# Segment Any Text: A Universal Approach for Robust, Efficient and Adaptable Sentence Segmentation

**Markus Frohmann**[1,2] **Igor Sterner**[3]
**Ivan Vulić**[*3] **Benjamin Minixhofer**[*3] **Markus Schedl**[*1,2]
[1]Johannes Kepler University Linz [2]Linz Institute of Technology, AI Lab
[3]University of Cambridge
markus.{frohmann, schedl}@jku.at, {is473, bm644, iv250}@cam.ac.uk

## Abstract

Segmenting text into sentences plays an early and crucial role in many NLP systems. This is commonly achieved by using rule-based or statistical methods relying on lexical features such as punctuation. Although some recent works no longer exclusively rely on punctuation, we find that no prior method achieves all of (i) robustness to missing punctuation, (ii) effective adaptability to new domains, and (iii) high efficiency. We introduce a new model — Segment any Text (SaT) — to solve this problem. To enhance robustness, we propose a new pretraining scheme that ensures less reliance on punctuation. To address adaptability, we introduce an extra stage of parameter-efficient fine-tuning, establishing state-of-the-art performance in distinct domains such as verses from lyrics and legal documents. Along the way, we introduce architectural modifications that result in a three-fold gain in speed over the previous state of the art and solve spurious reliance on context far in the future. Finally, we introduce a variant of our model with fine-tuning on a diverse, multilingual mixture of sentence-segmented data, acting as a drop-in replacement and enhancement for existing segmentation tools. Overall, our contributions provide a universal approach for segmenting any text. Our method outperforms *all* baselines — including strong LLMs — across 8 corpora spanning diverse domains and languages, especially in practically relevant situations where text is poorly formatted.[1]

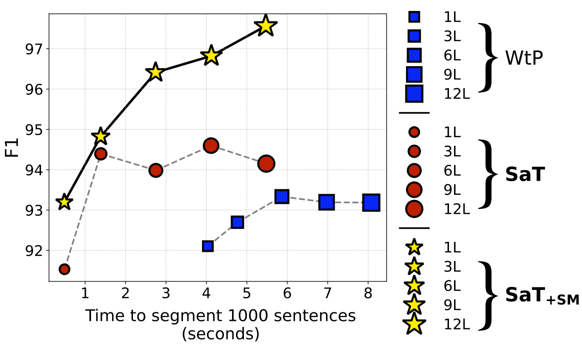



Figure 1: F1 scores and inference time for the prior SoTA (WtP) and our models (SaT and $\text{SaT}_{\text{+SM}}$), evaluated on the Ersatz sentence segmentation benchmark. We average over all 23 languages and show the average time (10 runs) for variants with different sizes (L = #layers) to segment 1,000 sentences using consumer hardware (1 Nvidia GTX 2080 Ti GPU).

## 1 Introduction

Sentence segmentation is defined as the task of identifying boundaries between sentences in a given text. High-quality sentence boundaries are crucial in many NLP tasks and systems since models often expect individual sentences as input (Reimers and Gurevych, 2019, 2020; Liu et al., 2021; Tiedemann and Thottingal, 2020, *inter alia*). Further, errors in segmentation can have detrimental effects on downstream task performance, e.g., in machine translation (Minixhofer et al., 2023; Wicks and Post, 2022; Savelka et al., 2017).

Existing sentence segmentation tools predominantly rely on punctuation marks. This limitation renders them impractical for text lacking punctuation. To address this issue, some recent methods aim to overcome this dependency (Honnibal et al., 2020; Minixhofer et al., 2023). Specifically, during the training of their model, WtP (Minixhofer et al., 2023) randomly removes punctuation characters to increase robustness against missing punctuation.

However, the performance of WtP as the current state-of-the-art (SoTA) model and all other segmenters is still poor on texts from more challenging domains. This includes, among others, user-generated text such as tweets and highly heterogeneous domains such as lyrics. Segmenting these texts is challenging because of missing and/or extra punctuation, inconsistent spacing, and espe-

[*]Equal senior authorship.
[1]Our models and code, including documentation, are available at https://github.com/segment-any-text/wtpsplit under the MIT license.



*Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11908–11941
November 12-16, 2024 ©2024 Association for Computational Linguistics

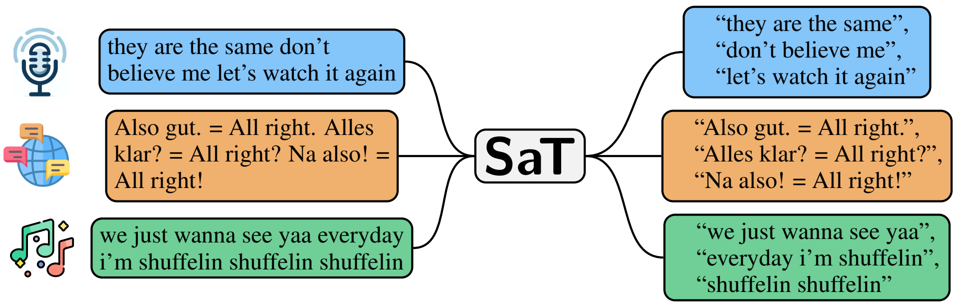



Figure 2: Examples of our model's predictions from (i) ASR output, (ii) multilingual text, and (iii) verse segmentation. (i) shows part of a transcribed TED talk, demonstrating our method is agnostic to punctuation and casing. (ii) is from a Reddit post of German-English translations; existing rule-based systems would segment at nearly every punctuation, and existing neural systems are too reliant on punctuation or need a language code. (iii) shows segmentation of lyrics into verses, showing our model's predictions in a distinct domain.

cially irregular casing. Furthermore, nearly all existing systems, including WTP, require the specification of the texts' language at inference time. This necessitates an additional preprocessing step of language identification, which often proves to be imperfect, particularly with user-generated content (Lui and Baldwin, 2014; Sterner and Teufel, 2023). Moreover, this necessity limits their applicability to code-switching text.

To address these challenges, we present a sentence segmentation method that does not rely on language codes or punctuation marks, making it *universally applicable* across a broad range of languages, corpora, and domains. Specifically, we train subword-based multilingual encoder language models (LMs) in a self-supervised way to predict naturally occurring newlines on web-scale text. We then continue training models on sentence-segmented data in a second, supervised stage to further improve sentence segmentation performance.

We deal with several major issues with previous tools: To ensure *robustness* against missing punctuation and noise, we propose a set of corruptions, applied randomly to the input during training. Crucially, our method does not rely on language codes. In addition, we mitigate issues observed with *short sequences* via a novel limited lookahead mechanism. Furthermore, we recognize the *variability of sentence boundaries* across domains and sentence definitions. To address this, we show how our models can be efficiently adapted to target domains via LoRA (Hu et al., 2022), outperforming previous adaptation methods, especially in data-constrained settings. Further, we improve efficiency by shedding the upper layers of the base model for our default 3-layer models, which segments 1000 sentences in approx. 0.5 seconds on our hardware.

Figure 1 shows that the standard 3-layer version of SAT outperforms the current open weights state-of-the-art, WTP, while achieving a $\approx$ 3x reduction in inference time. Overall, we present several innovations that overcome each of the shortcomings of previous methods, culminating in a *universal model for sentence segmentation*. We provide some examples of our model's predictions in Figure 2.

**Contributions.** **1)** We introduce *Segment any Text* (SAT), an efficient method for sentence segmentation that can reliably segment text across 85 languages regardless of lexical features such as punctuation or casing. **2)** We show how our models can be adapted to different domains via data-efficient means, requiring only a minimal set (e.g., 16) of sentence-segmented examples. **3)** We train and release SAT models in five sizes, covering 85 languages, and demonstrate state-of-the-art performance across 8 corpora, even outperforming newly introduced strong (open weights) LLM baselines.

## 2 Background and Related Work

We start by providing an overview of existing sentence segmentation systems. Following Read et al. (2012), we categorize them into **1)** rule-based, **2)** supervised statistical, and **3)** unsupervised statistical approaches. Then, we discuss the recently introduced state-of-the-art approach, WTP. Moreover, we discuss domain-specific segmentation approaches. Lastly, since we are the first to evaluate large language models (LLMs) for sentence segmentation broadly, we briefly survey them and discuss their usage in sentence segmentation tasks.

### 2.1 General Systems and Baselines

**1. Rule-based methods** segment text into sentences using hand-crafted rules. The segmenters in Moses (Koehn et al., 2007) and SpaCy (Honnibal et al., 2020) split on punctuation characters, except for predefined exceptions like abbreviations and acronyms. PySBD (Sadvilkar and Neumann, 2020) relies on exceptions and regular expression rules. Although generally efficient, these methods demand manual per-language effort to incorporate language-specific rules. This also necessitates specifying a language code at inference time.

**2. Supervised statistical methods** learn segmentation from a sentence-segmentation annotated corpus. One early method by Riley (1989) involved a decision tree to determine if each punctuation mark in a text represents a sentence boundary based on linguistic features surrounding punctuation. Satz (Palmer and Hearst, 1997) and Splitta (Gillick, 2009) build on this approach but utilize neural networks and SVMs, respectively. Similarly, in Ersatz, Wicks and Post (2021) propose to use a Transformer (Vaswani et al., 2017) with subwords as context around punctuation marks. However, these methods are limited by their reliance on punctuation to define sentence boundaries. This becomes problematic in poorly punctuated texts as non-punctuation characters cannot serve as sentence boundaries. Breaking from this limitation, the dependency parser in the SpaCy library (Honnibal et al., 2020) jointly learns dependency parsing and sentence segmentation on a labeled corpus without special treatment of punctuation.

**3. Unsupervised statistical methods** predict sentence boundaries from unsegmented text alone. Kiss and Strunk (NLTK; 2006) use features such as character length and internal punctuation to identify abbreviations, initials, and ordinal numbers, treating all other punctuation as sentence boundaries. Furthermore, Wicks and Post (2021) additionally introduces an unsupervised version of Ersatz, relying on punctuation preceding paragraph breaks.

### 2.2 Where's the Point (WtP)

WtP represents the current state-of-the-art in sentence segmentation (Minixhofer et al., 2023). Like our method, it can be used in unsupervised and supervised variations. Hence, we choose WTP as our main baseline and examine it in the following.

WTP is trained to predict the *newline probability* (i.e., the probability for any character to be followed by a \n symbol) on web-scale text data in 85 languages. Training is self-supervised since newline symbols occur naturally, typically corresponding to *paragraphs*, each containing multiple sentences. WTP thus takes characters as input and generates a probability for each character to be paragraph-ending. A character is treated as a boundary if the probability is greater than a selected threshold $\alpha$. To apply models trained in this way to segment text into *sentences*, Minixhofer et al. (2023) find it is sufficient to lower the threshold $\alpha$.

**Robustness to corruptions.** To make WTP less reliant on punctuation, Minixhofer et al. (2023) randomly remove some punctuation during training. In addition, they predict the likelihood of commonly occurring punctuation as an auxiliary objective. For details, we refer to Appendix A.5. While this helps make WTP models less reliant on punctuation, we still find that WTP models have major issues when text is inconsistently formatted, especially irregular casing.

**Efficiency.** WTP uses the character-level encoder LM Canine-S (Clark et al., 2022) as its backbone. Operating on *characters* as the fundamental unit constitutes a major bottleneck in terms of speed, resulting in poor efficiency.

**Multilinguality.** To increase language-specific capacity, WTP utilizes language adapters (Pfeiffer et al., 2022). This, however, confines its multilingual abilities since a *language code* must be specified at inference time. This is especially problematic in code-switching, where multiple languages are present, leading to ambiguity.

**Short texts.** We also found WTP models deficient in segmenting short sequences, such as tweets or sentence pairs. During training, paragraphs are packed to always fully use the model's context size. While being efficient at training, we hypothesize that this renders short sequences out-of-domain.

**Domain adaptation.** Minixhofer et al. (2023) also evaluate two supervised adaptation methods. First, $\text{WTP}_\text{T}$ tunes the segmentation threshold $\alpha$ based on an already sentence-segmented corpus. Second, based on the auxiliary punctuation prediction objective, $\text{WTP}_\text{PUNCT}$ fits a logistic regression on the probability distribution of the punctuation logits. However, these kinds of adaptations fall short on more challenging domains such as lyrics and code-switched text, as quantified later in Section 5.3.

### 2.3 Domain-specific Sentence Segmentation

Due to deviations from typical sentence structures, differences in sentence lengths, and non-standard punctuation, sentence segmentation is highly dependent on domain-specific characteristics (Sheik et al., 2024), providing a strong basis for domain-specific systems (Read et al., 2012).

Prior studies have focused on creating a dedicated model for a single domain. Reynar and Ratnaparkhi (1997) utilized features unique to the financial domain. Tuggener and Aghaebrahimian (2021) hosted a shared task on transcripts of spoken texts. Brugger et al. (2023) train models to segment sentences in the legal domain.

Previous approaches to segmenting lyrics into verses require songs to be already pre-segmented into lines. Watanabe et al. (2016) extract features based on repeated patterns and part-of-speech. Fell et al. (2018) improve upon this approach by using convolutions and a more refined set of features.

In contrast to prior domain-specific models, we propose a single model that can be efficiently adapted for segmenting sentences from wildly heterogeneous domains and languages, outperforming previous domain-specific models, even when using only a limited number of examples.

### 2.4 Large Language Models

Large language models (LLMs) have become a de facto tool for use in many NLP tasks (Zhao et al., 2023; Minaee et al., 2024). Most modern LLMs are decoder-only Transformers (Vaswani et al., 2017; Brown et al., 2020; Touvron et al., 2023; Jiang et al., 2024, *inter alia*). Recently, prompting has emerged as the dominating paradigm for solving a task (Ouyang et al., 2022; Liu et al., 2023).

However, despite widespread use, LLMs have yet to be extensively evaluated for sentence segmentation. In this work, we aim to bridge this gap by shedding light on how well popular LLMs can segment sentences when prompted to do so, particularly in more challenging domains such as lyrics, where using LLMs may be especially valuable.

## 3 SaT: Segment any Text

To create a reliable and effective system across various scenarios, we pre-train a model on paragraph segmentation as in Minixhofer et al. (2023). In the following, we outline how we solve each of the major issues of WtP discussed earlier, leading to a *universal model for sentence segmentation*.

**Efficiency.** We resort to models using subword tokenization, processing tokens consisting of *multiple characters* at a time, making them considerably faster than their character-level counterparts.

**Multilinguality.** Unlike Minixhofer et al. (2023), we do not rely on language adapters. In addition to improving inference time and storage requirements, this also improves multilinguality since no language has to be specified at inference time.

**Robustness to corruptions.** We randomly remove common punctuation-only tokens with probability $p$ and use the auxiliary punctuation-prediction objective during training. For details, see § 2.2 and Appendix A.5. Further, we randomly remove *all* casing and punctuation in 10% of samples within a batch during training. The resulting model, *Segment any Text* (SaT), already shows strong segmentation performance at improved efficiency.

Still, to further improve SaT, we continue training it on a **S**upervised **M**ixture of already-segmented sentences. To be even less dependent on patterns such as punctuation, spaces, and casing, we augment the data by introducing several additional corruption schemes, resulting in our more specialized model, SaT$_{\text{+SM}}$.

Our first corruption scheme removes all casing, if available, and punctuation tokens for all text. Secondly, we add randomness to the corruption in as many situations as we find useful, aiming to emulate user-generated text in tweets or forums. This includes duplicating punctuation, removing punctuation, lowercasing, and removing/adding spaces between sentences. Finally, we also use clean, non-corrupted text. We then sample uniformly across these three categories. For details, see Appendix A.2.

**Short texts.** To resolve issues with short sequences, we enforce SaT to use only the immediate $N$ future tokens for its predictions. We do so via a *limited lookahead* mechanism. Let $k_i$ be the token occurring at position $i$, and $\mathbf{a_{ij}}$ its corresponding attention mask, where $j$ corresponds to the token to be attended to. A naive modification of the attention mask would set $\mathbf{a_{ij}} = 0$ for $j > i + N$. However, using Transformer networks with multiple layers results in a lookahead of $N \times L$, where $L$ is the number of Transformer layers (Jiang et al., 2023). We thus split up the lookahead evenly into $L$ layers, resulting in the following attention mask:

$$\mathbf{a_{ij}} = 0 \text{ for } j > i + N_L,$$

where $N_L$ is the per-layer lookahead, i.e., $N_L = \frac{N}{L}$. Using an intermediate value for $N$ makes SaT robust to both short and long sequences – relying

| Domain | Dataset | Description | Characteristics | Source |
|---|---|---|---|---|
| Clean Text | Universal Dependencies (UD) | Treebanks in many languages. | Includes gold-standard segmentation into sentences. | de Marneffe et al. (2021); Nivre et al. (2020) |
| | OPUS100 | Sentences from subtitles and news in 100 languages. | A challenging sentence segmentation benchmark (Zhang et al., 2020) | Tiedemann (2012) |
| | Ersatz | Sentences from WMT Shared Tasks, mainly comprising news (commentary). | Includes manual sentence segmentation corrections by Wicks and Post (2021). | Wicks and Post (2021); Barrault et al. (2020) |
| Noisy Text | SEPP-NLG Shared Task (surprise test set) | 500 transcribed public TED talks in each of 4 European languages. | Neither casing nor punctuation tokens are present. | Tuggener and Aghaebrahimian (2021) |
| | Tweets | User-generated content in the form of Slovene (`sl`) and Serbian (`sr`) tweets. | Noisy; short in length (70/115 characters on average for `sl` and `sr`, respectively). | Fišer et al. (2020); Miličević and Ljubešić (2016) |
| Code-switching | *C.f.* Table 10. | Reddit posts for German-English (`de-en`); data for 3 additional language pairs taken by concatenating code-switching sentences from bilingual transcriptions. | We treat all data as transcriptions, removing all punctuation and casing; we only keep sentences with at least one token of each language. | Deuchar (2009), Osmelak and Wintner (2023), Nguyen and Bryant (2020) and Çetinoğlu (2017) |
| Legal | MultiLegalSBD | Laws and judgements from legal documents in 6 languages. | Domain-specific jargon and structure; formal and complex sentences. | Brugger et al. (2023) |
| Lyrics | Verses | 35,389 English songs across 16 genres spanning 3 levels of repetitiveness. | We replicated the setup by Fell et al. (2018). | Meseguer-Brocal et al. (2017) |

Table 1: Overview of the evaluation corpora we use. For more details, see Appendix A.2.

on some future context where appropriate, but not so much that it falters on short sequences.

Limited lookahead can be thought of as sliding window attention (Beltagy et al., 2020; Jiang et al., 2023) with two crucial tweaks: 1) the sliding window extends forward into the future instead of backward, 2) past tokens are not masked out.

**Domain adaptation.** Finally, some domains require more sophisticated adaptation than only changing the threshold or relying on punctuation logits. We thus explore low-rank adaptation (LoRA; Hu et al., 2022) to adapt our models efficiently, denoted by $\text{SAT}_{\text{+LoRA}}$. We show how this enables state-of-the-art performance on verse segmentation using our models later in § 5. In our setup, it trains $\approx 1\%$ of the parameters of SAT but results in *no inference overhead* since LoRA weights can be merged into the backbone LM weights at inference time (Pfeiffer et al., 2023).

# 4 Experimental Setup

## 4.1 Evaluation

To evaluate our method, we compare ground truth and predicted sentence boundaries on the test sets of corpora spanning a diverse set of languages, sources, and domains,[2] summarized in Table 1.

In addition, to evaluate how well our method can segment short sequences, we generate non-overlapping sentence pairs from the datasets categorized as clean text. We additionally simulate a real-time automatic speech recognition (ASR) scenario using transcripts from speeches in 76 languages. We generate sentence pairs in a similar way and remove all punctuation as well as all casing.

We report character-level F1 scores for the positive (i.e., sentence-ending) labels. For short sequences, we use the proportion of perfectly segmented sequences within a corpus; this is stricter than F1 since any segmentation error results in a score of zero for the entire sequence. For SEPP-NLG, we use the evaluation script and surprise test set provided by the shared task organizers (Tuggener and Aghaebrahimian, 2021), reporting F1 scores on the *token* level. In our evaluations on clean text across all 85 languages, we run all competitor and baseline systems ourselves. For these results, we test all differences for significance with paired two-tailed permutation tests. We approximate them with $N = 10,000$ and set the significance threshold at $\alpha = 0.05$. Additional evaluation and dataset details are provided in Appendix A.2.

**Baselines.** We compare against PySBD and NLTK as representatives of rule-based and unsupervised statistical methods. For supervised methods, we evaluate the punctuation-agnostic $\text{SpaCy}_{\text{DP}}$ and Spacy's multi-language model, $\text{SpaCy}_{\text{M}}$. We also compare against Ersatz. Our main comparison is against the current SoTA models: WtP, $\text{WtP}_{\text{T}}$, and $\text{WtP}_{\text{PUNCT}}$.

**LLM-based baselines.** To evaluate LLMs, we use **1)** Cohere's Command R as a recent LLM with claimed strong multilingual performance, and **2)** Meta's Llama $3_{8B}$ due to its popularity and strong performance. Officially, Command R supports 23 languages, whereas Llama $3_{8B}$ only supports

[2]We acknowledge the concept of *domains* remains an open issue in NLP (Holtermann et al., 2024; Raffel et al., 2019).

| Model | ar | cs | de | en | es | fi | hi | ja | ka | lv | pl | th | xh | zh | **81 langs** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | MULTILINGUAL | | | | | | | |
| SPACY$_{M}$ | - | 91.1 | 84.7 | 91.5 | 94.5 | 93.5 | - | - | - | 91.4 | 94.0 | - | - | - | - |
| ERSATZ | 77.2 | 90.9 | 87.0 | 91.4 | 95.1 | 93.9 | 84.8 | 69.3 | - | 91.1 | 94.8 | - | - | 77.2 | - |
| LLAMA 3$_{8B}$ | 78.2 | 93.4 | 92.6 | 95.2 | 96.0 | 95.5 | 85.6 | 64.7 | 89.2 | 93.0 | 96.2 | 66.0 | 71.7 | 82.0 | 79.1 |
| COMMAND R | 58.6 | 68.1 | 79.1 | 84.6 | 81.0 | 74.1 | 72.0 | 52.2 | 25.6 | 74.2 | 78.6 | 10.6 | 56.1 | 73.7 | 55.6 |
| SAT | 79.9 | 91.7 | 90.4 | 93.6 | 94.0 | 94.2 | 84.9 | **88.6** | 75.7 | 92.2 | 93.7 | 68.0 | 80.3 | 78.0 | 84.9 |
| SAT$_{+SM}$ | **80.7** | **95.7** | **94.0** | **96.5** | **97.3** | **96.9** | **90.3** | 88.1 | **93.6** | **96.1** | **97.7** | **72.9** | **89.6** | **88.9** | **91.6** |
| | | | | | | | | MONOLINGUAL | | | | | | | |
| NLTK | - | 90.8 | 87.1 | 92.2 | 94.1 | 93.9 | - | - | - | - | 94.5 | - | - | - | - |
| PYSBD | 37.4 | - | 80.6 | 69.6 | 56.9 | - | 70.1 | 76.1 | - | - | 49.3 | - | - | 86.9 | - |
| SPACY$_{DP}$ | - | - | 89.0 | 92.9 | 93.5 | 94.1 | - | 77.1 | - | - | 95.3 | - | - | 87.7 | - |
| WTP | 77.3 | 91.1 | 89.2 | 93.9 | 93.2 | 93.4 | 85.0 | 72.7 | 91.3 | 90.4 | 93.6 | 66.6 | 77.2 | 90.7 | 84.2 |
| WTP$_{T}$ | 79.9 | 92.0 | 92.0 | 93.5 | 94.2 | 94.1 | 85.2 | 85.6 | 91.1 | 93.1 | 93.5 | 69.7 | 80.7 | 89.3 | 85.9 |
| WTP$_{PUNCT}$ | 85.4 | **96.4** | 95.0 | **96.7** | 97.4 | **97.7** | 90.8 | 93.1 | 92.8 | 96.6 | 97.5 | 71.3 | 89.8 | **95.5** | 91.7 |
| SAT$_{+LoRA}$ | **86.3** | 96.2 | **95.4** | **96.7** | **97.7** | 97.5 | **92.9** | **94.4** | **93.3** | **97.0** | **97.7** | **73.7** | **90.8** | 94.9 | **93.1** |

Table 2: Mean sentence segmentation F1 scores over OPUS100, UD and Ersatz. For the average, we report macro F1 over languages from all datasets where train and test sets are available. Results are shown using 3-layer variations of all models. Numerically best results are in **bold**, statistically indistinguishable ones from this best are underlined.

| Model | en | de | fr | it | Avg. |
|---|---|---|---|---|---|
| htw+t2k | 77 | 82 | 76 | 75 | 78 |
| OnPoint | 80 | 82 | 77 | **77** | 79 |
| Unbabel | **83** | 78 | **78** | 76 | 79 |
| SAT | 73.4 | 79.9 | 73.1 | 72.9 | 74.8 |
| SAT$_{+SM}$ | 79.7 | **84.0** | **78.3** | **77.1** | **79.8** |

Table 3: F1 scores on the surprise test set of the SEPP-NLG Shared Task. For comparison, we provide results for the 3 best-performing systems from the Shared Task. We use 12-layer versions of our models.

English.[3] We split up each dataset into chunks of 10 sentences to avoid cases where sentences are cut off at critical positions and observed issues with long context lengths. Then, we prepend the prompt to each chunk and let the LLM segment 10 sentences.[4] Finally, to make evaluation metrics robust to unwanted alterations of the input by the LLM, we apply the Needleman-Wunsch algorithm (NW; Needleman and Wunsch, 1970) to align sentences within each input and output chunk. For the prompt and other implementation details, including alignment via NW, we refer to Appendix A.2.

### 4.2 Training Setup

We train Transformer models operating on subwords, initialized with the weights of XLM-RoBERTa (XLM-R; Conneau et al., 2020). We use a lookahead limit of 48 tokens, which we found to work well in practice on text of any length, leading to **SAT**. We use the mC4 (Raffel et al., 2019) corpus and sample text uniformly from the 85 languages also used by Minixhofer et al. (2023).

To train **SAT$_{+SM}$**, we continue training SAT on the training set of UD due to its high quality and availability in most of the 85 considered languages. For languages without UD data, we resort to silver-quality data from OPUS100 or NLLB (Costa-jussà et al., 2022), whichever is available.

To adapt to different user requirements w.r.t. *efficiency*, we train and release SAT and SAT$_{+SM}$ models in different sizes from 1-12 layers, where we remove the upper layers for models $<$ 12 layers.

For adaptation via LoRA (**SAT$_{+LoRA}$**), we use SAT as a starting point.[5] We use the respective training set using max. 10,000 sentences.

The full details of the experiment setup regarding the datasets, infrastructure, training, and hyperparameters are provided in Appendix A.2.

## 5 Results

### 5.1 Performance on Clean Text

Table 2 shows evaluation results on clean text, averaged over OPUS100, UD, and Ersatz on a diverse selection of languages, including an average over 81 languages.[6] We categorize methods into

[3] Due to imperfect filtering of common web-crawled corpora, all LLMs can be considered multilingual to some extent.

[4] For a fair comparison, we thus exclude every 10th label when calculating F1 scores. For songs and short sequences, we feed in whole samples and hence do not exclude any labels.

[5] We include its task head since we found that it improves stability. We also experimented with applying LoRA to SAT$_{+SM}$, but did not find it to improve upon SAT$_{+LoRA}$.

[6] For the average, we only consider languages with datasets with both train and test sets for a fair comparison. While we evaluate on 85 languages, this is the case in 81 languages.

| Model | Tweets | | Sentence Pairs | | Macro Avg. |
|---|---|---|---|---|---|
| | sl | sr | Speeches | Ersatz | |
| LLAMA $3_{8B}$ | 73.4 | 76.0 | 66.9 | 94.8 | 77.8 |
| COMMAND R | 53.8 | 47.4 | 23.0 | 70.0 | 48.6 |
| WTP | 70.8 | 71.4 | 12.6 | 78.0 | 58.2 |
| $\text{WTP}_{\text{T}}$ | 70.4 | 71.4 | 18.9 | 79.0 | 59.9 |
| $\text{WTP}_{\text{PUNCT}}$ | 80.1 | 82.3 | 37.9 | 91.5 | 72.9 |
| SAT | 80.5 | 75.5 | 28.8 | 84.0 | 67.2 |
| $\text{SAT}_{\text{+SM}}$ | 78.0 | 72.9 | 41.7 | 92.5 | 71.3 |
| $\text{SAT}_{\text{+LoRA}}$ | **87.2** | **89.1** | **56.8** | **93.9** | **81.8** |

Table 4: Proportion of perfectly segmented short sequences. For Speeches and Ersatz, we are averaging scores over languages. We use 12-layer versions of each model given the task's increased difficulty.[7]

(i) *multilingual*, which take only text as input, and (ii) *monolingual*, which additionally rely on a language code or, in the case of $\text{WTP}_{\text{T}}$, $\text{WTP}_{\text{PUNCT}}$, and $\text{SAT}_{\text{+LoRA}}$, are adapted to a target domain.

Both SAT and $\text{SAT}_{\text{+SM}}$ outperform the current non-domain-adapted SoTA model, WTP. Meanwhile, unlike WTP, our models do not rely on specifying a language code as input.

Remarkably, $\text{SAT}_{\text{+SM}}$ and $\text{WTP}_{\text{PUNCT}}$ are not statistically significantly different, achieving average F1 scores of 91.6 and 91.7 respectively. This is despite $\text{WTP}_{\text{PUNCT}}$ relying on adaptation to a target sentence-segmented corpus, whilst $\text{SAT}_{\text{+SM}}$ is a general-purpose multilingual model. Finally, $\text{SAT}_{\text{+LoRA}}$ significantly outperforms the existing domain-adapted SoTA, $\text{WTP}_{\text{PUNCT}}$, making it the best overall model. Our domain-adapted model outperforms $\text{WTP}_{\text{PUNCT}}$ in 63 out of 81 languages.

Among the LLMs, COMMAND R, despite being trained in 23 languages, does surprisingly poorly, with LLAMA $3_{8B}$ surpassing it by 23.5% absolute avg. F1. Nevertheless, LLAMA $3_{8B}$ still falls short compared to all variants of SAT. On the English benchmarks, given the abundance of English text, we expected our models to be easily outperformed by LLMs; yet, unlike WTP, $\text{SAT}_{\text{+SM}}$ outperforms both LLMs on *every* dataset.

We provide full per-dataset results, including all 85 languages, in § A.4. We also conduct ablation studies on each of our stages' components in § A.1.

## 5.2 Performance on Noisy and Short Text

Table 3 presents the results of our method when evaluated on the SEPP-NLG Shared Task. $\text{SAT}_{\text{+SM}}$ establishes a new state-of-the-art, outperforming the SEPP-NLG winners. This is despite our model supporting 81 additional languages and use cases not considered in the Shared Task.

Furthermore, Table 4 shows evaluation results on short sequences, including tweets and sentence pairs taken from manually corrupted speeches and Ersatz. We observe similar patterns on these corpora: SAT and $\text{SAT}_{\text{+SM}}$ outperform WTP, improving avg. F1 scores by 9% and 13.1%, respectively, $\text{SAT}_{\text{+LoRA}}$ continues to be the best overall model, also outperforming both LLMs. We additionally provide an ablation study showing the importance of limited lookahead in SAT in Table 9.

| Model | es en | de en | vi en | tr de | Macro Avg. |
|---|---|---|---|---|---|
| LLAMA $3_{8B}$ | 47.9 | 56.3 | 35.5 | 33.9 | 43.4 |
| COMMAND R | 30.4 | 51.9 | 30.0 | 17.6 | 32.5 |
| $\text{SPACY}_{\text{DP}}$* | 17.6 | 8.6 | 11.3 | 12.2 | 12.2 |
| WTP* | 38.6 | 39.0 | 25.5 | 33.5 | 29.1 |
| $\text{WTP}_{\text{T}}$* | 52.2 | 45.7 | 46.7 | 34.4 | 43.2 |
| $\text{WTP}_{\text{PUNCT}}$* | 62.1 | 60.1 | 59.0 | 41.0 | 54.9 |
| SAT | 54.5 | 49.2 | 49.3 | 39.8 | 48.2 |
| $\text{SAT}_{\text{+SM}}$ | 59.6 | 58.4 | 57.3 | 42.4 | 54.4 |
| $\text{SAT}_{\text{+LoRA}}$ | **65.0** | **65.6** | **67.5** | **48.8** | **61.7** |

Table 5: Sentence segmentation F1 scores for code-switched text. We use 12-layer versions of each model. * indicates models using language codes, where we try both language codes and show the better score. We show results using both language codes in Appendix A.4.

## 5.3 Performance on Challenging Domains

**Code-switching.** The results in Table 5 reveal that WTP achieves an average F1 score of 29.1%, while the highest-performing LLM scores 43.4%. SAT and $\text{SAT}_{\text{+SM}}$ achieve average F1 of 48.2% and 54.4%, respectively. $\text{SAT}_{\text{+LoRA}}$ continues to improve performance, achieving 61.7%. To the best of our knowledge, this is the first comprehensive evaluation of sentence segmentation tools on code-switching text. While our models now represent SoTA, the evaluation results indicate that it is a challenging task.

We now evaluate domain adaptation performance of our method on two highly distinct domains: lyrics and legal data.

**Lyrics.** Table 6 shows results on verse segmentation (i.e., segmenting songs into verse, chorus, bridge, etc.). None of the other baseline systems, including LLMs, can improve over the current domain-specific SotA, $\text{SSM}_{\text{string}}$. In contrast, $\text{SAT}_{\text{+LoRA}}$ outperforms $\text{SSM}_{\text{string}}$ by 10% avg. F1. The difference is especially pronounced in hard-to-segment songs that are low in repetitiveness (e.g.,

[7]We exclude other baselines since none of them support sl/sr or all languages from TED/Ersatz.

| Model | Corrupted? | | Repetitiveness | | |
|---|---|---|---|---|---|
| | ✓ | ✗ | High | Mid | Low |
| SSM$_{\text{string}}$ † | - | 63.8 | 71.3 | 64.8 | 47.3 |
| LLAMA 3$_{8B}$ | 45.5 | 49.7 | 48.9 | 46.7 | 33.8 |
| COMMAND R | 36.3 | 38.3 | 38.0 | 37.1 | 28.7 |
| WTP$_{\text{PUNCT}}$@100 | 46.9 | 53.8 | 55.8 | 55.2 | 35.9 |
| WTP$_{\text{PUNCT}}$@1000 | 49.1 | 56.1 | 58.4 | 57.5 | 44.9 |
| WTP$_{\text{PUNCT}}$ | 49.2 | 56.2 | 58.4 | 57.6 | 44.9 |
| SAT$_{\text{+LoRA}}$@100 | 60.8 | 62.4 | 67.8 | 62.9 | 51.6 |
| SAT$_{\text{+LoRA}}$@1000 | 67.3 | 72.4 | 76.5 | 73.1 | **62.7** |
| SAT$_{\text{+LoRA}}$ | **68.5** | **73.8** | **77.9** | **74.8** | 62.3 |

Table 6: Macro-averaged verse segmentation performance over per-genre F1 scores. †Values for SSM$_{\text{string}}$ taken from Fell et al. (2018), with lyrics already presegmented into lines. @N corresponds to using a maximum of N songs per genre for adaptation.

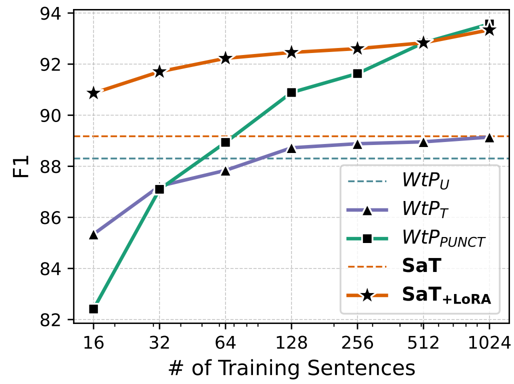



Figure 3: Macro avg. F1 vs. number of sentences used for adaptation, averaged over languages in {OPUS100, UD, Ersatz}. Per-corpus results shown in Appendix A.1.

Rap music), with a 15% difference in F1 scores. When evaluating SAT$_{\text{+LoRA}}$ on manually corrupted lyrics, it still outperforms *all* baselines, even when compared to baselines evaluated on non-corrupted songs. Additionally, SAT$_{\text{+LoRA}}$@1000, using 1000 songs per genre for adaptation, still outperforms all baselines. We provide complete results, including those for each genre, in Appendix A.4.

**Legal and qualitative examples..** We provide comprehensive results on MultiLegalSBD in Appendix A.4. Finally, We provide qualitative examples from several domains in Appendix A.3.

## 6 Discussion

**LLMs.** Contrary to our expectations, our evaluation results reveal that LLMs generally underperform, particularly in non-English languages. Notably, when using LLMs for sentence segmentation via prompting, each sentence is processed twice – once as part of the input, appended to the prompt, and once within the output. This redundancy leads to inefficient processing, needing to copy the input verbatim to the output, ideally only adding newlines. However, in practice, LLMs are highly prone to alter their input (Barbero et al., 2024). We found this issue to be particularly severe for noisy text and lyrics.[8] This is highly problematic for a specific task requiring input and output characters to remain the same. Still, we tried to address this by using the Needleman-Wunsch sequence alignment algorithm to make pure segmentation performance comparable to other methods.[9]

Aiming to improve the segmentation performance of LLMs, we experimented with few-shot prompting. However, this did not yield the desired improvements; in fact, it degraded performance. Additionally, we tested varying the number of input-output sentences. The results of both of these ablation studies are presented in Appendix A.1.

**Efficiency.** For our method, we rely on XLM-R as the LM backbone. Operating on subwords makes SAT considerably faster than WTP. We compare sentence segmentation performance and time on Ersatz across model sizes from 1-12 layers, illustrated in Figure 1. Additional datasets are shown in Figure 4. The standard 3-layer variations of SAT take $\approx$ 0.5 seconds to segment 1000 sentences on the hardware specified in Appendix A.2, making them 3 times faster than WTP, while also outperforming WTP models in *all* sizes on Ersatz. Furthermore, for SAT, performance plateaus with sizes $>$ 3 layers, whereas SAT$_{\text{+SM}}$ continues to improve when scaling up its size, making it by far the best model, despite never being exposed to Ersatz.

**Few-shot domain adaptation.** We now analyze how many sentences are needed to adapt our domain adaptation method, SAT$_{\text{+LoRA}}$, to a target corpus, and compare it to previous methods. As shown in Figure 3, WTP$_{\text{T}}$ and WTP$_{\text{PUNCT}}$ perform similarly when using 1024 sentences for domain adaptation. However, WTP$_{\text{PUNCT}}$ fails to outperform the fully self-supervised variation of WTP when $\leq$ 32 sentences are available. In contrast, SAT$_{\text{+LoRA}}$ markedly improves upon the self-supervised SAT with only 16 available sentences, and outperforms WTP$_{\text{PUNCT}}$ by almost 10% in F1 score, making it substantially *more sample-efficient*.

[8] LLAMA 3$_{8B}$ and COMMAND R altered 1.5% and 2% of all characters within lyrics, respectively, even though we prompted them not to alter their input (c.f. Appendix A.2).

[9] The same objective could be achieved via other means, e.g., constrained decoding (Beurer-Kellner et al., 2024).

## 7 Conclusion

We proposed SAT, an efficient, robust, and highly adaptable multilingual sentence segmentation method that neither relies on language codes nor punctuation. Further, we introduced $SAT_{+SM}$, improving SAT via supervised adaptation using multiple corruption schemes. Our method consistently achieves state-of-the-art performance among open weights models in experiments across 85 languages and eight diverse corpora, even outperforming newly introduced and optimized strong LLM baselines. We also demonstrated that SAT can be efficiently domain-adapted via LoRA, setting new performance standards on segmentation of lyrics and code-switching text. Overall, we hope SAT will unlock significantly improved text data (pre-)processing across a range of NLP applications for multiple languages and domains via its robust and consistently strong performance, versatility, and high efficiency.

## Limitations

To the best of our knowledge, our evaluations are the most comprehensive to date, spanning 8 diverse corpora across different domains, languages, and noise levels, and sequence lengths. Still, we may not have covered every possible scenario. Second, since we use XLM-R as our backbone, we also use its tokenizer, which has been shown to tokenize text less efficiently in some language (Liang et al., 2023), potentially exacerbating existing biases. We try to minimize bias w.r.t. performance by sampling text from all languages uniformly in both stages. Furthermore, our use of subword LMs merges characters into subwords. Theoretically, this could limit sentence boundaries to end-of-token positions; however, in practice, we did not find this to be an issue. Finally, language support could be further improved by e.g., replacing mC4 with MADLAD-400 (Kudugunta et al., 2023) for the pre-training stage. We leave this to future work.

## Ethical Considerations

Our work is multifaceted, as are the ethical dimensions it encompasses. First, we acknowledge the possibility of NLP datasets and models for encoding unfair stereotypical (Blodgett et al., 2020) and exclusive (Dev et al., 2021) biases that may lead to representational and allocational harms (Barocas et al., 2017). This issue is a general property of pre-trained LMs, and the models and datasets utilized in our study are similarly at risk. We advise practitioners to use these models with the appropriate care and we refer to existing works (Liang et al., 2021; Lauscher et al., 2021) for discussions on bias mitigation. Second, one key aspect of our work deals with efficiency. On the one hand, considering the well-documented relationship between model training efforts and potential $CO_2$ emissions (Strubell et al., 2019), our research contributes to Green AI by improving the environmental sustainability of state-of-the-art sentence segmentation systems. On the other hand, since the training of language models often comes with high infrastructure prerequisites only available to certain user groups (Bender et al., 2021), we hope that our work also contributes to the continued democratization of language technology by reducing resource- and language-related usage barriers.

## Acknowledgments

This research was funded in whole or in part by the Austrian Science Fund (FWF): `https://doi.org/10.55776/P33526`, `https://doi.org/10.55776/DFH23`, `https://doi.org/10.55776/COE12`, `https://doi.org/10.55776/P36413`. In addition, Ivan Vulić and Benjamin Minixhofer have been supported through the Royal Society University Research Fellowship *'Inclusive and Sustainable Language Technology for a Truly Multilingual World'* (no 221137) awarded to Ivan Vulić. This research has also been supported with Cloud TPUs from Google's TPU Research Cloud (TRC). This work was also supported by compute credits from a Cohere For AI Research Grant, these grants are designed to support academic partners conducting research with the goal of releasing scientific artifacts and data for good projects. We also thank Simone Teufel for fruitful discussions.

## References

Federico Barbero, Andrea Banino, Steven Kapturowski, Dharshan Kumaran, Joao G.M. Ara'ujo, Alex Vitvitskyi, Razvan Pascanu, and Petar Velivckovi'c. 2024. Transformers need glasses! information over-squashing in language tasks.

Solon Barocas, Kate Crawford, Aaron Shapiro, and Hanna Wallach. 2017. The problem with bias: Allocative versus representational harms in machine learning. In *9th Annual Conference of the Special Interest Group for Computing, Information and Society*.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *ArXiv*, abs/2004.05150.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Luca Beurer-Kellner, Marc Fischer, and Martin T. Vechev. 2024. Guiding llms the right way: Fast, non-invasive constrained generation. *ArXiv*, abs/2403.06988.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of "bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Tobias Brugger, Matthias Sturmer, and Joel Niklaus. 2023. Multilegalsbd: A multilingual legal sentence boundary detection dataset. *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*.

Özlem Çetinoğlu. 2017. A code-switching corpus of Turkish-German conversations. In *Proceedings of the 11th Linguistic Annotation Workshop*, pages 34–40, Valencia, Spain. Association for Computational Linguistics.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. Canine: Pre-training an efficient tokenization-free encoder for language representation. *Transactions of the Association for Computational Linguistics*, 10:73–91.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Y. Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loïc Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *CoRR*, abs/2207.04672.

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

Margaret Deuchar. 2009. The miami corpus: Documentation file.

Sunipa Dev, Masoud Monajatipoor, Anaelia Ovalle, Arjun Subramonian, Jeff Phillips, and Kai-Wei Chang. 2021. Harms of gender exclusivity and challenges in non-binary representation in language technologies. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1968–1994, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Michael Fell, Yaroslav Nechaev, Elena Cabrio, and Fabien Gandon. 2018. Lyrics segmentation: Textual macrostructure detection using convolutions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2044–2054, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Darja Fišer, Nikola Ljubešić, and Tomaž Erjavec. 2020. The janes project: language resources and tools for slovene user generated content. *Language resources and evaluation*, 54(1):223–246.

Dan Gillick. 2009. Sentence boundary detection and the problem with the U.S. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association* for Computational Linguistics, Companion Volume: Short Papers*, pages 241–244, Boulder, Colorado. Association for Computational Linguistics.

Carolin Holtermann, Markus Frohmann, Navid Rekabsaz, and Anne Lauscher. 2024. What the weight?! a unified framework for zero-shot knowledge composition. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1138–1157, St. Julian's, Malta. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python. Available at `https://doi.org/10.5281/zenodo.1212303`.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, L'elio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *ArXiv*, abs/2401.04088.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *ArXiv*, abs/2310.06825.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Sneha Kudugunta, Isaac Caswell, Biao Zhang, Xavier García, Christopher A. Choquette-Choo, Katherine Lee, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, and Orhan Firat. 2023. Madlad-400: A multilingual and document-level large audited dataset. *ArXiv*, abs/2309.04662.

Anne Lauscher, Tobias Lueken, and Goran Glavaš. 2021. Sustainable modular debiasing of language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4782–4797, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Davis Liang, Hila Gonen, Yuning Mao, Rui Hou, Naman Goyal, Marjan Ghazvininejad, Luke Zettlemoyer, and Madian Khabsa. 2023. XLM-V: Overcoming the vocabulary bottleneck in multilingual masked language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13142–13152, Singapore. Association for Computational Linguistics.

Paul Pu Liang, Chiyu Wu, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2021. Towards understanding and mitigating social biases in language models. In *International Conference on Machine Learning*, pages 6565–6576. PMLR.

Fangyu Liu, Ivan Vulić, Anna Korhonen, and Nigel Collier. 2021. Fast, effective, and self-supervised: Transforming masked language models into universal lexical and sentence encoders. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1442–1459, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

Marco Lui and Timothy Baldwin. 2014. Accurate language identification of Twitter messages. In *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)*, pages 17–25, Gothenburg, Sweden. Association for Computational Linguistics.

Gabriel Meseguer-Brocal, Geoffroy Peeters, Guillaume Pellerin, Michel Buffa, Elena Cabrio, Catherine Faron Zucker, Alain Giboin, Isabelle Mirbel, Romain Hennequin, Manuel Moussallam, Francesco Piccoli, and Thomas Fillon. 2017. WASABI: a Two Million Song Database Project with Audio and Cultural Metadata plus WebAudio enhanced Client Applications. In *Web Audio Conference 2017 – Collaborative Audio #WAC2017*, London, United Kingdom. Queen Mary University of London.

Maja Miličević and Nikola Ljubešić. 2016. Tviterasi, tviteraši ali twiteraši? izdelava in analiza normaliziranega nabora hrvaških in srbskih tvitov. *Slovenščina 2.0: empirične, aplikativne in interdisciplinarne raziskave*, 4(2):156–188.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Asgari Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *ArXiv*, abs/2402.06196.

Benjamin Minixhofer, Jonas Pfeiffer, and Ivan Vulić. 2023. Where's the point? self-supervised multilingual punctuation-agnostic sentence segmentation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7215–7235, Toronto, Canada. Association for Computational Linguistics.

Saul B. Needleman and Christian D. Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48 3:443–53.

Li Nguyen and Christopher Bryant. 2020. CanVEC - the canberra Vietnamese-English code-switching natural speech corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4121–4129, Marseille, France. European Language Resources Association.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Doreen Osmelak and Shuly Wintner. 2023. The denglisch corpus of German-English code-switching. In *Proceedings of the 5th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 42–51, Dubrovnik, Croatia. Association for Computational Linguistics.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. 2022. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155.

David D. Palmer and Marti A. Hearst. 1997. Adaptive multilingual sentence boundary disambiguation. *Computational Linguistics*, 23(2):241–267.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

Jonas Pfeiffer, Naman Goyal, Xi Lin, Xian Li, James Cross, Sebastian Riedel, and Mikel Artetxe. 2022. Lifting the curse of multilinguality by pre-training modular transformers. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3479–3495, Seattle, United States. Association for Computational Linguistics.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54.

Jonas Pfeiffer, Sebastian Ruder, Ivan Vulić, and Edoardo Ponti. 2023. Modular deep learning. *Transactions on Machine Learning Research*. Survey Certification.

Clifton Poth, Hannah Sterz, Indraneil Paul, Sukannya Purkayastha, Leon Engländer, Timo Imhof, Ivan Vulić, Sebastian Ruder, Iryna Gurevych, and Jonas Pfeiffer. 2023. Adapters: A unified library for parameter-efficient and modular transfer learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 149–160, Singapore. Association for Computational Linguistics.

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

Jonathon Read, Rebecca Dridan, Stephan Oepen, and Lars Jørgen Solberg. 2012. Sentence boundary detection: A long solved problem? In *Proceedings of COLING 2012: Posters*, pages 985–994, Mumbai, India. The COLING 2012 Organizing Committee.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Jeffrey C. Reynar and A. Ratnaparkhi. 1997. A maximum entropy approach to identifying sentence boundaries. In *Applied Natural Language Processing Conference*.

Michael D. Riley. 1989. Some applications of tree-based modelling to speech and language. In *Speech and Natural Language: Proceedings of a Workshop Held at Cape Cod, Massachusetts, October 15-18, 1989*.

Nipun Sadvilkar and Mark Neumann. 2020. PySBD: Pragmatic sentence boundary disambiguation. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, pages 110–114, Online. Association for Computational Linguistics.

Jaromir Savelka, Vern R. Walker, Matthias Grabmair, and Kevin D. Ashley. 2017. Sentence boundary detection in adjudicatory decisions in the united states. In *ICON*.

Reshma Sheik, Sneha Rao Ganta, and S. Jaya Nirmala. 2024. Legal sentence boundary detection using hybrid deep learning and statistical models. *Artificial Intelligence and Law*.

Igor Sterner and Simone Teufel. 2023. TongueSwitcher: Fine-grained identification of German-English code-switching. In *Proceedings of the 6th Workshop on Computational Approaches to Linguistic Code-Switching*, pages 1–13, Singapore. Association for Computational Linguistics.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal. European Association for Machine Translation.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Don Tuggener and Ahmad Aghaebrahimian. 2021. The sentence end and punctuation prediction in nlg text (sepp-nlg) shared task 2021. In *Swiss Text Analytics Conference*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Kento Watanabe, Yuichiroh Matsubayashi, Naho Orita, Naoaki Okazaki, Kentaro Inui, Satoru Fukayama, Tomoyasu Nakano, Jordan Smith, and Masataka Goto. 2016. Modeling discourse segments in lyrics using repeated patterns. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1959–1969, Osaka, Japan. The COLING 2016 Organizing Committee.

Rachel Wicks and Matt Post. 2021. A unified approach to sentence segmentation of punctuated text in many languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3995–4007, Online. Association for Computational Linguistics.

Rachel Wicks and Matt Post. 2022. Does sentence segmentation matter for machine translation? In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 843–854, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In

*Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Z. Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jianyun Nie, and Ji rong Wen. 2023. A survey of large language models. *ArXiv*, abs/2303.18223.

## A Appendix

### A.1 Ablation Studies

| Model | Variation | Clean Text | Tweets | Code Switching |
|---|---|---|---|---|
| SAT | - | **84.9** | 78.0 | **48.2** |
| | Only clean text | 84.7 | 33.5 | 16.5 |
| SAT+SM | - | **91.6** | 75.5 | **54.4** |
| | Only clean text | 91.5 | **77.2** | 10.2 |
| | No pre-training | 89.9 | 42.1 | 44.0 |
| SAT+LoRA | - | **93.1** | **88.2** | **61.7** |
| | No pre-training | 88.4 | 74.5 | 12.4 |

Table 7: Effect of various components of our method's variants. We report macro average F1 scores for each domain and use models with the same number of layers for each category of text as in the main text. *Only clean text* does not apply any corruptions. *No pre-training* skips the paragraph segmentation stage on web-scale mC4, and thus starts from XLM-R weights. Best per-category results are **bold**.

| Model | Variation | Lyrics Corrupted? | | Legal Corrupted? | |
|---|---|---|---|---|---|
| | | ✓ | ✗ | ✓ | ✗ |
| SAT+LoRA@100 | - | **60.8** | **62.4** | **81.1** | **93.6** |
| | No pre-training | 20.3 | 34.3 | 61.2 | 79.8 |
| SAT+LoRA | - | **68.5** | **73.8** | **83.3** | **95.1** |
| | No pre-training | 59.9 | 62.1 | 82.5 | 94.9 |

Table 8: Effect of the web-scale pre-training stage on adaptation to hard domains, averaged over genres/legal categories. @100 corresponds to using a maximum of 100 songs/documents per genre/category for adaptation.

**SAT components.** We show the effect of removing different components of our corruption schemes used in SAT and SAT+SM in Table 7. For SAT, only using clean text even slightly hurts performance on clean text and strongly degrades performance in our more noisy tweets and code-switching evaluations. A similar pattern occurs for SAT+SM: Only using clean text hurts performance. Moreover, skipping the web-scale pre-training stage (*No pre-training*) also decreases performance to a large extent, with the difference being particularly large for tweets and code-switching. Finally, for SAT+LoRA, *no pre-training* similarly degrades performance, with the difference being particularly large in code-switching, where SAT+LoRA is better by 49.3% absolute F1.

Moreover, Table 8 compares domain adaptation performances via LoRA to models without our web-scale pre-training to SAT models with it. As observed before, *no pre-training* markedly degrades performance in both lyrics and legal data. The difference is especially large in cases where only 100 songs or documents are available, clearly showing that our pre-training stage *improves sample-efficiency*.

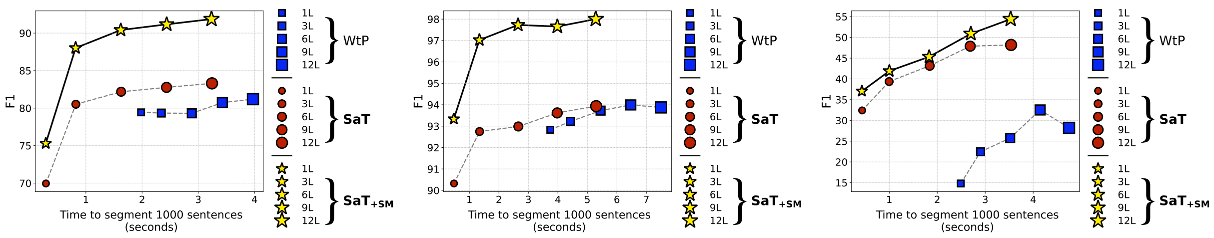
**Limited lookahead.** We further provide an ablation study on the effect of disabling the limited lookahead mechanism using sentence pairs in Table 9. Without limited lookahead, SAT is outperformed by WTP. On the contrary, with limited lookahead, SAT outperforms WTP by a considerable margin, where the difference is even more pronounced for 12-layer variations. Moreover, SAT+SM hardly benefits from limited lookahead, justifying our decision to disable it for SAT+SM.

**Effect of model size.** Figure 4 shows the effect of scaling up model sizes on OPUS100, UD, and code-switching, respectively. Remarkably, all 3-layer variations of SAT+SM clearly outperform WTP, despite not relying on language codes and being $\approx 5$x faster. The difference is particularly pronounced in code-switching, where even the 1-layer variations of both SAT and SAT+SM outperform the best variation of WTP. In general, performance continues to increase when further scaling up model sizes up to 12 layers.

| Model | Layers | Look-ahead | OPUS100 | UD | Ersatz | TED |
|---|---|---|---|---|---|---|
| WTP | 3 | ∞ | 52.4 | 80.6 | 78.0 | 9.8 |
| | 12 | ∞ | 52.8 | 77.9 | 78.0 | 12.6 |
| SAT | 3 | ∞ | 4.4 | 1.8 | 3.5 | 1.9 |
| | | 48 | 56.9 | 82.4 | 82.2 | 16.9 |
| | 12 | ∞ | 31.0 | 55.0 | 55.5 | 20.9 |
| | | 48 | **63.3** | **85.2** | **84.0** | **28.8** |
| SAT+SM | 3 | ∞ | 72.2 | 91.4 | 85.9 | 29.4 |
| | | 48 | 73.7 | 93.1 | 85.8 | 28.4 |
| | 12 | ∞ | 78.0 | 93.5 | **92.5** | **41.7** |
| | | 48 | **78.6** | **93.6** | 91.3 | 38.3 |

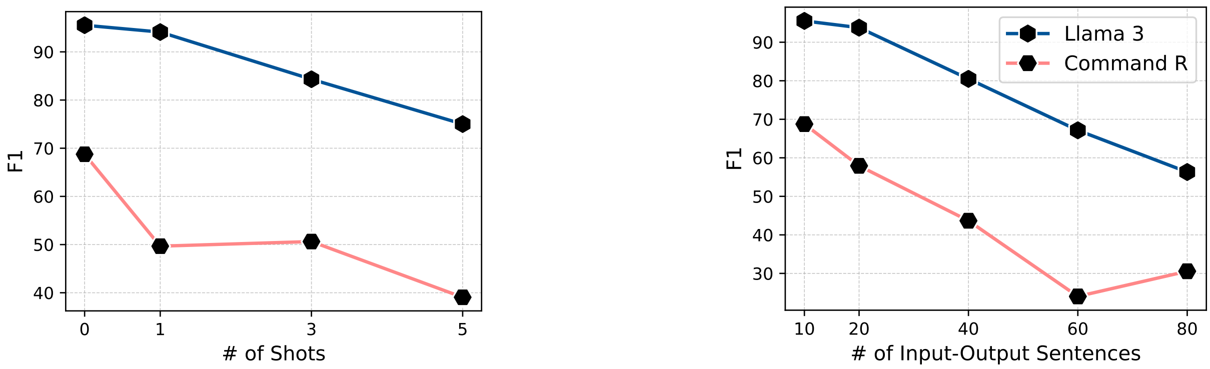
Table 9: Proportion of perfectly segmented sequences within additional corpora.

**LLMs.** Figure 5 shows the effect of few-shot prompting and varying the number of input-output sentences for LLAMA $3_{8B}$ and COMMAND R. Contrary to our expectations, in-context learning via few-shot prompting does not improve sentence segmentation performance for both LLMs in consideration. Providing only a single example already degrades performance, and providing more examples further degrades it. Furthermore, increasing the




(a) OPUS100 (b) UD (c) Code-Switching

Figure 4: F1 scores and inference time for the prior SoTA (WTP) and our models (SAT and SAT$_{+SM}$), evaluated on additional sentence segmentation benchmarks. We average over all 23 languages and show the average time (10 runs) for variants with different sizes (L = #layers) to segment 1,000 sentences using consumer hardware (1 Nvidia GTX 2080 Ti GPU). Performance on Ersatz is shown in Figure 1.




(a) Effect of few-shot prompting.

(b) Effect of varying the number of input-output sentences.

Figure 5: Ablation study on sentence segmentation performance of LLMs.

number of input-output sentences from our favorably low default of 10 results in considerable performance decreases. Notably, when using 80 input-output sentences per chunk, both LLMs achieve F1 scores of only $< 60\%$.

## A.2 Complete Experiment Details

| Language | Number of Sentences | | Source |
|---|---|---|---|
| | Train | Test | |
| sl | 2728 | 2728 | Fišer et al. (2020) |
| sr | 1727 | 192 | Miličević and Ljubešić (2016) |
| es-en | 1335 | 1334 | Deuchar (2009) |
| en-de | 678 | 599 | Osmelak and Wintner (2023) |
| tr-de | 578 | 805 | Çetinoğlu (2017) |
| vi-en | 1360 | 1361 | Nguyen and Bryant (2020) |

Table 10: Number of train and test sentences from tweets and code-switched text, including their source.

**Dataset details.** We give an overview of all used languages and their evaluation dataset sizes for clean text in Table 15. Furthermore, we provide statistics of splits for noisy text and additional domains in Table 10 for tweets and code-switching, Table 11 for lyrics, and Table 12 for legal data.

| Repetitiveness | Genre | Number of Songs | |
|---|---|---|---|
| | | Train | Test |
| High | Punk Rock | 778 | 190 |
| | Pop Punk | 512 | 141 |
| | Country | 2916 | 711 |
| Mid | Rock | 4611 | 1182 |
| | Pop | 3490 | 891 |
| | RnB | 3542 | 915 |
| | Alternative Rock | 3370 | 856 |
| | Alternative Metal | 651 | 155 |
| | Soul | 494 | 110 |
| | Hard Rock | 1821 | 430 |
| | Indie Rock | 1193 | 305 |
| | Pop Rock | 1633 | 412 |
| | Heavy Metal | 988 | 216 |
| | Indie Rock | 1193 | 305 |
| Low | Southern Hip Hop | 836 | 208 |
| | Gangsta Rap | 270 | 64 |

Table 11: Number of train and test songs per genre.

If a given corpus does not provide train and test splits, we set aside 10,000 sentences for testing and keep the rest for training if more than 10,000 sentences are available. If a corpus is smaller, we set aside 50% for testing and use the rest for adaptation. To train SAT$_{+SM}$, if neither UD nor OPUS100 train data is available, we resort to NLLB. This is the case in Cebuano (`ceb`), Javanese (`jv`), Mongolian (`mn`), and Yoruba (`yo`), where we take 10,000

| Language | Number of Documents | | | |
|---|---|---|---|---|
| | Laws | | Judgements | |
| | Train | Test | Train | Test |
| de | 10 | 3 | 104 | 27 |
| en | - | - | 64 | 16 |
| es | 494 | 183 | 151 | 39 |
| fr | 1672 | 459 | 252 | 63 |
| it | 2206 | 704 | 194 | 49 |

Table 12: Number of legal train and test documents per category. We discard Portuguese since there is no training data available.

sentences each.

To simulate our real-time automatic speech recognition (ASR) scenario, we take publicly available TED talk transcripts in 76 languages, available at `opus.nlpl.eu/TED2020/corpus/version/TED2020`. We generate non-overlapping sentence pairs as done in other experiments to evaluate performance on short sequences. Additionally, we remove fully lowercase all pairs and all punctuation tokens. We generally derive punctuation tokens with the commonly used Moses tokenizer (Koehn et al., 2007) for languages where it is available. For all other languages, we simply remove all punctuation characters.

Moreover, we observe that used tweets in `sl` and `sr` are inconsistent w.r.t. segmenting single emojis. We thus filter out all emojis as a simple preprocessing step. Similarly, we normalize tweets by filtering out words starting with *http*, *#*, and *@*.

**Computing infrastructure.** We train SAT on a TPUv4 VM with 8 cores, $\text{SAT}_{\text{+LoRA}}$ on a TPUv3 VM with 1 core, and $\text{SAT}_{\text{+SM}}$ using a single A100 GPU. To measure inference time, we use a consumer-grade GPU, the Nvidia GTX 2080 Ti with an AMD EPYC 7402P CPU.

**Implementation details.** We use the `PyTorch` (Paszke et al., 2019) and `transformers` (Wolf et al., 2020) libraries for all experiments. For adaptation via LoRA, we make use of the `adapters` library (Poth et al., 2023; Pfeiffer et al., 2020) library, a wrapper around the `transformers` library. Our code and models are released under the MIT License, ensuring open access to the community for further development.

**Training of SAT.** We train SAT using a context window of 256 since we observed that it improves performance. During inference, we use the full context size of XLM-R, 512. Moreover, we follow Minixhofer et al. (2023) and sample paragraphs to ensure that a maximum of 10% of paragraphs do not end in punctuation (except for Thai, which does not use sentence-ending punctuation). We also sample paragraphs of languages uniformly. We continue training XLM-R on naturally occurring newline symbols for 200k training steps using a batch size of 512. We use a linearly increasing learning rate warmup from 0 to 1e-4, and decay the learning rate to 0 for the remaining 195k steps. We use the AdamW optimizer (Kingma and Ba, 2015). For the auxiliary objective as introduced by Minixhofer et al. (2023), we set the removal probability $p = 0.25$ using the union of the 30 most common punctuation characters in every language. We then take the corresponding tokens as used by XLM-R as labels for the auxiliary objective.

For models without limited lookahead (cf. Table 9), we follow Minixhofer et al. (2023) and use a threshold of 0.01 for sentence boundary detection. When using limited lookahead, we observe that the optimal threshold increases. We thus use a constant threshold of 0.025 with limited lookahead.

**Training of $\text{SAT}_{\text{+SM}}$.** In our supervised mixture stage, we continue training SAT using the same context window of 256. The training data now consists of sentence-segmented text, and we train $\text{SAT}_{\text{+SM}}$ predicting sentence-ending tokens.

For each language, we corrupt the data in two ways. In the first, we lowercase and remove all punctuation tokens. This aims to roughly emulate the output of an automatic speech recognition (ASR) system. In the second, we lowercase all text with probability 0.5, remove all punctuation with probability 0.5, duplicate punctuation (e.g., changing ! to !!!) with geometric distribution scaling with the number of duplications (i.e., doubling with probability 0.5, tripling with probability 0.25, etc.), and join sentences without a whitespace with probability 0.1 (or with a space for the four languages which do not generally use a whitespace to split sentences, see Table 1.) This aims to emulate user-generated text.

We generally pack sentences into chunks. For the uncorrupted sentences and sentences corrupted with the first scheme, we pack until each chunk fully fills up the model's context window. For our second corruption scheme, we include $s$ sentences in each block, where $s$ is drawn from the same geometric distribution as used before.

We train with a batch size of 128, linear learning rate warmup from 0 to 3e-5 for 500 steps, and linearly decay for another 19,500 steps. We uniformly sample batches of sentences from a single language and evenly sample batches corrupted with one of the three corruption schemes. During inference, we use a constant threshold of $0.25$.

**Training of SaT$_{\text{+LoRA}}$.** For adaptation via LoRA, we use a learning rate of 3e-4. We train LoRA modules with AdamW for a target domain for 30 epochs, where we linearly warm up the learning rate for the first 10% of training, followed by a decay to 0 for the remaining training steps. We do not apply early stopping. We apply LoRA to the query and value matrices of the attention block, as well as the intermediate layer of the Transformer, using a rank $r = 16$ and scaling factor $a = 32$. We noticed this to positively impact performance at a comparably low computational cost. Moreover, similarly to WtP$_{\text{T}}$ and WtP$_{\text{PUNCT}}$, we additionally tune the classification threshold on the same training data if more than $512$ *sentences* are available. We noticed that this helps performance in such cases. For verse segmentation, we use SaT models without limited lookahead, since, with verses, it is both helpful and desirable to rely on future verses.

**LLM details.** We use default hyperparameters for both Llama 3$_{\text{8B}}$ and Command R. For Command R, We used the Cohere API. This led to some API refusals, particularly for lyrics. We thus only consider chunks that were not refused when calculating metrics. To align input and output chunks using the Needleman-Wunsch sequence alignment algorithm, we use a gap penalty of $-0.5$, a gap extension penalty of $-0.5$, a match reward of $1$, and a mismatch penalty of $-0.5$. If no alignment is found, the LLM produced output that strongly deviated from the input chunk. We thus assign no sentence boundaries to the predictions of the LLMs for this input chunk.

We experiment with several prompts, optimizing performance on the training set, resulting in the following final prompt:

**General LLM Prompt**

> Separate the following text into sentences by adding a newline between each sentence. Do not modify the text in any way and keep the exact ordering of words! If you modify it, remove or add anything, you get fined $1000 per word. Provide a concise answer without any introduction. Indicate sentence boundaries only via a single newline, no more than this!

We then append *\n\n# Input: \n\n*, followed by the input chunk, followed by *\n\n# Output: \n\n*, resulting in the complete input to the LLM.

For few-shot prompting, we append the prompt with *When provided with multiple examples, you are to respond only to the last one*. In addition, we indicate chunk $n$ with *Input N:* and *Output N:* , respectively.

Since it is a highly distinct task, we use the following prompt for verse segmentation:

**LLM Lyrics Prompt**

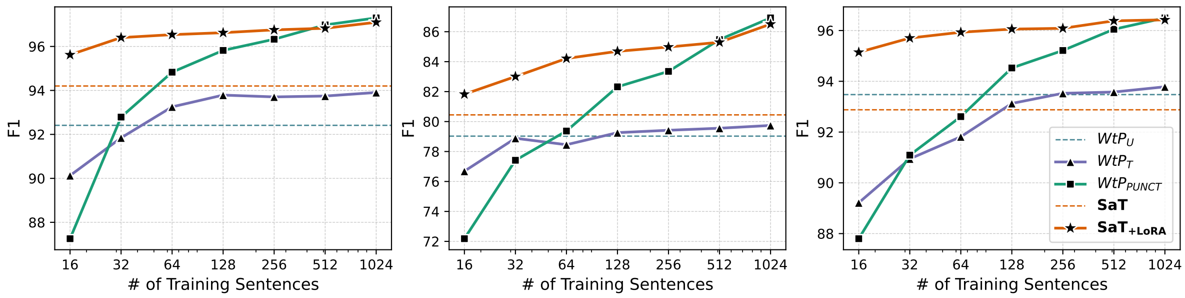
> Separate the following song's lyrics into semantic units (e.g., verse, chorus, bridge, intro/outro, etc - similar to how they are presented in a lyrics booklet) via double newlines, but do not annotate them. Only include the song in the output, no annotations. Do not modify the song in any way and keep the exact ordering of words! If you modify it, remove or add anything, you get fined $1000 per word. Indicate semantic units by double newlines.

### A.3 Qualitative Examples

**ASR output.** We show predictions of SaT$_{\text{+SM}}$ on parts of transcribed TED talks in different languages in Table 16.

**Code-switching.** We also show predictions of SaT$_{\text{+SM}}$ on code-switching text in four language pairs in Table 17.

**Verse segmentation.** In addition, we show predictions of SaT$_{\text{+LoRA}}$ on verse segmentation in Tables 18, 19, and 20 for songs of high, mid, and low levels of repetitiveness, respectively.




(a) Ersatz (b) OPUS100 (c) UD

Figure 6: Avg. F1 vs. number of sentences used for adaptation, averaged over languages within a given dataset.

| Model | OPUS100 | UD | Ersatz | Macro Avg. |
|---|---|---|---|---|
| SPACY$_{\text{M}}$ | 88.8 | 91.7 | 94.0 | 91.5 |
| ERSATZ | 87.6 | 89.1 | 97.5 | 91.4 |
| LLAMA 3$_{\text{8B}}$ | 92.8 | 94.8 | <u>98.2</u> | 95.3 |
| COMMAND R | 89.5 | 77.1 | 87.2 | 84.6 |
| SAT | 90.4 | 93.9 | 96.7 | 93.7 |
| SAT$_{\text{+SM}}$ | <u>**94.6**</u> | <u>**96.7**</u> | <u>**98.3**</u> | <u>**96.5**</u> |
| NLTK | 88.2 | 90.8 | 97.7 | 92.2 |
| PYSBD | 59.6 | 75.3 | 73.9 | 69.6 |
| SPACY$_{\text{DP}}$ | 89.0 | 91.3 | 98.5 | 92.9 |
| WTP | 90.6 | 94.5 | 96.5 | 93.9 |
| WTP$_{\text{T}}$ | 89.4 | 94.5 | 96.7 | 93.5 |
| WTP$_{\text{PUNCT}}$ | <u>94.7</u> | <u>**96.9**</u> | <u>98.6</u> | <u>96.7</u> |
| SAT$_{\text{+LoRA}}$ | <u>**94.8**</u> | <u>96.8</u> | <u>**98.7**</u> | <u>**96.8**</u> |

Table 13: English (en) sentence segmentation F1 scores. We use 3-layer versions of each model. Numerically best results are in **bold**, statistically indistinguishable ones from this best are <u>underlined</u>.

## A.4 Additional Results

**Results in English.** We provide an overview of the performance of different models on different corpora in Table 13.

**Legal data.** Table 21 shows the sentence segmentation performance of different models on MultiLegalSBD. Furthermore, Table 22 shows performance on MultiLegalSBD when applying the same corruptions as on Speeches, removing all casing and punctuation tokens.

**More few-shot results.** Figure 6 shows the per-dataset few-shot domain adaptation results, comparing WTP$_{\text{T}}$, WTP$_{\text{PUNCT}}$, and SAT$_{\text{+LoRA}}$.

**Effect of stride.** For both SAT and WTP, we use a default stride of 64 during evaluation. Each subword or character is thus processed multiple times, where we average predictions for overlapping positions. Since SAT operates on subwords but WTP on characters, this results in different scaling behaviors, illustrated in Figure 7.

| Model | es en | de en | tr de | vi en | Macro Avg. |
|---|---|---|---|---|---|
| NLTK | 0.0/0.0 | 0.0/1.1 | 0.0/0.0 | -/0.0 | 0.3/- |
| PYSBD | 0.0/0.0 | 2.1/2.1 | -/0.0 | -/0.0 | 0.5/0.5 |
| SPACY$_{\text{DP}}$ | 0.0/17.6 | 8.6/8.0 | -/12.2 | -/11.3 | 12.2/- |
| WTP | 20.8/38.6 | 39.0/31.4 | 33.5/21.0 | 22.7/25.5 | 29.1/29.0 |
| WTP$_{\text{T}}$ | 46.9/52.2 | 45.7/39.1 | 33.3/34.4 | 46.7/36.7 | 40.6/43.2 |
| WTP$_{\text{PUNCT}}$ | 60.7/62.1 | 60.1/58.1 | 39.9/41.0 | 59.0/50.7 | 53.0/54.9 |

Table 14: Complete sentence segmentation F1 scores for code-switched text for systems relying on language codes, where the first number corresponds to the first language shown.

**Complete verse segmentation results.** We provide complete per-genre results for verse segmentation in Tables 23 and 24. Furthermore, Tables 25 and 26. show verse segmentation performance when applying the same corruptions as on Speeches, removing all casing and punctuation tokens.

**Complete results on clean data.** Results of SAT, its variations, and other methods on all languages are shown in Tables 27-32.

**Complete code-switching result.** We show results using both language codes on code-switched text for models using language codes in Table 14.

## A.5 Auxiliary Punctuation Prediction Objective

As mentioned in Section 3, we adopt the auxiliary punctuation prediction objective from WtP (Minixhofer et al., 2023) as our base corruption scheme.

For clarity, we first specify the target without the auxiliary objective. Here, the original sequence of tokens $\boldsymbol{c}$ within some corpus is first stripped of newline characters:

$$\boldsymbol{x} = \{c_i \mid c_i \in \boldsymbol{c}, c_i \neq \texttt{\textbackslash n}\}. \quad (1)$$

We then create labels, which we set positive if the following token in the original sequence is a newline character:

$$\boldsymbol{y} = \left\{ \begin{cases} 1 & \text{if } c_{i+1} = \texttt{\textbackslash n} \\ 0 & \text{otherwise} \end{cases} \mid c_i \in \boldsymbol{x} \right\}, \tag{2}$$

where $c_i$ indexes into the original sequence $\boldsymbol{c}$. Using these labels, we optimize the standard cross-entropy of these labels and the model's predictions. Note that the newline character is not contained in our base model's vocabulary and will thus only appear as a single character. We also tokenize the whole batch at the start before applying any corruptions and do not re-tokenize later. We found this to be more effective than re-tokenizing the sequence after applying corruptions.

**Auxiliary Punctuation Prediction.** For the auxiliary objective, we adapt the methodology from Minixhofer et al. (2023) to tokens and identify the union of the 30 most common punctuation-only *tokens* within the training set. For simplicity, we ignore tokens containing multiple (potentially non-punctuation) characters. We also include the <UNK> token in this resulting set $P$, resulting in 109 punctuation tokens. We then define a random binary mask that determines which punctuation characters to remove among $P$, resulting in the new sequence $\boldsymbol{x}'$:

$$\boldsymbol{x}' = \left\{ c_i \mid \begin{array}{l} c_i \in \boldsymbol{c}, c_i \neq \texttt{\textbackslash n}, \\ c_i \notin P \text{ or } p_i = 0 \end{array} \right\} \tag{3}$$

Here, we do not remove two consecutive character tokens to be able to reconstruct the original sequence. In addition, unlike WtP, we only remove character tokens if the following token is not a newline token. For the remaining characters, the auxiliary labels $\boldsymbol{z}$ indicate which (if any) character among $P$ followed them in the original sequence:

$$\boldsymbol{z} = \left\{ \begin{cases} c_{i+1} & \text{if } c_{i+1} \in P \\ 0 & \text{otherwise} \end{cases} \mid c_i \in \boldsymbol{x}' \right\} \tag{4}$$

To avoid needing two separate forward passes through the model, we substitute the input $\boldsymbol{x}$ with $\boldsymbol{x}'$ also for the main (newline prediction) objective. The final loss $\mathcal{L}$ is obtained by summing up the main newline prediction objective and the auxiliary objective of predicting punctuation:

$$\mathcal{L} = \mathcal{L}^{\text{main}} + \mathcal{L}^{\text{aux}} \tag{5}$$

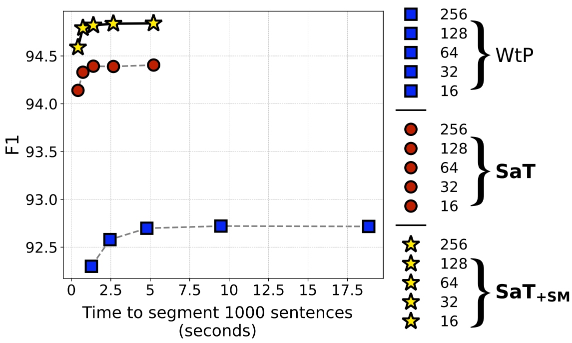



Figure 7: Sentence segmentation F1 scores vs. execution time across different strides (default 64), evaluated on Ersatz. We use the standard 3-layer variants of each model. Higher `stride` values result in faster inference.

| Language | iso | Space | UD | OPUS100 | Ersatz | Speeches | Language | iso | Space | UD | OPUS100 | Ersatz | Speeches |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Afrikaans | af | | AfriBooms (425) | 1.9k | | 1.3k | Kurdish | ku | | | 1.9k | | 10.0k |
| Amharic | am | | | 2.0k | | 514 | Kirghiz | ky | | | 1.7k | | 3.0k |
| Arabic | ar | | PADT (680) | 2.0k | 1.5k | 10.0k | Latin | la | | ITTB (2.1k) | | | 10.0k |
| Azerbaijani | az | | | 2.0k | | 6.8k | Lithuanian | lt | | ALKSNIS (684) | 2.0k | 1.0k | 2.1k |
| Belarusian | be | | HSE (1.1k) | 2.0k | | 6.2k | Latvian | lv | | LVTB (2.3k) | 2.0k | 2.0k | 10.0k |
| Bulgarian | bg | | BTB (1.1k) | 2.0k | | 10.0k | Malagasy | mg | | | 2.0k | | 103 |
| Bengali | bn | | BRU (56) | 2.0k | | 5.2k | Macedonian | mk | | | 2.0k | | 10.0k |
| Catalan | ca | | AnCora (1.8k) | 2.0k | | 10.0k | Malayalam | ml | | | 2.0k | | 2.1k |
| Cebuano | ceb | | GJA (188) | | | 55 | Mongolian | mn | | | 4.2k | | 5.5k |
| Czech | cs | | PDT (10.1k) | 2.0k | 1.7k | 10.0k | Marathi | mr | | UFAL (47) | 2.0k | | 10.0k |
| Welsh | cy | | CCG (953) | 1.8k | | - | Malay | ms | | | 1.9k | | 2.1k |
| Danish | da | | DDT (565) | 2.0k | | 10.0k | Maltese | mt | | MUDT (518) | 2.0k | | 10.0k |
| German | de | | GSD (977) | 1.9k | 2.0k | 10.0k | Burmese | my | ✗ | | 2.0k | | 5.5k |
| Greek | el | | GDT (456) | 2.0k | | 10.0k | Nepalese | ne | | | 1.9k | | 6.7k |
| English | en | | GUM (1.1k) | 2.0k | 7.7k | 10.0k | Dutch | nl | | Alpino (596) | 2.0k | | 10.0k |
| Esperanto | eo | | | 2.0k | | 10.0k | Norwegian | no | | Bokmaal (1.9k) | 2.0k | | 2.1k |
| Spanish | es | | AnCora (1.7k) | 2.0k | 3.1k | 10.0k | Panjabi | pa | | | 2.0k | | 3.8k |
| Estonian | et | | EDT (3.2k) | 2.0k | 2.0k | 5.9k | Polish | pl | | PDB (2.2k) | 2.0k | 1.0k | 548 |
| Basque | eu | | BDT (1.8k) | 2.0k | | 10.0k | Pushto | ps | | | 1.8k | 2.7k | 10.0k |
| Persian | fa | | PerDT (1.5k) | 2.0k | | 10.0k | Portuguese | pt | | Bosque (1.2k) | 2.0k | | 5.5k |
| Finnish | fi | | TDT (1.6k) | 2.0k | 2.0k | 10.0k | Romanian | ro | | Nonstandard (1.1k) | 2.0k | 2.0k | 10.0k |
| French | fr | | GSD (416) | 2.0k | 1.7k | - | Russian | ru | | Taiga (881) | 2.0k | 991 | 10.0k |
| Western Frisian | fy | | | 1.9k | | 46 | Sinhala | si | | | 2.0k | | 516 |
| Irish | ga | | IDT (454) | 2.0k | | - | Slovak | sk | | SNK (1.1k) | 2.0k | | 10.0k |
| Scottish Gaelic | gd | | ARCOSG (545) | 1.1k | | 10.0k | Slovenian | sl | | SSJ (1.3k) | 2.0k | | 10.0k |
| Galician | gl | | TreeGal (400) | 2.0k | | 7.9k | Albanian | sq | | TSA (60) | 2.0k | | 10.0k |
| Gujarati | gu | | | 1.9k | 1.0k | 13 | Serbian | sr | | SET (520) | 2.0k | | 5.6k |
| Hausa | ha | | | 2.0k | | 10.0k | Swedish | sv | | LinES (1.0k) | 2.0k | | 2.6k |
| Hebrew | he | | IAHLTwiki (393) | 2.0k | | 10.0k | Tamil | ta | | TTB (120) | 2.0k | 1.0k | 3.0k |
| Hindi | hi | | HDTB (1.7k) | 2.0k | 2.5k | 10.0k | Telugu | te | | | 2.0k | | 303 |
| Hungarian | hu | | Szeged (449) | 2.0k | | 10.0k | Tajik | tg | | | 2.0k | | 10.0k |
| Armenian | hy | | BSUT (595) | 7.0k | | 104 | Thai | th | | PUD (1.0k) | 2.0k | | 7.8k |
| Indonesian | id | | PUD (1.0k) | 2.0k | | 1.6k | Turkish | tr | | IMST (983) | 2.0k | 3.0k | 3.8k |
| Igbo | ig | | | 1.7k | | 10.0k | Ukrainian | uk | | IU (892) | 2.0k | | 10.0k |
| Icelandic | is | | IcePaHC (5.2k) | 2.0k | | 10.0k | Urdu | ur | | UDTB (535) | 1.9k | | 7.8k |
| Italian | it | | ISDT (482) | 2.0k | | 5.8k | Uzbek | uz | | | 2.0k | | 3.8k |
| Japanese | ja | ✗ | GSD (543) | 2.0k | 1.1k | 533 | Vietnamese | vi | | VTB (800) | 1.9k | | 10.0k |
| Javanese | jv | | CSUI (125) | | | 1.1k | Xhosa | xh | | | 1.9k | | 5.6k |
| Georgian | ka | | | 2.0k | | 10.0k | Yiddish | yi | | | 1.3k | | 2.6k |
| Kazakh | kk | | KTB (1.0k) | 1.9k | 1.0k | 4.1k | Yoruba | yo | | YTB (318) | 9.4k | | 10.0k |
| Khmer | km | ✗ | | 1.9k | 2.4k | 12 | Chinese | zh | ✗ | GSDSimp (500) | 2.0k | 2.0k | 1.7k |
| Kannada | kn | | | 906 | | 10.0k | Zulu | zu | | | 1.9k | | 8.1k |
| Korean | ko | | Kaist (2.3k) | 2.0k | | 215 | | | | | | | |

Table 15: List of the 85 languages considered, whether they generally use whitespace to split sentences, and the corresponding evaluation dataset size, measured in sentences. For UD, we use UDv2.13, where the treebank name used is also shown. We use *Speeches* only in pairwise evaluations.

| | |
|---|---|
| English (en) | we use science to create something wonderful<br>we use story and artistic touch to get us to a place of wonder<br>this guy wall-e is a great example of that<br>he finds beauty in the simplest things<br>but when he came in to lighting we knew we had a big problem<br>we got so geeked-out on making wall-e this convincing robot that we made his binoculars practically optically perfect<br>laughter<br>(*) his binoculars are one of the most critical acting devices he has<br>he doesn't have a face or even traditional dialogue for that matter<br>so the animators were heavily dependent on the binoculars to sell his acting and emotions<br>we started lighting and we realized the triple lenses inside his binoculars were a mess of reflections<br>he was starting to look glassy-eyed |
| German (de) | aber ich habe auch das marfan-syndrom<br>das ist eine erbkrankheit<br>1992 nahm ich an einer genetikstudie teil<br>zu meinem entsetzen erfuhr ich dass wie sie hier sehen meine aorta ascendens nicht im normalbereich war die grüne linie hier unten<br>alle hier im raum werden bei 3,2 und 3,6 cm liegen<br>ich war bereits bei 4,4<br>wie sie sehen können erweiterte sich meine aorta zunehmend und allmählich geriet ich an den punkt dass eine operation nötig sein würde<br>die angebotene operation war ziemlich gruselig<br>anästhesie öffnen des brustkorbs man hängt sie an eine künstliche herzlungenmaschine lässt ihre körpertemperatur auf etwa 18 grad fallen hält ihr herz an schneidet die aorta raus ersetzt sie mit einer klappe<br>und aorta aus plastik<br>und am wichtigsten verdonnert sie lebenslang zu antikoagulationstherapie<br>(*) normalerweise mit warfarin<br>der gedanke an diese operation war nicht gerade ansprechend |
| French (fr) | j'ai montré mon intro et j'ai mis la scène de la méduse<br>le réalisateur est resté silencieux pendant un très long moment (\|) assez long pour que je puisse me dire oh non c'est foutu<br>et il a commencé à applaudir<br>puis le concepteur artistique<br>(*) et finalement toute la salle<br>c'est pour ces moments que je fais ce travail<br>le moment où tout fait sens et où l'on crée un monde auquel on peut croire<br>on utilise la science et la programmation pour créer ces mondes incroyables<br>on utilise les histoires et l'art pour leur donner vie<br>c'est la coexistence de l'art et la science qui transforme le monde en un lieu magique un lieu avec une âme un lieu auquel on peut croire un lieu où les choses qu'on imagine deviennent réelles – et un monde où tout d'un coup une fille réalise qu'elle n'est pas seulement une scientifique mais aussi une artiste<br>merci (*) applaudissements |
| Italian (it) | cosa state leggendo<br>beau lotto<br>(*) cosa state leggendo<br>mancano metà delle lettere<br>giusto<br>non c'è nessuna ragione a priori perché una h debba comparire tra la w e la a<br>ma ne collocate una lì<br>perché<br>perché nella statistica della vostra esperienza passata sarebbe stato utile fare così<br>quindi lo fate di nuovo<br>e tuttavia non collocate una lettera dopo quella prima t<br>perché (\|) perché non si sarebbe dimostrato utile nel passato<br>quindi non lo fate di nuovo |

Table 16: Examples of predictions of $\text{SAT}_{+\text{SM}}$ taken from random positions from transcribed TED talks in four langauges. (|) marks a missing sentence boundary (false negative), and (*) marks a wrongly inserted sentence boundary (false positive). All others are correctly segmented, according to the ground truth segmentation.

| | |
|---|---|
| German-English (de-en) | its about the rundfunkstaatsvertrag and the licence you need to stream to more than 500 viewers<br>just go to the amt on your day off arrive at 9 and bring all the papers and a book<br>hat ihr kein editor angekreidet<br>guess op thought everyone pays freiwillige gesetzliche krankenversicherung und pflegeversicherung which is around 780eurs month per person family depends (\|) is the date that matters<br>the bescheinigungszeitraum on my ausdruck der elektronischen lohnsteuerbescheinigung für 2013 or my first anmeldung<br>na anti-establishment anti-kapitalismus oder generell anti-zwang<br>(*) zum beispiel (\|) ich bin software engineer mit data-management data-security background und er is front-end mobile dev<br>noch besser in flughäfen in england kommt keine höflich message wie achten sie bitte auf ihr eigenes gepäck... sondern direkt eine drohung for security reasons baggage left unattended will be removed and destroyed<br>da bleibt irgendwie nicht mehr viel übrig |
| Spanish-English (es-en) | in the morning over there cada vez que yo decía algo él me decía algo<br>the best thing about her ella no complain you know (\|) tiene she has a great personality<br>hasta que tú pushed the wrong button<br>linda lópez la testing<br>ay but she's cool<br>el teniente y ella han tenido tú sabes conflicts<br>entonces tina es the computer person<br>so ella ella es la jefa de linda<br>sí i i have a room<br>no<br>(*) pero tal vez consigue un roommate<br>un roommate (\|) mandó un e-mail diciendo que le había que había otra persona en la dirección en lugar de ella<br>yo me dio a entender según como leí yo que era ella e edith |
| Turkish-German (tr-de) | ja bence ich probiere es einfach<br>ja vor allem sınav olduğu zaman muss ja muss ja schon so sein<br>geçen sene ich weiß noch eh sınavların olduğu günlerde (\|) ja tam denk geldi weißt du (\|) die woche noch ich denke mir so tutayım mı tutmayayım mı kann gar nicht mehr<br>ben tutmuştum bir tanesinde<br>(*) und ich dachte so tamam bittin sen die<br>(*) hani das ist nicht gut gegangen<br>(*) die prüfung das war auch so<br>(*) ich konnte mich gar konzentrieren überhaupt nicht<br>hani yemek de değil<br>(*) weißt du einfach nur wasser<br>ja bir de o geçen sene da war es so heiß |
| Vietnamese-English (vi-en) | bởi vậy lux đường có overconfident (\|) ai cũng cần phải improve (\|) nên lux phải phải phải đưa cho chị ti với anh alex check nha (\|) thì design đến đâu rồi lux (\|) cái graphic design của lux<br>(*) lux design được đến đâu rồi come up with idea để ghi ra<br>(*) chị ti sẽ cùng help lux to write down<br>(*) hoặc là ở woden qua đây ăn dinner với chị ti<br>get it out of the way là done với một cái đó rồi là xong<br>thái writing cũng đâu có good đâu<br>deadline như vậy được chưa<br>vậy là tối mai lux biết spend time write it tomorrow (\|) được rồi ta design như vậy nè<br>mọi thứ là lux phải plan ahead như vậy chứ |

Table 17: Examples of predictions of SAT$_{+SM}$ taken from random positions from code-switching text in four language pairs. (|) marks a missing sentence boundary (false negative), and (*) marks a wrongly inserted sentence boundary (false positive). All others are correctly segmented according to the ground truth segmentation. There are many ambiguous sentence boundaries in these corpora.

| **Original (non-corrupted) song** | **Corrupted song** |
|---|---|
| Have yourself a merry little Christmas<br>Let your hearts be light<br>From now on<br>Our troubles will be out of sight | i'd never leave the perfect girl<br>or rip apart the perfect world<br>just up and leave in the middle of a song |
| Have yourself a merry little Christmas<br>Make the Yule-tide gay<br>From now on<br>Our troubles will be miles away. | i'd never pack my things in a silverado<br>drive on out to colorado<br>just to find some freedom i thought was gone<br>(\|) ooooh<br>there are things i'd never do |
| Here we are as in olden days<br>Happy golden days of yore<br>Faithful friends who are dear to us<br>Will be near to us once more | but here i am in this hotel room<br>thinking bout you and what i 've done<br>oh what have i done<br>head in my hands<br>thinking about a lot of things<br>i wish that i could change<br>it's sad but it's true<br>i 've done a lot of things i'd never do |
| Through the years We all will be together<br>If the Fates allow<br>Until then we'll have to muttle through somehow<br>So have yourself a merry little Christmas now | i'd never ever work so much<br>that i'd lose sight i'd lose touch<br>of everything a man could ever want |
| Here we are as in olden days<br>Happy golden days of yore<br>Faithful friends who are dear to us<br>Will be near to us once more | i'd never lose my cool and say<br>those words that cut just like a blade<br>and leave you dying crying all alone |
| Through the years We all will be together<br>If the Fates allow<br>Until then we'll have to muttle through somehow<br>So have yourself a merry little Christmas now | i'd never leave the perfect girl<br>or rip apart the perfect world<br>just up and leave in the middle of a song |
| Have yourself a merry little Christmas now<br>Merry Christmas | |

Table 18: Examples of predictions of $\text{SAT}_{+\text{LoRA}}$ taken from songs categorized as *Country* (*High Repetitiveness*). (|) marks a missing verse boundary (false negative), and (*) marks a wrongly inserted verse boundary (false positive). All others are correctly segmented according to the ground truth segmentation. While the task was only to segment songs into *verses* and no line segmentation was provided, we format songs using both lines and verses for clarity.

| **Original (non-corrupted) song** | **Corrupted song** |
|---|---|
| Hope, a new beginning<br>Time, time to start living<br>Just like just before we died | lock me up inside my room<br>leave me without toys and food<br>keep that monster in my bed<br>just remember i'm not dead |
| There's no going back to the place we started from | you forget my memory lives way beyond these walls<br>you forget my indecision's taking all control |
| Hurt, falling through fingers<br>Trust, trust in the feeling<br>There's something left inside | once in a far land i grabbed you<br>and you woke me up to my origin |
| There's no going back to the place we started from<br>All secrets known | people have to understand my innocence has gone<br>(\|) go beyond my urge<br>or make an effort to<br>living on my own plagued by images |
| Calm, old wounds are healing<br>Strong, truth is worth saving<br>I want to feel alive | once in a far land i grabbed you<br>and you woke me up to my origin |
| There's no going back to the place we started from<br>All secrets known | |

Table 19: Examples of predictions of $\text{SAT}_{+\text{LoRA}}$ from songs categorized as *Alternative Metal* (*Mid Repetitiveness*).

| Original (non-corrupted) song | Corrupted song |
|---|---|
| | zaytoven on the track |
| | (l) zay-tiggy |
| | gucci |
| | gucci |
| | |
| | so watch entertainment |
| | lets go |
| Let me chirp these fools | |
| | |
| Juice got weed Juice got pills | they call me chef-boy-r.g. |
| Juice got the work on the corner cutting deals | but hold that thought |
| Juice know you haters out there snitching ain't for real | its a kodak moment |
| So Juice got some gang niggas down for the kill | but hold that thought |
| Juice know the feds got surveillance on the field | hurricane wrist game |
| We never had a job but we sitting on a mill | turn that junk off |
| We ball out in the club wit our niggas staying trill | hot as piggly wiggly |
| We never wrote a check just them big face bills | cant kermit the frog dog |
| A player drinking Makers Marka, cranberry vodka | |
| Wearing a mink coat thats furry as Chewbacca | |
| I saw ya main girl and a player had to stop her | early in the mornin |
| Her name wasn't Silkk but her face was The Shocker | i aint even yawnin |
| The feds taking pictures of us balling but I got 'em | cookin up a cake |
| A 7 footer hole for his body we gonna drop 'em | like i'm doin a performance |
| We always on the grind we be watching when they watching | when it come to flossin |
| And when they turn they back its the clucka-clucka-rock 'em yeah! | i aint even talkin |
| | diamonds on my joint |
| If you boys got beef we can (roll wit it) | got my chevy moonwalkin |
| In the club or the street we can (go wit it) | 10 bricks on my bart simpson just look |
| It don't make me none (blow for blow wit it) | my watch 35 pounds of kush |
| Crack his head wit a gun (I'ma sho split it) | my ring 36 oz's my nig |
| | my bracelet 500 lbs of mid |
| We got them tones in the club and them bulletproof vests | a gucci wrapped tour bus |
| Them three fifty seven titanium Smith-N-Wess | yall hoes follow us |
| And plus we deep as hell and prepared to bust | party pack pills man |
| You gonna have hell if you fuck wit us and thats whats up | hoes gonna swallow us |
| | naturally a loner |
| (*) The whole club we maintain | but love my kid |
| These hydrashock bullets mushroom in ya brain | mix the soda with the cola |
| We in bed with the med we give 'em something to do | i can buy me a friend |
| Cause clown ass niggas love to act a fool | new swag somethin like |
| | trap house times 10 |
| My hood is real nigga my hood ain't fake | ery nigga round me |
| My hood is home nigga everything straight | bust heads, ya-dig |
| My hood will rob you with mask on they face | iced out grill |
| My hood will do it to put food on they plate | i can't buy that bullshit |
| My hood ain't tame dog they wanna jump fool | i'm wit some street shit, |
| My hood they hang together they all jump you | like a reverend in the pulpit |
| And if you don't believe me then come to my hood | |
| And you will see that it ain't all good | they call me chef-boy-r.g. |
| | but hold that thought |
| | its a kodak moment |
| | but hold that thought |
| | hurricane wrist game |
| | turn that junk off |
| | hot as piggly wiggly |
| | cant kermit the frog dog |
| | ... |

Table 20: Examples of predictions of $\text{SAT}_{+\text{LoRA}}$ from songs categorized as *Southern Hip Hop* (*Low Repetitiveness*).

| Model | fr | | es | | it | | en | de | | Macro Avg. |
|---|---|---|---|---|---|---|---|---|---|---|
| | Judg. | Laws | Judg. | Laws | Judg. | Laws | Judg. | Judg. | Laws | |
| SPACY$_{M}$ | 82.7 | 61.2 | 67.7 | 85.2 | 73.6 | 53.8 | 81.4 | 65.8 | 67.9 | 71.0 |
| ERSATZ | 81.5 | 51.7 | 63.6 | 81.9 | - | - | 79.5 | 59.8 | 68.5 | 69.5 |
| LLAMA 3$_{8B}$ | 85.5 | 52.7 | 70.4 | 66.5 | 81.1 | 77.2 | 89.2 | 78.3 | 83.4 | 76.0 |
| COMMAND R | 62.5 | 51.2 | 55.5 | 52.8 | 56.2 | 70.3 | 69.0 | 55.7 | 64.3 | 59.7 |
| SAT | 82.7 | 86.8 | 68.8 | 73.2 | 85.1 | 71.2 | 82.6 | 67.8 | 86.4 | 78.3 |
| SAT$_{+SM}$ | **85.1** | **95.8** | **80.1** | **89.3** | **88.3** | **80.6** | **93.1** | **81.3** | **92.7** | **87.4** |
| NLTK | 75.6 | 51.5 | 65.2 | 87.7 | 72.9 | 45.3 | 76.3 | 65.1 | 73.3 | 68.1 |
| PYSBD | 74.2 | 50.5 | 60.8 | 79.7 | 74.1 | 55.0 | 75.0 | 67.6 | 70.4 | 67.5 |
| SPACY$_{DP}$ | 71.6 | 74.3 | 64.9 | 87.3 | 74.0 | 54.4 | 84.5 | 68.2 | 65.2 | 71.6 |
| WTP | 87.8 | 63.0 | 75.6 | 84.6 | 84.3 | 80.0 | 88.8 | 79.7 | 85.8 | 81.1 |
| WTP$_{T}$ | 87.0 | 80.7 | 76.3 | 87.1 | 85.0 | 79.5 | 88.7 | 80.4 | 85.9 | 83.4 |
| WTP$_{PUNCT}$ | **96.8** | 98.4 | 88.7 | 94.6 | 94.0 | 96.9 | 96.3 | 87.3 | 93.7 | 94.1 |
| MLSBD-T $^{\text{Mono}}_{\text{Specific}}$ | 96.4 | 97.7 | 88.7 | 93.8 | 93.7 | 98.0 | 95.3 | 86.9 | 93.5 | 93.8 |
| MLSBD-T $^{\text{Mono}}_{\text{Both}}$ | 96.5 | 98.5 | 88.2 | 93.7 | 87.1 | 84.4 | 95.3 | 87.6 | 92.7 | 91.6 |
| MLSBD-T $^{\text{Multi}}_{\text{Specific}}$ | 96.4 | **98.9** | 89.0 | **94.8** | 94.5 | 98.1 | 95.7 | 87.5 | 93.6 | 94.3 |
| MLSBD-T $^{\text{Multi}}_{\text{Both}}$ | 96.6 | **98.9** | 88.7 | **94.8** | 94.6 | 98.2 | 95.6 | 78.6 | 93.2 | 93.2 |
| SAT$_{+LoRA}$@10 | 95.1 | 96.6 | 86.5 | 93.5 | 94.0 | 85.6 | 96.3 | 87.7 | **97.3** | 92.5 |
| SAT$_{+LoRA}$@100 | 96.7 | 97.0 | **89.3** | 94.0 | 95.4 | 87.1 | 97.0 | 88.8 | **97.3** | 93.6 |
| SAT$_{+LoRA}$ | **96.8** | 98.5 | 89.1 | 94.4 | **95.5** | **98.3** | **97.3** | **88.9** | 97.1 | **95.1** |

Table 21: Sentence segmentation F1 score for legal data (MultiLegalSBD). We take the macro F1 scores over documents within a given category. @N correspond to using a maximum of N documents per category for adaptation, respectively. Transformer-based MLSBD-T baselines are taken from Brugger et al. (2023). For these domain-specific baselines, *mono* and *multi* correspond to models trained on documents from only one or all languages, respectively. *Both* corresponds to models trained on both laws and judgments, whereas *specific* corresponds to models trained on a given category (laws/judgments). Best per-category results are in **bold**.

| Model | fr | | es | | it | | en | de | | Macro Avg. |
|---|---|---|---|---|---|---|---|---|---|---|
| | Judg. | Laws | Judg. | Laws | Judg. | Laws | Judg. | Judg. | Laws | |
| SpaCy$_{\text{M}}$ | 0.0 | 0.0 | 1.5 | 0.0 | 0.4 | 0.0 | 0.2 | 2.2 | 0.0 | 0.5 |
| Ersatz | 0.7 | 0.1 | 4.7 | 0.0 | - | - | 0.6 | 4.1 | - | 1.7 |
| Llama 3$_{\text{8B}}$ | 60.0 | 40.7 | 50.5 | 55.9 | 54.2 | 35.6 | 69.2 | 59.2 | 74.2 | 55.5 |
| Command R | 54.3 | 57.7 | 42.8 | 44.5 | 46.4 | 42.9 | 56.0 | 45.4 | 64.1 | 50.5 |
| SaT | 63.7 | 80.6 | 54.9 | 63.3 | 56.0 | 55.9 | 49.9 | 57.6 | 75.0 | 61.9 |
| SaT$_{\text{+SM}}$ | **68.8** | **89.5** | **65.3** | **83.3** | **62.3** | **71.9** | **74.5** | **72.0** | **80.9** | **74.3** |
| NLTK | 1.4 | 0.0 | 3.8 | 0.0 | 1.3 | 1.6 | 0.2 | 4.9 | 0.0 | 1.5 |
| PySBD | 21.7 | 0.2 | 3.8 | 0.0 | 20.0 | 0.2 | 1.3 | 0.4 | 0.0 | 5.3 |
| SpaCy$_{\text{DP}}$ | 7.9 | 4.4 | 4.6 | 0.0 | 5.3 | 2.3 | 2.4 | 13.1 | 8.8 | 5.4 |
| WtP | 32.2 | 19.5 | 38.0 | 41.1 | 36.9 | 28.8 | 28.1 | 25.0 | 19.0 | 29.8 |
| WtP$_{\text{T}}$ | 48.9 | 64.7 | 51.8 | 71.5 | 46.5 | 43.8 | 54.5 | 52.4 | 62.0 | 55.1 |
| WtP$_{\text{PUNCT}}$ | 65.0 | 83.4 | 67.2 | 83.1 | 58.3 | 66.9 | 73.5 | 73.0 | 84.1 | 72.7 |
| mLSBD-T$^{\text{Mono}}_{\text{Specific}}$ | 9.6 | 45.1 | 7.9 | 13.2 | 6.9 | 47.1 | 3.6 | 12.1 | 0.3 | 16.2 |
| mLSBD-T$^{\text{Mono}}_{\text{Both}}$ | 9.0 | 43.0 | 8.1 | 15.6 | 6.7 | 39.6 | 3.6 | 7.7 | 0.6 | 14.9 |
| mLSBD-T$^{\text{Multi}}_{\text{Specific}}$ | 8.8 | 44.4 | 7.3 | 34.4 | 6.9 | 47.2 | 2.5 | 9.2 | 0.7 | 17.9 |
| mLSBD-T$^{\text{Multi}}_{\text{Both}}$ | 8.9 | 43.5 | 5.6 | 24.6 | 6.9 | 47.4 | 1.3 | 2.4 | 0.4 | 15.7 |
| SaT$_{\text{+LoRA}}$@10 | 70.8 | 82.9 | 67.8 | 79.9 | 63.5 | 62.6 | 80.6 | 78.6 | **93.5** | 75.6 |
| SaT$_{\text{+LoRA}}$@100 | 76.9 | 86.6 | 75.9 | 84.1 | 69.3 | 75.3 | **84.2** | **84.1** | **93.5** | 81.1 |
| SaT$_{\text{+LoRA}}$ | **77.5** | **90.2** | **76.1** | **85.5** | **71.2** | **87.5** | **84.2** | 83.8 | **93.5** | **83.3** |

Table 22: Sentence segmentation F1 score for corrupted legal data (MultiLegalSBD), where we *remove all casing and punctuation tokens*. We take the macro F1 scores over documents within a given category.

| Model | Repetitiveness | | | | |
|---|---|---|---|---|---|
| | High | | | Low | |
| | Country | Punk Rock | Pop Punk | Southern Hip Hop | Gangsta Rap |
| SSM$_{\text{string}}$ † | - | - | - | - | - |
| Llama 3$_{\text{8B}}$ | 47.0 | 50.2 | 49.5 | 34.7 | 32.8 |
| Command R | 35.3 | 39.7 | 39.2 | 27.5 | 29.9 |
| WtP$_{\text{PUNCT}}$@100 | 56.5 | 56.1 | 54.9 | 43.3 | 42.4 |
| WtP$_{\text{PUNCT}}$@1000 | 58.9 | 58.8 | 57.5 | 45.9 | 43.9 |
| WtP$_{\text{PUNCT}}$ | 58.9 | 58.8 | 57.5 | 45.9 | 43.9 |
| SaT$_{\text{+LoRA}}$@100 | 66.7 | 67.5 | 69.2 | 53.0 | 50.3 |
| SaT$_{\text{+LoRA}}$@1000 | 76.7 | 76.0 | 76.8 | **64.5** | **60.9** |
| SaT$_{\text{+LoRA}}$ | **79.3** | **76.8** | **77.6** | 64.2 | 60.3 |

Table 23: Complete verse segmentation F1 scores for songs categorized as *low* and *high* repetitiveness. Categorization of songs into genres and repetitiveness taken from Fell et al. (2018). We report the macro average over songs. †Values for SSM$_{\text{string}}$ taken from Fell et al. (2018), with lyrics already pre-segmented into lines.@N corresponds to using a maximum of N and 1000 songs per genre for adaptation, respectively.

| Model | Mid Repetitiveness | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Rock | Pop | RnB | Soul | Alt. Rock | Alt. Metal | Indie Rock | Pop Rock | Hard Rock |
| $\text{SSM}_{\text{string}}$ † | 64.8 | 66.6 | 65.6 | 63.0 | 67.9 | 68.5 | 65.6 | 65.8 | 67.7 |
| LLAMA $3_{8B}$ | 48.2 | 47.0 | 45.7 | 48.7 | 49.3 | 47.6 | 48.4 | 47.1 | 48.7 |
| COMMAND R | 37.8 | 36.3 | 33.3 | 36.5 | 40.1 | 39.1 | 40.2 | 37.6 | 38.4 |
| $\text{WTP}_{\text{PUNCT}}$@100 | 57.5 | 55.9 | 51.4 | 52.5 | 59.9 | 58.6 | 56.3 | 57.5 | 57.0 |
| $\text{WTP}_{\text{PUNCT}}$@1000 | 60.5 | 57.7 | 53.1 | 55.0 | 63.0 | 60.5 | 59.8 | 58.1 | 59.4 |
| $\text{WTP}_{\text{PUNCT}}$ | 60.7 | 58.2 | 53.5 | 55.0 | 63.3 | 60.5 | 60.1 | 58.3 | 59.5 |
| $\text{SAT}_{\text{+LoRA}}$@100 | 65.4 | 63.8 | 61.5 | 59.8 | 64.9 | 68.6 | 63.9 | 63.8 | 64.1 |
| $\text{SAT}_{\text{+LoRA}}$@1000 | 74.7 | 72.7 | 71.2 | 71.1 | 75.3 | **77.7** | 72.6 | 74.5 | 75.7 |
| $\text{SAT}_{\text{+LoRA}}$ | **78.1** | **75.6** | **73.4** | **71.7** | **77.4** | **77.7** | **73.5** | **75.6** | **76.6** |

Table 24: Complete verse segmentation F1 scores for songs categorized as *mid* repetitiveness.

| Model | Repetitiveness | | | | |
|---|---|---|---|---|---|
| | High | | | Low | |
| | Country | Punk Rock | Pop Punk | Southern Hip Hop | Gangsta Rap |
| $\text{SSM}_{\text{string}}$ † | 70.2 | 70.9 | 72.7 | 47.0 | 47.7 |
| LLAMA $3_{8B}$ | 54.6 | 53.5 | 57.1 | 36.6 | 36.2 |
| COMMAND R | 38.8 | 42.6 | 41.4 | 35.0 | 26.3 |
| $\text{WTP}_{\text{PUNCT}}$@100 | 48.9 | 52.5 | 50.0 | 35.5 | 36.4 |
| $\text{WTP}_{\text{PUNCT}}$@1000 | 51.7 | 55.6 | 53.1 | 36.5 | 39.5 |
| $\text{WTP}_{\text{PUNCT}}$ | 51.9 | 55.6 | 53.1 | 36.5 | 39.5 |
| $\text{SAT}_{\text{+LoRA}}$@100 | 66.7 | 67.2 | 67.7 | 44.9 | 48.4 |
| $\text{SAT}_{\text{+LoRA}}$@1000 | 72.7 | **71.8** | 74.5 | 54.7 | **55.8** |
| $\text{SAT}_{\text{+LoRA}}$ | **75.2** | 71.4 | **74.7** | **54.8** | **55.8** |

Table 25: Complete verse segmentation F1 scores for songs categorized as *low* and *high* repetitiveness, where we *remove all casing and punctuation tokens*. We report the macro average over songs.

| Model | Mid Repetitiveness | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Rock | Pop | RnB | Soul | Alt. Rock | Alt. Metal | Indie Rock | Pop Rock | Hard Rock |
| $\text{SSM}_{\text{string}}$ † | - | - | - | - | - | - | - | - | - |
| LLAMA $3_{8B}$ | 53.8 | 51.7 | 47.4 | 51.9 | 54.1 | 52.3 | 54.0 | 52.8 | 52.2 |
| COMMAND R | 41.1 | 37.5 | 34.0 | 40.0 | 41.4 | 40.7 | 42.6 | 39.5 | 40.6 |
| $\text{WTP}_{\text{PUNCT}}$@100 | 51.5 | 47.0 | 41.5 | 47.3 | 52.6 | 52.7 | 52.4 | 48.8 | 50.4 |
| $\text{WTP}_{\text{PUNCT}}$@1000 | 52.8 | 48.9 | 43.1 | 48.6 | 54.9 | 54.2 | 55.1 | 51.2 | 52.0 |
| $\text{WTP}_{\text{PUNCT}}$ | 53.1 | 49.4 | 43.2 | 48.6 | 55.0 | 54.2 | 54.8 | 51.0 | 52.6 |
| $\text{SAT}_{\text{+LoRA}}$@100 | 64.5 | 62.2 | 58.8 | 60.8 | 67.4 | 66.9 | 62.7 | 63.5 | 63.6 |
| $\text{SAT}_{\text{+LoRA}}$@1000 | 70.4 | 68.5 | 65.2 | **65.6** | 71.4 | 73.1 | **69.9** | 68.3 | 71.4 |
| $\text{SAT}_{\text{+LoRA}}$ | **73.2** | **71.5** | **67.1** | 65.1 | **73.3** | **73.5** | 69.5 | **70.4** | **72.7** |

Table 26: Complete verse segmentation F1 scores for songs categorized as *mid* repetitiveness, where we *remove all casing and punctuation tokens*. We report the macro average over songs.

| | Model | af | am | ar | az | be | bg | bn | ca | ceb | cs | cy | da | de | el | en |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UD | SpaCy$_{M}$ | 98.3 | - | 79.1 | - | 80.0 | 93.8 | 47.9 | 98.8 | 98.5 | 89.6 | 98.9 | 94.9 | 87.7 | 93.8 | 91.7 |
| | Ersatz | - | - | 79.7 | - | - | - | - | - | - | 89.3 | - | - | 92.4 | - | 89.1 |
| | LLaMA 3$_{8B}$ | **100.0** | - | 80.1 | - | 88.1 | 97.3 | 96.9 | 99.4 | **99.7** | 93.0 | 98.7 | 95.2 | 96.7 | 94.7 | 94.8 |
| | Command R | 86.1 | - | 75.6 | - | 68.1 | 76.9 | 59.2 | 76.6 | 89.9 | 68.9 | 80.8 | 73.8 | 85.3 | 66.7 | 77.1 |
| | SaT | 99.1 | - | 82.9 | - | 89.0 | 97.9 | 96.2 | 98.9 | 99.7 | 91.5 | 99.1 | 95.5 | 96.4 | 97.5 | 93.9 |
| | SaT$_{+SM}$ | **100.0** | - | **84.5** | - | **93.5** | **99.3** | **100.0** | **99.7** | **99.7** | **94.3** | **99.4** | **98.5** | **97.8** | **97.9** | **96.7** |
| | NLTK | - | - | - | - | - | - | - | - | - | 89.1 | - | 94.4 | 92.6 | 92.7 | 90.8 |
| | PySBD | - | - | 28.1 | - | - | 74.9 | - | - | - | - | - | 72.6 | 80.0 | 91.0 | 75.3 |
| | SpaCy$_{DP}$ | - | - | - | - | - | - | - | 99.8 | - | - | - | 94.0 | 96.7 | 94.0 | 91.3 |
| | WtP | 98.3 | - | 80.5 | - | 88.9 | 98.2 | **93.5** | 98.3 | **99.7** | 92.3 | 99.2 | 95.1 | 95.6 | 97.3 | 94.5 |
| | WtP$_{T}$ | 99.0 | - | 86.4 | - | 88.8 | 98.1 | - | 98.4 | - | 92.0 | 99.2 | 94.3 | 95.8 | 97.7 | 94.5 |
| | WtP$_{PUNCT}$ | 99.9 | - | **87.4** | - | **91.9** | **99.6** | - | 99.6 | - | **95.4** | 99.5 | **98.9** | 96.5 | 97.8 | **96.9** |
| | SaT$_{+LoRA}$ | **100.0** | - | 86.6 | - | 91.2 | 99.4 | - | **99.9** | - | 95.2 | **99.6** | 98.7 | **96.8** | **98.9** | 96.8 |
| OPUS100 | SpaCy$_{M}$ | 41.9 | 6.3 | 57.9 | 72.3 | 33.9 | 93.4 | 37.2 | 88.0 | - | 87.3 | 25.8 | 90.2 | 72.9 | 89.2 | 88.8 |
| | Ersatz | - | - | 59.2 | - | - | - | - | - | - | 86.5 | - | - | 73.1 | - | 87.6 |
| | LLaMA 3$_{8B}$ | 65.4 | 36.2 | 62.2 | 76.8 | 54.5 | 94.6 | 80.0 | 90.9 | - | 91.3 | 46.3 | 93.2 | 83.8 | 94.3 | 92.8 |
| | Command R | 55.8 | 6.6 | 44.5 | 60.5 | 36.1 | 57.9 | 4.8 | 66.0 | - | 69.0 | 39.8 | 75.4 | 76.2 | 65.1 | 89.5 |
| | SaT | 78.4 | 58.0 | **67.0** | 75.0 | 70.4 | 93.5 | 80.5 | 87.7 | - | 89.2 | 72.0 | 90.9 | 78.2 | 92.0 | 90.4 |
| | SaT$_{+SM}$ | **86.2** | **70.8** | 65.2 | **85.3** | **87.8** | **96.2** | **86.1** | **93.0** | - | **94.3** | **79.6** | **94.0** | **86.9** | **95.9** | **94.6** |
| | NLTK | - | - | - | - | - | - | - | - | - | 86.7 | - | 89.9 | 73.3 | 84.8 | 88.2 |
| | PySBD | - | 5.9 | 38.0 | - | - | 72.9 | - | - | - | - | - | 70.2 | 66.5 | 62.7 | 59.6 |
| | SpaCy$_{DP}$ | - | - | - | - | - | - | - | 87.3 | - | - | - | 90.2 | 74.0 | 91.1 | 89.0 |
| | WtP | 74.6 | 58.2 | 64.5 | 74.9 | 71.7 | 93.2 | 77.9 | 87.7 | - | 87.5 | 68.7 | 88.2 | 76.7 | 90.9 | 90.6 |
| | WtP$_{T}$ | 76.4 | 63.7 | 64.6 | 74.6 | 72.5 | 92.8 | 82.1 | 88.6 | - | 90.0 | 74.2 | 90.1 | 84.6 | 91.9 | 89.4 |
| | WtP$_{PUNCT}$ | 87.8 | 70.5 | 76.1 | 83.0 | 89.1 | 96.2 | 86.5 | 94.0 | - | 94.9 | 81.0 | 94.5 | 89.4 | 95.9 | 94.7 |
| | SaT$_{+LoRA}$ | **88.5** | **75.9** | **79.1** | **85.0** | **89.4** | **96.4** | **88.6** | **94.6** | - | **95.0** | **83.0** | **95.2** | **90.1** | **96.1** | **94.8** |
| Ersatz | SpaCy$_{M}$ | - | - | 91.1 | - | - | - | - | - | - | 96.4 | - | - | 93.5 | - | 94.0 |
| | Ersatz | - | - | 92.8 | - | - | - | - | - | - | 96.7 | - | - | 95.4 | - | 97.5 |
| | LLaMA 3$_{8B}$ | - | - | **92.4** | - | - | - | - | - | - | 95.9 | - | - | **97.3** | - | 98.2 |
| | Command R | - | - | 55.8 | - | - | - | - | - | - | 66.4 | - | - | 75.9 | - | 87.2 |
| | SaT | - | - | 89.7 | - | - | - | - | - | - | 94.3 | - | - | 96.6 | - | 96.7 |
| | SaT$_{+SM}$ | - | - | 92.3 | - | - | - | - | - | - | **98.5** | - | - | 97.2 | - | **98.3** |
| | NLTK | - | - | - | - | - | - | - | - | - | 96.7 | - | - | 95.3 | - | 97.7 |
| | PySBD | - | - | 46.2 | - | - | - | - | - | - | - | - | - | 95.3 | - | 73.9 |
| | SpaCy$_{DP}$ | - | - | - | - | - | - | - | - | - | - | - | - | 96.3 | - | 98.5 |
| | WtP | - | - | 87.0 | - | - | - | - | - | - | 93.6 | - | - | 95.3 | - | 96.5 |
| | WtP$_{T}$ | - | - | 88.7 | - | - | - | - | - | - | 93.9 | - | - | 95.6 | - | 96.7 |
| | WtP$_{PUNCT}$ | - | - | 92.7 | - | - | - | - | - | - | **99.0** | - | - | **99.3** | - | 98.6 |
| | SaT$_{+LoRA}$ | - | - | **93.1** | - | - | - | - | - | - | 98.5 | - | - | **99.3** | - | **98.7** |

Table 27: Sentence segmentation test F1 scores on languages af-en. Results are shown using 3-layer variations of all models. Numerically best results are in **bold**, statistically indistinguishable ones from this best are underlined.

| | Model | eo | es | et | eu | fa | fi | fr | fy | ga | gd | gl | gu | ha | he | hi |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UD | SPACY$_{\text{M}}$ | - | 98.6 | 93.6 | 95.7 | 99.8 | 92.7 | 96.1 | - | 96.3 | 62.9 | 92.1 | - | - | 93.9 | - |
| | ERSATZ | - | 97.5 | 92.9 | - | - | 92.8 | 97.3 | - | - | - | - | - | - | - | 99.5 |
| | LLAMA 3$_{\text{8B}}$ | - | 98.5 | 94.1 | 97.3 | 99.7 | 95.9 | **99.1** | - | 94.5 | 67.6 | 94.6 | - | - | 93.4 | 99.8 |
| | COMMAND R | - | 81.8 | 74.9 | 79.4 | 95.5 | 75.3 | 92.3 | - | 73.5 | 54.2 | 73.7 | - | - | 77.2 | 91.8 |
| | SAT | - | 97.0 | 92.8 | 97.0 | 98.5 | 93.8 | 97.5 | - | 87.3 | 68.1 | 97.4 | - | - | 94.2 | 96.2 |
| | SAT$_{\text{+SM}}$ | - | **99.4** | **98.2** | **100.0** | **99.9** | **97.2** | 98.3 | - | **95.5** | **84.3** | **98.9** | - | - | 95.5 | **99.9** |
| | NLTK | - | 98.5 | 93.6 | - | - | 92.6 | 97.0 | - | - | - | - | - | - | - | - |
| | PYSBD | - | 46.2 | - | - | 98.8 | - | 62.1 | - | - | - | - | - | - | - | 99.8 |
| | SPACY$_{\text{DP}}$ | - | 99.1 | - | - | - | 94.9 | 91.6 | - | - | - | - | - | - | - | - |
| | WTP | - | 96.5 | 92.6 | 97.1 | 96.6 | 92.1 | 96.4 | - | 84.3 | 71.2 | 97.5 | - | - | 95.1 | 96.1 |
| | WTP$_{\text{T}}$ | - | 96.9 | 92.5 | 97.3 | 97.8 | 92.7 | 96.6 | - | 90.4 | 70.7 | **98.7** | - | - | 96.1 | 96.8 |
| | WTP$_{\text{PUNCT}}$ | - | **99.7** | **98.0** | 99.9 | **100.0** | **98.1** | 98.3 | - | 98.2 | 79.6 | 98.6 | - | - | **97.1** | **99.9** |
| | SAT$_{\text{+LoRA}}$ | - | 99.6 | 97.7 | **100.0** | **100.0** | 97.7 | **98.9** | - | **98.8** | **82.2** | 98.7 | - | - | 96.4 | 99.9 |
| OPUS100 | SPACY$_{\text{M}}$ | 88.4 | 90.4 | 87.0 | 80.7 | 54.5 | 92.9 | 85.1 | 21.7 | 61.0 | 36.4 | 88.2 | 5.2 | 88.5 | 91.8 | 52.8 |
| | ERSATZ | - | 90.0 | 87.4 | - | - | 92.7 | 86.1 | - | - | - | - | 21.2 | - | - | 58.0 |
| | LLAMA 3$_{\text{8B}}$ | 91.6 | 93.6 | 89.0 | 84.5 | 60.4 | 93.9 | 91.4 | 42.4 | 71.3 | 62.8 | 90.8 | 33.0 | 88.1 | 91.6 | 59.6 |
| | COMMAND R | 84.6 | 80.1 | 67.2 | 57.8 | 42.9 | 68.4 | 80.8 | 30.6 | 55.9 | 50.5 | 71.6 | 16.7 | 65.8 | 62.7 | 39.3 |
| | SAT | 90.3 | 91.1 | 84.6 | 82.2 | 56.8 | 91.3 | 87.3 | 64.9 | **82.3** | 81.6 | 87.9 | 71.5 | 83.8 | 89.4 | 62.9 |
| | SAT$_{\text{+SM}}$ | **95.2** | **95.3** | **93.0** | **90.0** | **60.1** | **95.3** | **92.6** | **91.0** | 80.8 | **82.5** | **93.1** | **83.4** | **91.4** | **92.2** | **72.8** |
| | NLTK | - | 89.8 | 87.6 | - | - | 93.1 | 85.8 | - | - | - | - | - | - | - | - |
| | PYSBD | - | 67.6 | - | - | 41.3 | - | 80.9 | - | - | - | - | - | - | - | 23.0 |
| | SPACY$_{\text{DP}}$ | - | 88.0 | - | - | - | 92.1 | 84.1 | - | - | - | - | - | - | - | - |
| | WTP | 90.9 | 89.9 | 82.5 | 84.4 | 59.6 | 90.8 | **86.8** | 44.4 | 77.8 | 83.5 | 88.5 | 69.2 | 82.8 | 90.2 | 65.1 |
| | WTP$_{\text{T}}$ | 90.5 | 91.4 | 87.7 | 86.0 | 59.3 | 92.5 | - | 61.0 | 77.3 | 83.7 | 88.9 | 69.6 | 88.9 | 89.4 | 64.3 |
| | WTP$_{\text{PUNCT}}$ | 95.4 | 95.0 | 94.6 | 91.6 | 72.7 | 95.5 | - | 88.1 | 87.5 | 92.7 | 93.9 | 76.5 | **91.5** | 93.8 | 76.3 |
| | SAT$_{\text{+LoRA}}$ | **95.8** | **95.8** | **94.7** | **92.8** | **74.1** | **96.3** | - | **88.8** | **90.6** | **94.5** | **94.6** | **80.5** | 91.0 | **94.1** | **81.5** |
| Ersatz | SPACY$_{\text{M}}$ | - | 97.2 | 97.0 | - | - | 95.0 | 96.4 | - | - | - | - | 3.8 | - | - | 17.9 |
| | ERSATZ | - | 96.6 | 98.0 | - | - | 96.0 | 96.3 | - | - | - | - | 94.3 | - | - | 96.8 |
| | LLAMA 3$_{\text{8B}}$ | - | 98.3 | 97.0 | - | - | 96.7 | 97.9 | - | - | - | - | **93.1** | - | - | 97.3 |
| | COMMAND R | - | 76.8 | 77.2 | - | - | 78.8 | 84.6 | - | - | - | - | 74.8 | - | - | 84.9 |
| | SAT | - | 98.4 | 95.9 | - | - | 97.6 | 97.3 | - | - | - | - | 92.0 | - | - | 95.5 |
| | SAT$_{\text{+SM}}$ | - | **99.5** | **98.9** | - | - | **98.4** | **98.5** | - | - | - | - | 70.7 | - | - | **98.1** |
| | NLTK | - | 96.4 | 97.5 | - | - | 95.9 | 96.0 | - | - | - | - | - | - | - | - |
| | PYSBD | - | 84.2 | - | - | - | - | 96.0 | - | - | - | - | - | - | - | 87.5 |
| | SPACY$_{\text{DP}}$ | - | - | - | - | - | - | 91.0 | - | - | - | - | - | - | - | - |
| | WTP | - | **98.7** | 96.0 | - | - | 97.4 | 97.2 | - | - | - | - | 89.7 | - | - | 93.9 |
| | WTP$_{\text{T}}$ | - | - | 95.7 | - | - | 97.2 | 96.6 | - | - | - | - | 88.9 | - | - | 94.6 |
| | WTP$_{\text{PUNCT}}$ | - | - | 98.0 | - | - | **99.4** | 98.4 | - | - | - | - | **96.7** | - | - | 96.3 |
| | SAT$_{\text{+LoRA}}$ | - | - | **99.0** | - | - | 98.6 | **99.1** | - | - | - | - | 95.4 | - | - | **97.3** |

Table 28: Sentence segmentation test F1 scores on languages eo-hi.

| | Model | hu | hy | id | ig | is | it | ja | jv | ka | kk | km | kn | ko | ku | ky |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UD | SpaCy$_M$ | 98.3 | 11.3 | 97.8 | - | 95.1 | 92.3 | 0.0 | 97.9 | - | 96.1 | - | - | 99.8 | - | - |
| | Ersatz | - | - | - | - | - | - | 93.4 | - | - | 95.6 | - | - | - | - | - |
| | Llama 3$_{8B}$ | 98.1 | 92.0 | 99.3 | - | 94.1 | 98.1 | **97.9** | 99.1 | - | 98.5 | - | - | 99.9 | - | - |
| | Command R | 73.2 | 50.3 | 91.7 | - | 74.6 | 88.8 | 90.7 | 83.3 | - | 86.1 | - | - | 79.9 | - | - |
| | SaT | 97.0 | 97.6 | 98.5 | - | 73.4 | 96.3 | 96.1 | 98.8 | - | 95.8 | - | - | 99.6 | - | - |
| | SaT$_{+SM}$ | **99.5** | **98.4** | **99.6** | - | **95.7** | **98.7** | 97.1 | **99.3** | - | **99.3** | - | - | **100.0** | - | - |
| | NLTK | - | - | - | - | - | 95.2 | - | - | - | - | - | - | - | - | - |
| | PySBD | - | 92.8 | - | - | - | 74.9 | 97.9 | - | - | 95.5 | - | - | - | - | - |
| | SpaCy$_{DP}$ | - | - | - | - | - | 99.5 | 97.8 | - | - | - | - | - | 99.9 | - | - |
| | WtP | 96.0 | 96.0 | **98.0** | - | 85.8 | 93.7 | 93.4 | **97.8** | - | **97.4** | - | - | 99.2 | - | - |
| | WtP$_T$ | 96.3 | 96.2 | - | - | 88.9 | 93.7 | 95.7 | - | - | 82.8 | - | - | 99.4 | - | - |
| | WtP$_{PUNCT}$ | **99.5** | **98.5** | - | - | **96.6** | 99.3 | **98.1** | - | - | 84.7 | - | - | **99.9** | - | - |
| | SaT$_{+LoRA}$ | **99.5** | 98.3 | - | - | **96.8** | **99.7** | 98.0 | - | - | 96.9 | - | - | **99.9** | - | - |
| OPUS100 | SpaCy$_M$ | 93.0 | 24.2 | 89.6 | 29.8 | 95.0 | 85.8 | 0.1 | - | 38.3 | 42.1 | 0.0 | 9.8 | 50.9 | 26.8 | 21.7 |
| | Ersatz | - | - | - | - | - | - | 28.8 | - | - | 37.9 | 0.0 | - | - | - | - |
| | Llama 3$_{8B}$ | 94.2 | 73.1 | 92.3 | 39.3 | 95.1 | 90.7 | 9.0 | - | 89.2 | 56.4 | 9.9 | 26.7 | 70.3 | 41.9 | 58.3 |
| | Command R | 62.6 | 46.7 | 71.3 | 28.0 | 77.7 | 78.6 | 6.0 | - | 25.6 | 34.5 | 4.3 | 5.5 | 38.9 | 29.8 | 26.5 |
| | SaT | 92.3 | 85.3 | 86.6 | 81.7 | 93.7 | 87.0 | **83.1** | - | 75.7 | 80.3 | 70.4 | 70.2 | 72.4 | 77.2 | 84.1 |
| | SaT$_{+SM}$ | **95.8** | **90.6** | **93.7** | **92.1** | **96.5** | **90.9** | 78.0 | - | **93.6** | **92.2** | **86.6** | **87.9** | **78.9** | **91.1** | **91.8** |
| | NLTK | - | - | - | - | - | 87.5 | - | - | - | - | - | - | - | - | - |
| | PySBD | - | 58.4 | - | - | - | 70.2 | 43.4 | - | - | 35.7 | - | - | - | - | - |
| | SpaCy$_{DP}$ | - | - | - | - | - | 85.3 | 42.6 | - | - | - | - | - | 46.6 | - | - |
| | WtP | 91.6 | **85.1** | 88.9 | 78.6 | 93.9 | 84.6 | 44.6 | - | 91.3 | 73.4 | 71.8 | 64.5 | 56.7 | 78.1 | 84.5 |
| | WtP$_T$ | 92.1 | - | 89.5 | 82.1 | 94.4 | 88.4 | 79.5 | - | 91.1 | 74.9 | 71.1 | 60.4 | 70.4 | 66.4 | 84.4 |
| | WtP$_{PUNCT}$ | 96.2 | - | 93.9 | 90.2 | 96.6 | 93.4 | 86.7 | - | 92.8 | 92.1 | 79.0 | 78.0 | 81.6 | 85.1 | 90.3 |
| | SaT$_{+LoRA}$ | **96.3** | - | **94.1** | **92.3** | **96.6** | **94.3** | **90.5** | - | **93.3** | **92.1** | **87.8** | **83.9** | **81.8** | **90.9** | **92.0** |
| Ersatz | SpaCy$_M$ | - | - | - | - | - | - | 0.0 | - | - | 96.9 | 0.0 | - | - | - | - |
| | Ersatz | - | - | - | - | - | - | 85.7 | - | - | 99.6 | 31.3 | - | - | - | - |
| | Llama 3$_{8B}$ | - | - | - | - | - | - | 87.1 | - | - | 99.0 | 85.6 | - | - | - | - |
| | Command R | - | - | - | - | - | - | 59.7 | - | - | 86.7 | 6.7 | - | - | - | - |
| | SaT | - | - | - | - | - | - | 86.5 | - | - | 97.1 | **89.4** | - | - | - | - |
| | SaT$_{+SM}$ | - | - | - | - | - | - | **89.1** | - | - | **99.7** | 83.9 | - | - | - | - |
| | NLTK | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| | PySBD | - | - | - | - | - | - | 87.0 | - | - | 64.7 | - | - | - | - | - |
| | SpaCy$_{DP}$ | - | - | - | - | - | - | 91.0 | - | - | - | - | - | - | - | - |
| | WtP | - | - | - | - | - | - | 80.2 | - | - | 96.3 | 70.2 | - | - | - | - |
| | WtP$_T$ | - | - | - | - | - | - | 81.5 | - | - | 95.7 | 91.4 | - | - | - | - |
| | WtP$_{PUNCT}$ | - | - | - | - | - | - | 96.7 | - | - | 99.7 | 92.0 | - | - | - | - |
| | SaT$_{+LoRA}$ | - | - | - | - | - | - | **94.6** | - | - | **99.8** | **92.0** | - | - | - | - |

Table 29: Sentence segmentation test F1 scores on languages hu-ky.

| | Model | la | lt | lv | mg | mk | ml | mn | mr | ms | mt | my | ne | nl | no | pa |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UD | SpaCy$_{M}$ | 0.0 | 93.8 | 98.5 | - | - | - | - | 92.5 | - | 85.7 | - | - | 92.8 | 96.9 | - |
| | Ersatz | - | 92.2 | 96.9 | - | - | - | - | - | - | - | - | - | - | - | - |
| | Llama 3$_{8B}$ | 90.0 | 94.3 | 97.0 | - | - | - | - | 90.0 | - | **94.4** | - | - | **96.0** | 96.5 | - |
| | Command R | 65.1 | 65.7 | 69.5 | - | - | - | - | 71.6 | - | 70.1 | - | - | 66.3 | 69.5 | - |
| | SaT | 67.8 | 97.1 | 96.4 | - | - | - | - | 83.5 | - | 86.7 | - | - | 92.6 | 98.6 | - |
| | SaT$_{+SM}$ | **96.6** | **97.9** | **99.2** | - | - | - | - | **100.0** | - | 93.0 | - | - | 95.7 | **99.0** | - |
| | NLTK | - | - | - | - | - | - | - | - | - | - | - | - | 95.7 | 95.6 | - |
| | PySBD | - | - | - | - | - | - | - | 60.3 | - | - | - | - | 93.6 | - | - |
| | SpaCy$_{DP}$ | - | 92.0 | - | - | - | - | - | - | - | - | - | - | 93.1 | - | - |
| | WtP | 89.2 | 98.2 | 96.5 | - | - | - | - | 89.4 | - | 89.8 | - | - | 94.1 | 98.2 | - |
| | WtP$_{T}$ | 89.3 | 97.9 | 96.4 | - | - | - | - | 92.0 | - | 87.3 | - | - | 92.9 | 98.4 | - |
| | WtP$_{PUNCT}$ | 97.3 | **99.6** | **99.0** | - | - | - | - | **98.8** | - | **93.6** | - | - | **97.0** | **99.4** | - |
| | SaT$_{+LoRA}$ | **97.5** | 98.2 | 98.9 | - | - | - | - | 96.5 | - | 90.9 | - | - | 96.0 | 99.1 | - |
| OPUS100 | SpaCy$_{M}$ | - | 76.5 | 76.9 | 83.1 | 93.2 | 38.7 | 32.8 | 86.5 | 87.6 | 55.6 | 0.0 | 6.4 | 93.0 | 95.0 | 4.9 |
| | Ersatz | - | 77.0 | 77.6 | - | - | - | - | - | - | - | - | - | - | - | - |
| | Llama 3$_{8B}$ | - | 82.8 | 83.1 | 85.7 | 94.1 | 80.9 | 49.3 | 88.1 | 91.0 | 78.8 | 16.3 | 39.7 | 93.7 | 95.3 | 26.7 |
| | Command R | - | 70.3 | 70.4 | 64.0 | 48.2 | 0.7 | 26.6 | 65.2 | 67.9 | 65.1 | 17.1 | 23.3 | 74.8 | 71.5 | 14.4 |
| | SaT | - | 82.6 | 82.6 | 88.5 | 93.0 | 77.6 | 73.5 | 89.9 | 85.8 | 62.2 | 74.1 | 69.8 | 92.1 | 94.8 | 69.5 |
| | SaT$_{+SM}$ | - | **90.0** | **89.7** | **91.6** | **95.4** | **85.9** | **90.1** | **93.3** | **94.1** | **85.1** | **89.4** | **82.0** | **94.8** | **95.8** | **81.0** |
| | NLTK | - | - | - | - | - | 80.2 | - | - | - | - | - | - | **93.4** | 94.5 | - |
| | PySBD | - | - | - | - | - | - | - | 86.2 | - | - | 27.4 | - | 18.2 | - | - |
| | SpaCy$_{DP}$ | - | 78.9 | - | - | 82.2 | - | - | - | - | - | - | - | 92.4 | - | - |
| | WtP | - | 76.5 | 78.0 | 89.1 | 92.3 | 80.0 | **80.7** | 88.5 | 87.0 | 60.6 | 68.9 | 68.9 | 91.5 | 94.2 | 55.6 |
| | WtP$_{T}$ | - | 84.1 | 85.6 | 91.5 | 92.4 | 81.7 | - | 88.6 | 87.9 | 80.4 | 74.3 | 68.7 | - | 94.3 | 62.2 |
| | WtP$_{PUNCT}$ | - | 90.1 | 91.5 | 95.3 | **95.6** | **86.5** | - | 93.5 | **94.0** | 88.4 | 82.2 | 74.3 | - | 96.1 | 77.3 |
| | SaT$_{+LoRA}$ | - | **92.6** | **92.8** | **95.4** | 95.5 | 86.5 | - | **94.9** | 93.8 | **89.2** | **86.7** | **77.3** | - | **96.3** | **79.6** |
| Ersatz | SpaCy$_{M}$ | - | 93.3 | 98.6 | - | - | - | - | - | - | - | - | - | - | - | - |
| | Ersatz | - | 95.0 | 98.7 | - | - | - | - | - | - | - | - | - | - | - | - |
| | Llama 3$_{8B}$ | - | 94.9 | 98.8 | - | - | - | - | - | - | - | - | - | - | - | - |
| | Command R | - | 78.3 | 82.7 | - | - | - | - | - | - | - | - | - | - | - | - |
| | SaT | - | 96.3 | 97.5 | - | - | - | - | - | - | - | - | - | - | - | - |
| | SaT$_{+SM}$ | - | **98.3** | **99.3** | - | - | - | - | - | - | - | - | - | - | - | - |
| | NLTK | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| | PySBD | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| | SpaCy$_{DP}$ | - | 74.9 | - | - | - | - | - | - | - | - | - | - | - | - | - |
| | WtP | - | 96.5 | 96.9 | - | - | - | - | - | - | - | - | - | - | - | - |
| | WtP$_{T}$ | - | 96.4 | 97.2 | - | - | - | - | - | - | - | - | - | - | - | - |
| | WtP$_{PUNCT}$ | - | **99.2** | 99.3 | - | - | - | - | - | - | - | - | - | - | - | - |
| | SaT$_{+LoRA}$ | - | 98.4 | **99.4** | - | - | - | - | - | - | - | - | - | - | - | - |

Table 30: Sentence segmentation test F1 scores on languages `la`-`pa`.

| | Model | pl | ps | pt | ro | ru | si | sk | sl | sq | sr | sv | ta | te | tg | th |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| UD | SpaCy$_{M}$ | | | | | | | | | | | | | | | |
| | Ersatz | 97.4 | - | - | 98.3 | 77.7 | - | - | - | - | - | - | 90.5 | - | - | - |
| | Llama 3$_{8B}$ | **99.3** | - | 93.5 | 93.5 | 88.5 | - | 90.6 | 98.6 | **100.0** | 96.5 | 95.5 | 96.7 | - | - | **81.4** |
| | Command R | 91.4 | - | 72.2 | 59.6 | 61.2 | - | 73.5 | 75.4 | 92.0 | 82.0 | 72.8 | 0.0 | - | - | 12.4 |
| | SaT | 95.3 | - | 96.6 | 73.2 | 82.7 | - | 94.8 | 96.8 | **100.0** | 98.0 | 95.0 | **99.5** | - | - | 72.7 |
| | SaT$_{+SM}$ | 99.2 | - | **97.2** | **99.0** | **92.4** | - | **96.5** | **99.2** | 99.1 | **99.4** | **96.2** | **99.5** | - | - | 71.2 |
| | NLTK | 97.2 | - | 91.9 | - | 78.3 | - | - | - | - | - | 93.9 | - | - | - | - |
| | PySBD | 84.8 | - | - | - | 67.9 | - | 86.2 | - | - | - | - | - | - | - | - |
| | SpaCy$_{DP}$ | 98.5 | - | 98.1 | 95.3 | 80.3 | - | - | - | - | - | 87.0 | - | - | - | - |
| | WtP | 94.4 | - | 95.9 | 80.9 | 84.9 | - | 96.0 | 95.7 | **100.0** | 97.9 | 94.7 | 96.9 | - | - | **67.3** |
| | WtP$_{T}$ | 95.5 | - | 95.6 | 93.5 | 86.8 | - | 95.8 | 96.2 | - | 98.2 | 95.0 | 97.7 | - | - | - |
| | WtP$_{PUNCT}$ | **99.3** | - | **98.4** | 99.4 | **93.1** | - | **98.1** | 99.1 | - | **99.8** | 96.5 | **100.0** | - | - | - |
| | SaT$_{+LoRA}$ | 98.9 | - | 97.8 | **99.5** | 92.7 | - | 96.9 | **99.2** | - | 99.7 | **96.5** | **99.5** | - | - | - |
| OPUS100 | SpaCy$_{M}$ | 92.0 | 2.4 | 91.6 | 91.3 | 75.2 | 75.4 | 91.6 | 92.6 | 92.6 | 94.6 | 93.1 | 36.6 | 64.1 | 69.5 | 21.7 |
| | Ersatz | 92.1 | 1.8 | - | 92.8 | 68.6 | - | - | - | - | - | - | 45.3 | - | - | - |
| | Llama 3$_{8B}$ | 93.8 | 26.1 | **94.1** | 94.9 | **89.3** | 76.9 | 94.3 | 93.8 | 92.3 | 94.6 | 94.3 | 45.3 | 65.9 | 76.1 | 66.0 |
| | Command R | 74.4 | 7.2 | 80.8 | 70.4 | 88.1 | 8.6 | 74.8 | 63.8 | 59.3 | 67.2 | 76.9 | 1.9 | 36.1 | 54.9 | 10.6 |
| | SaT | 92.4 | 68.6 | 91.7 | 91.1 | 82.8 | 79.2 | 91.4 | 91.6 | 89.7 | 94.0 | 91.3 | 60.5 | 76.5 | 79.6 | 68.0 |
| | SaT$_{+SM}$ | **95.6** | **85.6** | 93.8 | **95.9** | 84.1 | **85.4** | **95.9** | **95.3** | **95.7** | **95.9** | **95.0** | **75.0** | **86.8** | **92.3** | **72.9** |
| | NLTK | 92.5 | - | 92.2 | - | 75.8 | - | - | - | - | - | 92.5 | - | - | - | - |
| | PySBD | 17.5 | - | - | - | 64.9 | - | 29.5 | - | - | - | - | - | - | - | - |
| | SpaCy$_{DP}$ | 92.9 | - | 90.8 | 91.9 | 75.4 | - | - | - | - | - | 90.2 | - | - | - | - |
| | WtP | 91.8 | 63.3 | 89.9 | 88.5 | **80.6** | 79.5 | 89.7 | 91.3 | 88.7 | 93.8 | 90.4 | 64.3 | 77.4 | 80.1 | 66.6 |
| | WtP$_{T}$ | 92.2 | 70.4 | 91.4 | 89.0 | - | 80.1 | 92.4 | 92.7 | 89.9 | 94.3 | 92.5 | 65.7 | 77.5 | 82.7 | 69.7 |
| | WtP$_{PUNCT}$ | 95.6 | 76.1 | 95.3 | 96.7 | - | **85.4** | 95.9 | 95.0 | **95.5** | **96.4** | 95.8 | 75.4 | 83.6 | 91.0 | 71.3 |
| | SaT$_{+LoRA}$ | **96.0** | **77.6** | **95.4** | **97.3** | - | **85.9** | **96.5** | **95.3** | 95.5 | 96.1 | **95.9** | **80.8** | **85.6** | **92.1** | **73.7** |
| Ersatz | SpaCy$_{M}$ | 93.3 | 94.3 | - | 94.8 | 93.2 | - | - | - | - | - | - | 67.9 | - | - | - |
| | Ersatz | 94.9 | 93.7 | - | 96.0 | 94.2 | - | - | - | - | - | - | 95.2 | - | - | - |
| | Llama 3$_{8B}$ | 95.4 | **96.3** | - | 97.7 | 97.0 | - | - | - | - | - | - | 97.0 | - | - | - |
| | Command R | 70.1 | 70.7 | - | 71.3 | 80.2 | - | - | - | - | - | - | 1.3 | - | - | - |
| | SaT | 93.5 | 92.4 | - | 97.5 | 97.2 | - | - | - | - | - | - | 97.6 | - | - | - |
| | SaT$_{+SM}$ | **98.3** | 82.7 | - | **99.1** | **98.8** | - | - | - | - | - | - | **98.5** | - | - | - |
| | NLTK | 94.0 | - | - | - | 93.7 | - | - | - | - | - | - | - | - | - | - |
| | PySBD | 45.7 | - | - | - | 55.2 | - | - | - | - | - | - | - | - | - | - |
| | SpaCy$_{DP}$ | 94.5 | - | - | 94.4 | 94.1 | - | - | - | - | - | - | - | - | - | - |
| | WtP | 94.6 | 83.7 | - | 97.5 | 97.5 | - | - | - | - | - | - | 94.1 | - | - | - |
| | WtP$_{T}$ | 92.8 | 91.0 | - | 96.9 | 97.6 | - | - | - | - | - | - | 94.7 | - | - | - |
| | WtP$_{PUNCT}$ | 97.7 | 95.9 | - | **99.4** | **99.4** | - | - | - | - | - | - | 97.8 | - | - | - |
| | SaT$_{+LoRA}$ | **98.1** | **96.1** | - | 99.3 | 98.8 | - | - | - | - | - | - | **98.1** | - | - | - |

Table 31: Sentence segmentation test F1 scores on languages pl-th.

| | Model | `tr` | `uk` | `ur` | `uz` | `vi` | `xh` | `yi` | `yo` | `zh` | `zu` |
|---|---|---|---|---|---|---|---|---|---|---|---|
| UD | SpaCy$_{M}$ | 97.5 | 93.9 | 0.0 | - | 96.0 | - | - | 79.2 | 0.0 | - |
| | Ersatz | 96.8 | - | - | - | - | - | - | - | 89.3 | - |
| | Llama 3$_{8B}$ | 97.5 | 94.9 | 97.0 | - | 98.5 | - | - | **89.8** | 95.8 | - |
| | Command R | 61.8 | 62.1 | 82.3 | - | 92.4 | - | - | 66.1 | 91.8 | - |
| | SaT | 96.3 | 92.7 | 97.7 | - | 90.8 | - | - | 77.0 | 94.5 | - |
| | SaT$_{+SM}$ | **98.6** | **98.3** | **99.3** | - | **99.5** | - | - | 89.6 | **98.6** | - |
| | NLTK | 93.2 | - | - | - | - | - | - | - | - | - |
| | PySBD | - | - | 99.2 | - | - | - | - | - | 98.9 | - |
| | SpaCy$_{DP}$ | - | 96.5 | - | - | - | - | - | - | 99.0 | - |
| | WtP | 95.9 | 92.0 | 91.7 | - | 88.5 | - | - | 83.5 | 97.9 | - |
| | WtP$_{T}$ | 95.6 | 92.1 | 95.8 | - | 93.7 | - | - | - | 98.0 | - |
| | WtP$_{PUNCT}$ | 98.4 | **98.6** | 99.5 | - | **99.7** | - | - | - | **99.9** | - |
| | SaT$_{+LoRA}$ | **98.5** | 98.1 | **99.5** | - | 99.4 | - | - | - | 99.3 | - |
| OPUS100 | SpaCy$_{M}$ | 93.6 | 89.2 | 29.4 | 63.6 | 90.1 | 64.6 | 4.1 | 27.2 | 0.0 | 25.4 |
| | Ersatz | 92.7 | - | - | - | - | - | - | - | 54.7 | - |
| | Llama 3$_{8B}$ | 93.8 | 91.1 | 47.6 | 67.4 | 92.9 | 71.7 | 13.6 | 37.5 | 55.7 | 42.5 |
| | Command R | 72.4 | 79.8 | 26.6 | 44.8 | 77.2 | 56.1 | 8.3 | 26.6 | 58.7 | 34.9 |
| | SaT | 93.2 | 88.3 | 51.3 | 76.3 | 91.1 | 80.3 | 61.1 | **67.3** | 55.6 | 82.1 |
| | SaT$_{+SM}$ | **94.7** | **93.5** | **60.5** | **84.9** | **94.7** | **89.6** | **89.0** | 52.3 | **77.5** | **93.3** |
| | NLTK | 93.4 | - | - | - | - | - | - | - | - | - |
| | PySBD | - | - | 31.4 | - | - | - | - | - | 69.0 | - |
| | SpaCy$_{DP}$ | - | 89.8 | - | - | - | - | - | - | 68.2 | - |
| | WtP | 92.8 | 88.2 | 53.0 | 76.4 | 90.1 | 77.2 | 73.0 | **75.4** | 80.5 | 72.7 |
| | WtP$_{T}$ | 93.1 | 89.0 | 50.7 | 78.9 | 90.3 | 80.7 | 73.9 | - | 76.6 | 83.1 |
| | WtP$_{PUNCT}$ | 95.3 | 94.3 | 66.3 | 85.0 | 94.5 | 89.8 | 80.7 | - | 88.8 | 90.6 |
| | SaT$_{+LoRA}$ | **95.4** | **94.7** | **72.0** | **87.6** | **94.8** | **90.8** | **86.6** | - | **90.5** | **92.0** |
| Ersatz | SpaCy$_{M}$ | 95.2 | - | - | - | - | - | - | - | 0.0 | - |
| | Ersatz | 96.2 | - | - | - | - | - | - | - | 87.4 | - |
| | Llama 3$_{8B}$ | 94.3 | - | - | - | - | - | - | - | **94.5** | - |
| | Command R | 71.8 | - | - | - | - | - | - | - | 70.5 | - |
| | SaT | 93.5 | - | - | - | - | - | - | - | 84.1 | - |
| | SaT$_{+SM}$ | **97.5** | - | - | - | - | - | - | - | 90.5 | - |
| | NLTK | 92.6 | - | - | - | - | - | - | - | - | - |
| | PySBD | - | - | - | - | - | - | - | - | 92.7 | - |
| | SpaCy$_{DP}$ | - | - | - | - | - | - | - | - | 95.9 | - |
| | WtP | 92.8 | - | - | - | - | - | - | - | 93.7 | - |
| | WtP$_{T}$ | 93.0 | - | - | - | - | - | - | - | 93.4 | - |
| | WtP$_{PUNCT}$ | **98.3** | - | - | - | - | - | - | - | **97.9** | - |
| | SaT$_{+LoRA}$ | 98.2 | - | - | - | - | - | - | - | 95.0 | - |

Table 32: Sentence segmentation test F1 scores on languages `tr`-`zu`.