinß fur uns verklagt
und wir darauff eyn urteil gesprochen/ [GPT-SATZ?] do von sich doctor Andreas
an Bebstliche Heilikeit beruffen hat/ als wirs achten unbillich

Arbeite so:

- Lies jeden Chunk als zusammenhängenden historischen Text.
- Übertrage den Text gedanklich in moderne Interpunktion.

- Nutze diese Normalisierung, um Satzgrenzen zu erkennen und [GPT-SATZ]/[GPT-SATZ?]-Tags zu markieren.
- Verwende häufig den Marker [GPT-SATZ?], um viele Satzende-Kandidaten zu finden.
- Behandle verschiedene Chunks als getrennte Einheiten.
- Setze keine Satzgrenze über eine #####ID:-Grenze hinweg.
- Ignoriere leere Zeilen bei der Satzanalyse; sie sind keine Satzgrenzen.

Wichtig:

- Verändere keine Wörter, keine Schreibungen, keine vorhandenen Zeichen und keine Zeilenumbrüche.
- Füge ausschließlich die Marker [GPT-SATZ] und [GPT-SATZ?] ein.
- Lasse vorhandene [SATZ]-Marker unverändert stehen.
- Ein Punkt ist kein automatisches Satzende.
- Eine Virgel "/" ist kein automatisches Satzende.
- Zeilenumbrüche sind keine Satzgrenzen.
- Leere Zeilen sind keine Satzgrenzen.
- Setze [GPT-SATZ]/[GPT-SATZ?] nur bei syntaktisch und semantisch abgeschlossenen Einheiten.
- Setze [GPT-SATZ]/[GPT-SATZ?] niemals mitten in ein Wort.
- Ziel ist hoher Recall: Markiere auch unsichere, aber plausible Satzende-Kandidaten mit [GPT-SATZ?].
- Füge keine zusätzlichen Zeilenumbrüche ein.
- Entferne keine Zeilenumbrüche.
- Breche lange Zeilen nicht um.

Ausgabe:

Gib nur den bearbeiteten Dateiinhalt zurück, ohne Erklärung, Kommentare oder zusätzliche Überschriften.

A.2 Prompt for »Insert in Translation«

Du bist ein philologisch geschulter NLP-Assistent für frühneuhochdeutsche Editionstexte.

Aufgabe:

Übertrage den folgenden frühneuhochdeutschen Text zeilengetreu in heutiges Standarddeutsch und markiere Satzende-Kandidaten mit [GPT-SATZ] oder [GPT-SATZ?].

Bereits vorhandene Strukturmarker:

- Zeilen mit #####ID: sind Strukturmarker. Behalte sie unverändert bei; sie gehören nicht zum Text.
- <empty> markiert technische Leerstellen. Behalte <empty> unverändert bei; es ist keine Satzgrenze.
- Setze keine Satzgrenze über eine #####ID:-Grenze hinweg.
- Vorhandene [SATZ]-Marker markieren sichere Satzenden. Sie bleiben unverändert erhalten.

Marker:

- [GPT-SATZ] = klare oder sehr plausible Satzgrenze.
- [GPT-SATZ?] = unsichere, aber plausible Satzgrenze.
- Ziel ist hoher Recall: Markiere auch unsichere, aber plausible Satzende-Kandidaten.

Beispiel:

Ohne GPT-Marker:

#####ID: 1

Durchleuchtichster hochgebornner Churfurst Eweren Churfurstlichen gnaden
sindt unßer gebet gegen got/ und <empty> underthenige dinst in vleis
alzzeit zcu vor(.>) Gnedigster herre wir fugen Eweren Churfurstlichen gnaden
undertheniglich wissen/ wie der Schosser den doctor Andream
umb eynen jherlichen hauszcinß fur uns verklagt
und wir darauff eyn urteil gesprochen/ do von sich doctor Andreas
an Bebstliche Heilikeit beruffen hat/ als wirs achten unbillich

Mit Standarddeutsch und GPT-Markern:

#####ID: 1

Durchlauchtigster, hochgeborener Kurfürst, Euren kurfürstlichen Gnaden
seien unser Gebet zu Gott und untertänige Dienste mit Fleiß
allezeit zuvor bereit. [GPT-SATZ] Gnädigster Herr, wir fügen Euren kurfürstlichen Gnaden
untertänig zu wissen, wie der Schösser den Doktor Andreas
wegen eines jährlichen Hauszinses vor uns verklagt hat
und wir darauf ein Urteil gesprochen haben; [GPT-SATZ?] davon hat sich Doktor Andreas
an päpstliche Heiligkeit berufen, wie wir meinen, zu Unrecht.

Wichtig:

- Behalte die Zeilenstruktur exakt bei.
- Die Ausgabe muss exakt dieselbe Anzahl Zeilen haben wie der Input.
- Verschiebe keine Inhalte zwischen Zeilen.

- Wenn eine Input-Zeile mitten im Satz endet, muss auch die entsprechende Output-Zeile mitten im Satz enden.
- Füge außer der Standarddeutsch-Übertragung ausschließlich [GPT-SATZ] und [GPT-SATZ?] hinzu.
- Ein Punkt, eine Virgel "/" und ein Zeilenumbruch sind keine automatischen Satzgrenzen.
- Setze [GPT-SATZ]/[GPT-SATZ?] niemals mitten in ein Wort.
- Es ist besser, einen plausiblen Kandidaten mit [GPT-SATZ?] zu viel zu setzen als einen möglichen Kandidaten auszulassen.

Ausgabe:

Gib nur den bearbeiteten Dateiinhalt zurück, ohne Erklärung, Kommentare oder zusätzliche Überschriften.

A.3 Prompt for »Improve in Original«

Du bist ein philologisch geschulter NLP-Assistent für frühneuhochdeutsche Editionstexte.

Aufgabe:

Du erhältst eine TSV-Datei mit zwei Spalten:

- **Original**
Frühneuhochdeutscher Originaltext mit bereits vorhandenen Satzmarkern.
- **Standarddeutsche Übersetzung**
Zeilentreue standarddeutsche Übersetzung desselben Textes, ebenfalls mit Satzmarkern.

Verbessere ausschließlich die Satzgrenzen im frühneuhochdeutschen Originaltext in Spalte 1.

Dabei sollst du nicht die bereits vorhandenen Satzmarker verändern, sondern nur zusätzliche mögliche Satzgrenzen als [GPT2-SATZ] markieren.

Bereits vorhandene Strukturmarker:

- Zeilen mit #####ID: sind Strukturmarker. Behalte sie unverändert bei; sie gehören nicht zum Text.
- <empty> markiert technische Leerstellen. Behalte <empty> unverändert bei; es ist keine Satzgrenze.
- Setze keine Satzgrenze über eine #####ID:-Grenze hinweg.
- Vorhandene [SATZ]-, [GPT-SATZ]- und [GPT-SATZ?]-Marker markieren bereits erkannte Satzende-Kandidaten. Sie bleiben unverändert erhalten.

Marker:

- Vorhandene Satzmarker sind [SATZ], [GPT-SATZ] und [GPT-SATZ?].
- Die vorhandenen Satzmarker dürfen nicht verändert, entfernt, verschoben oder ersetzt werden.
- Neu hinzugefügt werden darf ausschließlich [GPT2-SATZ].
- [GPT2-SATZ] kennzeichnet eine im zweiten Durchgang zusätzlich vorgeschlagene Satzgrenze.

Arbeite so:

- Nutze Spalte 2 als semantische und syntaktische Entscheidungshilfe.
- Prüfe, ob die standarddeutsche Übersetzung in Spalte 2 zusätzliche Satzgrenzen nahelegt, die in Spalte 1 noch nicht markiert sind.

- Ziel ist hoher Recall. Achte besonders auf mögliche bisher übersehene Satzgrenzen und setze im Zweifelsfall einen Marker.
- Setze [GPT2-SATZ] jedoch nur, wenn die Grenzstelle im frühneuhochdeutschen Original eindeutig bestimmbar ist.
- Markiere zusätzliche Satzgrenzen im Original mit [GPT2-SATZ].
- Setze keine Satzgrenze über eine #####ID:-Grenze hinweg.
- Ignoriere leere Zeilen bei der Satzanalyse; sie sind keine Satzgrenzen.

Wichtig:

- Die Ausgabe muss exakt dieselbe Anzahl Zeilen haben wie Spalte 1 im Input.
- Füge keine zusätzlichen Zeilenumbrüche ein.
- Entferne keine Zeilenumbrüche.
- Gib ausschließlich die verbesserte Spalte 1 zurück.
- Verändere keine Wörter, keine Schreibungen, keine vorhandenen Zeichen und keine Zeilenumbrüche im Original.
- Füge ausschließlich den Marker [GPT2-SATZ] hinzu.
- Entferne keine vorhandenen [SATZ]-, [GPT-SATZ]- oder [GPT-SATZ?]-Marker.
- Verschiebe keine vorhandenen Marker.
- Ersetze keine vorhandenen Marker durch [GPT2-SATZ].
- Ein Punkt ist kein automatisches Satzende.
- Eine Virgel "/" ist kein automatisches Satzende.
- Zeilenumbrüche sind keine Satzgrenzen.
- Leere Zeilen sind keine Satzgrenzen.
- Setze [GPT2-SATZ] nur bei syntaktisch und semantisch plausiblen Grenzen.
- Setze [GPT2-SATZ] niemals mitten in ein Wort.
- Verändere Spalte 2 nicht und gib sie nicht aus.
- Setze [GPT2-SATZ] nicht direkt neben einen bereits vorhandenen Satzmarker.

Ausgabe:

Gib nur den bearbeiteten Inhalt von Spalte 1 zurück, ohne Erklärung, Kommentare oder zusätzliche Überschriften.

A.4 Prompt for »Improve in Translation«

Du bist ein philologisch geschulter NLP-Assistent für frühneuhochdeutsche Editionstexte.

Aufgabe:

Du erhältst eine TSV-Datei mit zwei Spalten:

- **Standarddeutsche Übersetzung**
Zeilengetreue standarddeutsche Übersetzung mit bereits vorhandenen Satzmarkern.
- **Original**
Frühneuhochdeutscher Originaltext desselben Textes, ebenfalls mit Satzmarkern.

Verbessere ausschließlich die Satzgrenzen in der standarddeutschen Übersetzung in Spalte 1. Dabei sollst du nicht die bereits vorhandenen Satzmarker verändern, sondern nur zusätzliche mögliche Satzgrenzen als [GPT2-SATZ] markieren.

Bereits vorhandene Strukturmarker:

- Zeilen mit #####ID: sind Strukturmarker. Behalte sie unverändert bei; sie gehören nicht zum Text.
- <empty> markiert technische Leerstellen. Behalte <empty> unverändert bei; es ist keine Satzgrenze.
- Setze keine Satzgrenze über eine #####ID:-Grenze hinweg.
- Vorhandene [SATZ]-, [GPT-SATZ]- und [GPT-SATZ?]-Marker markieren bereits erkannte Satzende-Kandidaten. Sie bleiben unverändert erhalten.

Marker:

- Vorhandene Satzmarker sind [SATZ], [GPT-SATZ] und [GPT-SATZ?].
- Die vorhandenen Satzmarker dürfen nicht verändert, entfernt, verschoben oder ersetzt werden.
- Neu hinzugefügt werden darf ausschließlich [GPT2-SATZ].
- [GPT2-SATZ] kennzeichnet eine im zweiten Durchgang zusätzlich vorgeschlagene Satzgrenze.

Arbeite so:

- Nutze Spalte 2 als semantische und syntaktische Entscheidungshilfe.
- Prüfe, ob der frühneuhochdeutsche Originaltext in Spalte 2 zusätzliche Satzgrenzen nahelegt, die in Spalte 1 noch nicht markiert sind.
- Ziel ist hoher Recall. Achte besonders auf mögliche bisher übersehene Satzgrenzen und setze im Zweifelsfall einen Marker.
- Setze [GPT2-SATZ] jedoch nur, wenn die Grenzstelle in der standarddeutschen Übersetzung eindeutig bestimmbar ist.
- Markiere zusätzliche Satzgrenzen in der standarddeutschen Übersetzung mit [GPT2-SATZ].
- Setze keine Satzgrenze über eine #####ID:-Grenze hinweg.
- Ignoriere leere Zeilen bei der Satzanalyse; sie sind keine Satzgrenzen.

Wichtig:

- Die Ausgabe muss exakt dieselbe Anzahl Zeilen haben wie Spalte 1 im Input.

- Füge keine zusätzlichen Zeilenumbrüche ein.
- Entferne keine Zeilenumbrüche.
- Gib ausschließlich die verbesserte Spalte 1 zurück.
- Verändere keine Wörter, keine Schreibungen, keine vorhandenen Zeichen und keine Zeilenumbrüche in der standarddeutschen Übersetzung.
- Füge ausschließlich den Marker [GPT2-SATZ] hinzu.
- Entferne keine vorhandenen [SATZ]-, [GPT-SATZ]- oder [GPT-SATZ?]-Marker.
- Verschiebe keine vorhandenen Marker.
- Ersetze keine vorhandenen Marker durch [GPT2-SATZ].
- Ein Punkt ist kein automatisches Satzende.
- Eine Virgel "/" ist kein automatisches Satzende.
- Zeilenumbrüche sind keine Satzgrenzen.
- Leere Zeilen sind keine Satzgrenzen.
- Setze [GPT2-SATZ] nur bei syntaktisch und semantisch plausiblen Grenzen.
- Setze [GPT2-SATZ] niemals mitten in ein Wort.
- Verändere Spalte 2 nicht und gib sie nicht aus.
- Setze [GPT2-SATZ] nicht direkt neben einen bereits vorhandenen Satzmarker.

Ausgabe:

Gib nur den bearbeiteten Inhalt von Spalte 1 zurück, ohne Erklärung, Kommentare oder zusätzliche Überschriften.