Where's the Point? Self-Supervised Multilingual Punctuation-Agnostic Sentence Segmentation

Benjamin Minixhofer[*1] **Jonas Pfeiffer**[†2] **Ivan Vulić**[†3]
[1]Cohere for AI [2]Google DeepMind [3]University of Cambridge

Abstract

Many NLP pipelines split text into sentences as one of the crucial preprocessing steps. Prior sentence segmentation tools either rely on punctuation or require a considerable amount of sentence-segmented training data: both central assumptions might fail when porting sentence segmenters to diverse languages on a massive scale. In this work, we thus introduce a *multilingual punctuation-agnostic* sentence segmentation method, currently covering 85 languages, trained in a self-supervised fashion on unsegmented text, by making use of newline characters which implicitly perform segmentation into paragraphs. We further propose an approach that adapts our method to the segmentation in a given corpus by using only a small number (64-256) of sentence-segmented examples. The main results indicate that our method outperforms all the prior best sentence-segmentation tools by an average of 6.1% F1 points. Furthermore, we demonstrate that proper sentence segmentation has a point: the use of a (powerful) sentence segmenter makes a considerable difference for a downstream application such as machine translation (MT). By using our method to match sentence segmentation to the segmentation used during training of MT models, we achieve an average improvement of 2.3 BLEU points over the best prior segmentation tool, as well as massive gains over a trivial segmenter that splits text into equally sized blocks.

1 Introduction

Sentences are ubiquitous in NLP. Many datasets are made up of annotated sentences (de Marneffe et al., 2021; Aharoni et al., 2019; Conneau et al., 2018, *inter alia*) and models often expect individual sentences as input (Reimers and Gurevych, 2019, 2020; Liu et al., 2021; Tiedemann and Thottingal, 2020, *inter alia*). This mandates a need for tools to segment text into sentences: a requirement that typically slips under the radar of many modern NLP systems (Wicks and Post, 2022).

<table>
<tr><th>Collection</th><th>Sentences</th></tr>
<tr><td>UD</td><td>This is the high season for tourism; | between December and April few people visit and many tour companies and restaurants close down.</td></tr>
<tr><td>OPUS100</td><td>'I couldn't help it,' said Five, in a sulky tone; 'Seven jogged my elbow.' | On which Seven looked up and said, 'That's right, Five! Always lay the blame (...)!'</td></tr>
<tr><td>Ersatz</td><td>"A lot of people would like to go back to 1970," before program trading, he said. | "I would like to go back to 1970. | But we're not going back (...)"</td></tr>
</table>

Table 1: Example sentences of different collections. The pipe ('|') indicates sentence boundaries.

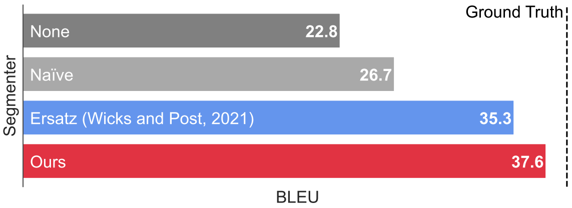

Figure 1: Impact of sentence segmentation on BLEU scores in MT. Full details in Table 5.

Theoretically, a sentence can be defined as a sequence of grammatically linked words conveying a complete thought (Sweet, 2014). In practice, there is ambiguity in what can be considered one sentence, as illustrated in Table 1. Do nested syntactic structures (e.g. through quotation marks) make up *one* sentence, or *multiple* ones? What about parentheses and enumerations? Sometimes, even colons and semicolons are considered sentence boundaries. In addition to the ambiguity in what makes up a sentence, there is practical difficulty in devising a tool to segment text into sentences. In many languages, punctuation is not limited to appearing at sentence boundaries, being used also for, e.g., acronyms and abbreviations. Other languages, such as Thai, do not use punctuation at all. In languages which do

[*] Work done during the time BM interned at Cohere.
[†] Equal senior authorship.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 7215–7235
July 9-14, 2023 ©2023 Association for Computational Linguistics

use punctuation, noisy user-generated text may still lack consistent punctuation (Kagan, 1980).

As surveyed later in §2, tools for sentence segmentation typically rely on sentence boundaries to occur exclusively at punctuation marks. This makes them applicable only to well-punctuated text in languages with sentence-ending punctuation. Some existing sentence segmentation tools do not rely on punctuation (Zhou et al., 2016; Honnibal and Johnson, 2015); they, however, need sentence-segmented training data, which makes them difficult to apply to low-resource setups and languages.

In order to address these core challenges, in this work we present a fully self-supervised sentence segmentation method which does not rely on punctuation, and is thus applicable to a wide spectrum of languages and different corpora. We pragmatically define a sentence as a sequence of characters which could plausibly be followed by a newline, following the intuition that paragraph breaks can never occur within a sentence. We then train a bidirectional character-level language model (ChLM) on text stripped of newline characters to predict, for each character, whether it was followed by a newline in the original text. A single configurable threshold then determines whether each newline-likelihood score should or should not be treated as a sentence boundary. Our method, though self-supervised, on average matches the performance of the best prior (supervised) segmentation tools.

In addition, we take into account the fact that sentence segmentation is subjective and might be considered corpus-specific. To this end, we devise an auxiliary punctuation-prediction objective which allows adapting our model to the sentence segmentation in a given corpus in a data-efficient way. This leads to an improvement of an average 6.1% F1 points over prior tools. We find that, while the precise definition of a sentence may not be important, consistency between training and inference is crucial: using our method to match sentence segmentation to the segmentation used during training leads to an improvement of an average 2.3 points in BLEU score of machine translation systems across 14 languages, as summarised in Figure 1.

Contributions. **1)** We introduce *'Where's the Point'* (WtP), a method for self-supervised sentence segmentation without relying on punctuation and without language-specific assumptions. **2)** We present a data-efficient way to adapt WtP models to the sentence segmentation in a given corpus using a small number (e.g., 64) of sentence-segmented examples. **3)** We train state-of-the-art WtP models in five different sizes covering 85 languages. Our code and models are publicly available at `github.com/bminixhofer/wtpsplit`.

Type	Method	NP?	Reference
1. RB	Moses		Koehn et al. (2007)
	SpaCy$_{\text{SENT}}$		Honnibal et al. (2020)
	PySBD		Sadvilkar and Neumann (2020)
2. SS	Riley		Riley (1989)
	Satz		Palmer and Hearst (1997)
	Splitta		Gillick (2009)
	SpaCy$_{\text{DP}}$	✓	Honnibal and Johnson (2015)
	Ersatz		Wicks and Post (2021)
3. US	Punkt		Kiss and Strunk (2006)
	Ersatz$_{\text{U}}$		Wicks and Post (2021)
	WtP	✓	Ours

Table 2: Taxonomy of sentence segmentation methods into *rule-based* (RB), *supervised statistical* (SS) and *unsupervised statistical* (US) approaches and whether they can segment text with no punctuation (NP?).

2 Related Work and Baseline Systems

Sentence Segmentation. Work on sentence segmentation can be divided into **1)** rule-based, **2)** supervised statistical, and **3)** unsupervised statistical approaches. Table 2 shows a taxonomy of sentence segmentation methods, discussed in what follows.

1. Rule-Based Methods rely on handcrafted rules to segment text into sentences. The sentence segmenters in Moses (Koehn et al., 2007) and SpaCy (SpaCy$_{\text{SENT}}$; Honnibal et al., 2020) split on every punctuation character, unless it occurs within a handcrafted set of exceptions (e.g., abbreviations and acronyms). PySBD (Sadvilkar and Neumann, 2020) uses a set of exceptions as well as regular expression rules. Rule-based approaches require extensive manual effort for every language, which makes scaling to many languages difficult.

2. Supervised Statistical Methods use a corpus of already sentence-segmented text to learn segmentation. One of the first approaches in this area was by Riley (1989), where they decide for each punctuation mark in a text, whether it constitutes a sentence boundary or not. They do so by learning a decision tree from lexical features of the context around the punctuation mark. Satz (Palmer and Hearst, 1997) and Splitta (Gillick, 2009) follow the same paradigm, but use a neural network with part-of-speech features, and an SVM with lexical features, respectively. The above approaches all suffer from the same limitation: the set of plausible sentence boundaries is made up of the set of punctuation marks, that is, a non-punctuation character can never make up a sentence boundary. This is a major restriction especially for text which is not well-punctuated, and text in languages which do not require punctuation (e.g. Thai).

The dependency parser in the SpaCy library (SpaCy$_{\text{DP}}$; Honnibal et al., 2020) is among the first to lift this restriction. SpaCy$_{\text{DP}}$ jointly learns dependency parsing and sentence segmentation on a labelled corpus using a transition-based parser without special treatment of punctuation (Honnibal and Johnson, 2015). Ersatz (Wicks and Post, 2021) modernizes Riley (1989)'s paradigm by using a Transformer (Vaswani et al., 2017) with subwords as context around punctuation. This again requires sentence boundaries to exclusively occur at punctuation marks.

3. Unsupervised Statistical Methods aim to learn segmentation using raw unsegmented text only. Punkt (Kiss and Strunk, 2006) identifies abbreviations, initials and ordinal numbers in an unsupervised way by using character length and internal punctuation, among other features. All punctuation marks occurring outside of these are considered sentence boundaries. More recently, in addition to their supervised model, Wicks and Post (2021) introduce a self-supervised model which follows the same paradigm as Ersatz, but is instead trained on punctuation preceding paragraph breaks. This allows training without any labelled data, since newline characters naturally segment text into paragraphs. We refer to this model as Ersatz$_{\text{U}}$.

Our method is most closely related to Ersatz$_{\text{U}}$. We also use newlines (i.e. paragraph breaks) as a signal to learn segmentation. In contrast to Ersatz$_{\text{U}}$, our method does not require punctuation. Also related, though specific to English, is Moore (2021) which uses n-gram occurences around paragraph breaks to predict sentence boundaries.

Character-Level Pretrained LMs. Pretraining LMs on a large amount of text in a self-supervised way before training on the target task was a paradigm shift induced by BERT (Devlin et al., 2019). Pretrained LMs typically use the Transformer architecture (Vaswani et al., 2017). Pretrained LMs often represent the input as subword tokens (Kudo and Richardson, 2018; Sennrich et al., 2016). However, recent efficiency improvements have enabled directly using characters as the input (Clark et al., 2022; Tay et al., 2022). Character-level LMs (ChLMs) are well suited for multilingual sentence segmentation since (i) merging characters into subword tokens restricts sentence boundaries to end-of-token positions and (ii) subword-based tokenization leads to problems of vocabulary allocation in multilingual LMs (Rust et al., 2021).

3 Method

In light of the ambiguity in what makes up a sentence, we resort to the following definition:

(D1) *A sentence is any sequence of characters which could plausibly be followed by a newline.*

This pragmatically driven definition corresponds closely to our intuitive understanding of a sentence due to two statements we assume to be true about sufficiently clean text: (i) a newline can generally not occur within a sentence and (ii) a newline can generally occur after any sentence.

This definition turns sentence segmentation into a character-level language modeling task. However, a causal language modeling objective as used in contemporary generative LMs (Brown et al., 2020; Radford et al., 2019, 2018) would restrict the model to unidirectional context. We devise a corruption method to allow using a bidirectional LM to model *newline-likelihood*. Let $\boldsymbol{c}$ denote the sequence of characters making up some corpus. $\boldsymbol{c}$ is preprocessed by stripping consecutive newline characters, and adding a space after every newline in languages which use whitespace to separate sentences. First, we corrupt the text by removing newline characters (`\n`) from $\boldsymbol{c}$, resulting in $\boldsymbol{x}$:

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

The target is to identify which characters were followed by a newline in the original sequence:[1]

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

Let the contextualized representations $\boldsymbol{h} = f_\theta(\boldsymbol{x})$ and predictions $\hat{\boldsymbol{y}} = \text{sigmoid}(g_\theta(\boldsymbol{h}))$ be produced by a character-level language model f_θ and a prediction head g_θ parameterized by θ. The loss is the standard binary cross-entropy between $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$:

$$\mathcal{L}_\theta^{\text{main}} = -\frac{1}{|\boldsymbol{y}|} \sum_{i=0}^{|\boldsymbol{y}|-1} y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)$$

[1] Note that c_i and c_{i+1} index into the original sequence $\boldsymbol{c}$.

The output $\hat{\boldsymbol{y}}$ can be interpreted as an estimate for the probability of a newline to occur after any character. This objective is comparable to objectives used in text-editing models (Malmi et al., 2022).[2] It remains to find a suitable threshold α such that characters with $\hat{y}_i \geq \alpha$ are considered a sentence boundary. In the simplest case, α can be set to a small constant value such as 0.01, where a higher α gives rise to more conservative segmentation. We denote the model variants with constant threshold as WtP_U. WtP_U models can segment text according to the general Definition D1. In practice, models are trained on different corpora following different definitions of what makes up a sentence. This can be addressed to some extent by selecting the threshold α to maximise performance on the target corpus (WtP_T). To allow for more sophisticated adaptation, we introduce an auxiliary objective.

3.1 Auxiliary Punctuation-Prediction

As an optional auxiliary objective, we predict the likelihood for punctuation characters among a predefined set P to follow any character in the input text.[3] We remove characters among P from the text with probability p. Let $\boldsymbol{p} \sim \text{Bernoulli}(p)^{|\boldsymbol{c}|}$ be a random binary mask. $\boldsymbol{x}'$ is corrupted in the same way as $\boldsymbol{x}$, with additional stochastic removal of punctuation characters among P. We **highlight** the additional criterion over Equation (1) in the following Equation (3).

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

In addition, we never remove two consecutive characters in Equation (3) to avoid ambiguity.[4] The auxiliary labels $\boldsymbol{z}$ indicate, for the remaining characters, which (if any) character among P followed them in the original sequence.

$$\boldsymbol{z} = \left\{ \begin{array}{ll} c_{i+1} & \text{if } c_{i+1} \in P \\ 0 & \text{otherwise} \end{array} \;\middle|\; c_i \in \boldsymbol{x}' \right\} \tag{4}$$

If the auxiliary objective is used, $\boldsymbol{y}$ and $\boldsymbol{h}$ are obtained using $\boldsymbol{x}'$ instead of $\boldsymbol{x}$ such that the same contextualized representations can be used for both objectives. Given predictions $\hat{\boldsymbol{z}} = \text{softmax}(q_\theta(\boldsymbol{h}))$, $\hat{\boldsymbol{z}} \in \mathbb{R}^{|\boldsymbol{x}'| \times |P|+1}$ where q_θ is an auxiliary prediction head parameterized by θ, the auxiliary loss is defined as the categorical cross-entropy between $\boldsymbol{z}$ and $\hat{\boldsymbol{z}}$ as follows:

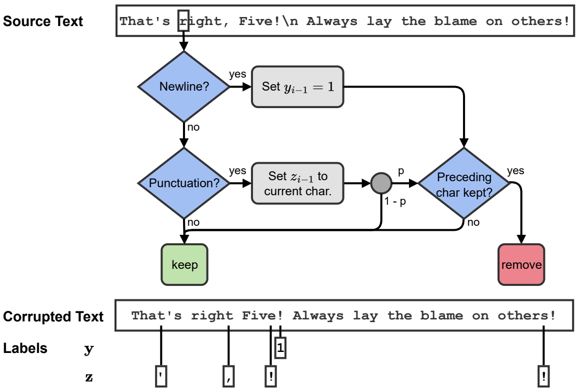
Source Text
That's right, Five!\n Always lay the blame on others!
Newline?
yes
no
Set $y_{i-1} = 1$
Punctuation?
yes
no
Set z_{i-1} to current char.
p
1 - p
Preceding char kept?
yes
no
keep
remove
Corrupted Text
That's right Five! Always lay the blame on others!
Labels
y
1
z
'
,
!
!

Figure 2: Flow diagram of the corruption process (§3.1).

$$\mathcal{L}_\theta^{\text{aux}} = -\frac{1}{|\boldsymbol{z}|} \sum_{i=0}^{|\boldsymbol{z}|-1} \sum_{j=0}^{|P|} \log \hat{z}_{i,j} \cdot \mathbb{I}(\text{i}(z_i) = j). \tag{5}$$

Here, i assigns a unique index to every element of $P \cup \{0\}$. The total loss $\mathcal{L}_\theta$ is the sum of the primary objective of predicting newlines and the auxiliary objective of predicting punctuation.

$$\mathcal{L}_\theta = \mathcal{L}_\theta^{\text{main}} + \mathcal{L}_\theta^{\text{aux}} \tag{6}$$

Figure 2 summarizes the corruption process. Sentence segmentation can be adapted to a target corpus with characters $\boldsymbol{c}_s$ and sentence boundaries $\boldsymbol{y}_s$ by finding the optimal coefficients $\boldsymbol{a}^* \in \mathbb{R}^{|P|+1}$ of a logistic regression over punctuation logits, denoted as $h_{\boldsymbol{a}}(\boldsymbol{c}) = \text{sigmoid}(q_\theta(f_\theta(\boldsymbol{c}))\boldsymbol{a})$.

$$\boldsymbol{a}^* = \text{argmin}_{\boldsymbol{a}} \left\| \begin{array}{l} y_s \odot \log h_{\boldsymbol{a}}(\boldsymbol{c}_s) + \\ (1 - y_s) \odot (1 - \log h_{\boldsymbol{a}}(\boldsymbol{c}_s)) \end{array} \right\| \tag{7}$$

This formulation is motivated by the hypothesis that the position of sentence boundaries is primarily determined by the probability distribution over punctuation marks at that position. It is a convex optimization problem. Effectively, it fits a one-layer neural network parameterized by $\boldsymbol{a}$ on the punctuation logits $q_\theta(f_\theta(\boldsymbol{c}))$ to predict sentence boundaries. We denote models adapted to a corpus using this method as WtP_PUNCT. We show how it leads to data-efficient adaptation later in §5.

[2] A model trained with this objective can be interpreted as a text-editing model with the singular operation *'insert newline'*.

[3] P can be, e.g., the union of the n most common punctuation characters in every language.

[4] This constraint allows reconstructing the original sequence (modulo repetitions) from $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z}$. This would be prevented by removing two consecutive characters since $\boldsymbol{y}$ and $\boldsymbol{z}$ can only represent insertion of a single character.

4 Experimental Setup

4.1 Training Setup

We train a multilingual character-level language model on text in 85 languages. We sample text from all languages uniformly from the mC4 corpus (Raffel et al., 2019). In languages which use sentence-ending punctuation (every language besides Thai), we sample paragraphs such that a maximum of 10% of paragraphs do not end in punctuation. For a list of languages, see Appendix A.

We use the pretrained CANINE-S ChLM (Clark et al., 2022) as the starting point. To make the model sufficiently fast for sentence segmentation, we remove layers 4-12 from CANINE-S, resulting in a 3-layer model (we also experiment with larger sizes in §5). We add language adapters as described in Pfeiffer et al. (2022) to efficiently increase per-language capacity in a modular fashion while keeping the same underlying model.

We continue training from the original CANINE-S ChLM using our objective for 400k training steps with 512 characters per example and a batch size of 512.[5] We warm up the randomly initialized language adapters for 5k steps with a constant learning rate of $1e{-4}$ (keeping other parameters frozen), then start training the entire model with a linearly increasing learning rate from zero to $1e{-4}$ for the next 5k steps.[6] We linearly decay the learning rate to zero over the remaining 390k steps. We use the AdamW optimizer (Loshchilov and Hutter, 2019).

For the auxiliary objective, we choose P to be the union of the 30 most common punctuation characters in every language (125 characters in total), and the removal probability $p = 0.5$.

4.2 Evaluation

To evaluate our method, we determine how closely our sentence segmentation corresponds to the segmentation in different sentence-segmented corpora. Although the segmentation may vary due to what annotators of different corpora consider as a standalone sentence, it should match in the many cases which are largely unambiguous. We obtain sentence-segmented corpora from three different sources as follows.

1. Universal Dependencies (UD; de Marneffe et al., 2021; Nivre et al., 2020) is a collection of datasets annotating grammar (POS, morphological features and syntactic dependencies) in many different languages. Used UDv2.10 treebanks include segmentation into words and sentences.

2. OPUS100 (Zhang et al., 2020) is a collection of sentences from various sources including subtitles and news sampled from OPUS (Tiedemann, 2012). It consists of sentences in 100 languages with corresponding parallel sentences in English. Sentences in OPUS-100 sometimes lack proper punctuation, making it a challenging benchmark for sentence segmentation (Sadvilkar and Neumann, 2020).

3. Ersatz (Wicks and Post, 2021) introduces a collection of sentence-segmented text along with their sentence segmentation tool. The corpus consists primarily of sentences from the WMT shared tasks (Barrault et al., 2020; Bojar et al., 2019) with manual corrections by the authors. Contrary to Wicks and Post (2021), we do not remove sentences without sentence-ending punctuation.

Evaluation details are shown in Appendix A. We use the test sets for evaluation. For adaptation via WtP_{T} and $\text{WtP}_{\text{PUNCT}}$, we use the training sets. We evaluate by F1 score, where a character belongs to the positive class if it is followed by a sentence boundary. We set $\alpha = 0.01$ for WtP_{U}.[7]

Baselines. We compare against $\text{SpaCy}_{\text{SENT}}$ and PySBD as representatives of rule-based methods, SpaCy_{DP} as supervised punctuation-agnostic method, and Ersatz as a recent Transformer-based method; see §2 again for their brief descriptions.

Languages. For clarity, in the main paper we present results on Arabic, Czech, German, English, Spanish, Finnish, Hindi, Japanese, Georgian, Latvian, Polish, Thai, Xhosa and Chinese as a linguistically diverse subset ranging from low-resource (Georgian, Xhosa) to high-resource (German, English) languages. For the full evaluation in the remaining languages, see Appendix A.

5 Results and Discussion

Main results are shown in Table 3, results for all languages in Appendix A. For UD and Ersatz, WtP_{U}, though self-supervised, matches the performance of the best prior tool (usually SpaCy_{DP} or Ersatz) on most languages. It falls behind on Hindi (hi),

[5] This is character-wise equivalent to ~10% of CANINE-S pretraining. It takes ~3 days on one TPUv3 with 8 cores.

[6] The learning rate was selected from the set $\{1e{-4}, 5e{-5}, 1e{-5}\}$ to minimize loss on a held-out set of mC4. The learning rate schedule was not tuned.

[7] The threshold was selected using the English UD data, so it is purely self-supervised for every language besides English.

		ar	cs	de	en	es	fi	hi	ja	ka	lv	pl	th	xh	zh
UD	SpaCy$_{\text{SENT}}$	73.4	83.7	89.9	89.3	89.3	87.3	95.2	96.3	-	92.4	93.0	-	-	96.7
	PySBD	29.5	-	79.9	75.5	46.2	-	99.7	97.9	-	-	85.0	-	-	98.9
	SpaCy$_{\text{DP}}$	-	-	**96.9**	91.7	99.0	95.5	-	**98.0**	-	-	98.5	-	-	98.2
	Ersatz	81.0	89.5	92.4	89.4	97.5	92.9	99.5	93.4	-	96.8	97.5	-	-	88.8
	Punkt	-	89.2	92.6	91.2	98.6	92.8	-	-	-	-	97.4	-	-	-
	WtP$_{\text{U}}$	82.1	92.5	95.7	95.0	96.7	92.8	96.5	93.9	-	96.5	94.9	**69.5**	-	98.1
	WtP$_{\text{T}}$	87.5	92.6	95.8	95.0	97.1	93.0	97.1	95.8	-	96.4	95.8	-	-	98.2
	WtP$_{\text{PUNCT}}$	**88.2**	**95.5**	96.7	**96.7**	**99.7**	**98.2**	**99.9**	**98.0**	-	**99.1**	**99.4**	-	-	**99.8**
OPUS100	SpaCy$_{\text{SENT}}$	51.4	84.6	70.0	86.8	78.4	91.4	54.0	43.6	-	58.0	89.4	-	-	64.1
	PySBD	39.1	-	66.6	59.8	68.0	-	23.1	42.9	-	-	17.6	-	-	69.8
	SpaCy$_{\text{DP}}$	-	-	74.5	89.4	88.4	92.9	-	42.2	-	-	92.9	-	-	69.4
	Ersatz	59.7	86.2	73.2	87.7	90.0	92.9	58.5	28.3	-	77.6	92.2	-	-	54.7
	Punkt	-	86.5	73.5	88.6	90.2	93.5	-	-	-	-	92.8	-	-	-
	WtP$_{\text{U}}$	66.2	88.5	78.5	91.3	90.8	91.5	66.7	44.9	91.9	79.6	92.4	68.8	78.7	81.0
	WtP$_{\text{T}}$	66.4	90.8	85.8	90.3	92.1	93.1	66.1	80.5	91.7	86.5	92.8	71.5	81.9	77.8
	WtP$_{\text{PUNCT}}$	**77.2**	**95.2**	**90.1**	**95.0**	**95.4**	**96.1**	**77.5**	**87.4**	**93.2**	**91.9**	**96.0**	**72.9**	**90.4**	**89.2**
Ersatz	SpaCy$_{\text{SENT}}$	89.4	84.1	89.9	89.8	85.0	94.7	89.9	84.7	-	89.8	77.6	-	-	90.6
	PySBD	47.9	-	95.5	74.2	84.6	-	87.8	87.7	-	-	46.1	-	-	92.7
	SpaCy$_{\text{DP}}$	-	-	96.3	98.3	96.4	95.2	-	91.2	-	-	94.4	-	-	95.8
	Ersatz	92.9	96.8	95.6	97.6	96.7	95.9	**96.9**	85.9	-	98.7	95.1	-	-	87.6
	Punkt	-	96.8	95.5	97.8	96.6	95.7	-	-	-	-	94.3	-	-	-
	WtP$_{\text{U}}$	87.8	93.7	95.7	96.8	98.8	97.5	94.4	81.5	-	97.2	94.8	-	-	93.5
	WtP$_{\text{T}}$	88.9	94.1	96.0	96.9	97.8	97.3	94.7	82.6	-	97.3	92.8	-	-	93.7
	WtP$_{\text{PUNCT}}$	**92.9**	**98.9**	**99.3**	**98.7**	**99.5**	**99.4**	96.4	**94.8**	-	**99.4**	**98.0**	-	-	**97.8**

Table 3: Sentence segmentation F1 scores. For Georgian (ka), Thai (th), and Xhosa (xh), no Ersatz and UD corpora are available. For Thai, no OPUS100 training data is available. We adapt WtP$_{\text{T}}$ and WtP$_{\text{PUNCT}}$ to each corpus using the corresponding training datasets. Bold numbers indicate the best results for each language and dataset.

Japanese (ja) and Chinese (zh). While the majority of the languages we train on uses Latin punctuation characters, Hindi, Japanese and Chinese use punctuation in their own script, which could cause this deficit. In addition, Japanese and Chinese do not use whitespace between words and sentences which makes segmentation more challenging. The deficit can be resolved by supervised adaptation: WtP$_{\text{PUNCT}}$ surpasses the prior best on 19 out of 22 of the UD and Ersatz datasets (2.1% on average), including 4 out of 6 of the Hindi, Japanese and Chinese datasets. On OPUS100, WtP$_{\text{U}}$ already outperforms the best prior tool in 10 out of 12 languages (3.2% on average). Adaptation via WtP$_{\text{T}}$ and WtP$_{\text{PUNCT}}$ increases the gap to 7.8% and 14.1%, respectively. The strong increase in performance on OPUS100 compared to prior tools may be caused by WtP being pretrained on web text, which could be better suited for the generally noisy text in OPUS100.

	ORCHID	UD	OPUS100	***Macro Avg.***
PyThaiNLP	55.6	64.7	62.5	60.9
WtP$_{\text{U}}$	67.9	69.5	68.8	68.7
WtP$_{\text{T:OPUS100}}$	**69.2**	**78.0**	71.5	**72.9**
WtP$_{\text{PUNCT:OPUS100}}$	51.8	77.1	**72.9**	67.3

Table 4: Thai segmentation F1 scores. Score on full dataset for ORCHID, on test sets for UD and OPUS100. WtP$_{*\text{:OPUS100}}$ denotes WtP adapted to the OPUS100 corpus; we do not use ORCHID and UD for adaptation.

Punctuation-Free Segmentation in Thai. Thai is especially relevant to our method since it is the only language in our set which does not, in general, use punctuation to separate sentences. Most prior sentence segmentation tools rely on punctuation (Sadvilkar and Neumann, 2020; Wicks and Post, 2021; Kiss and Strunk, 2006), which has led to the development of Thai-specific segmentation tools in a separate strand of work (Saetia et al., 2019; Nararatwong et al., 2018; Zhou et al., 2016; Phatthiyaphaibun et al., 2016). Our method is the first to segment Thai and other languages using the same methodology. To verify WtP performs as expected on Thai, we evaluate it on the Thai ORCHID corpus (Sornlertlamvanich et al., 1997) in addition to UD and OPUS100, and compare against the Open-Source PyThaiNLP toolkit's sentence segmentation (Phatthiyaphaibun et al., 2016):

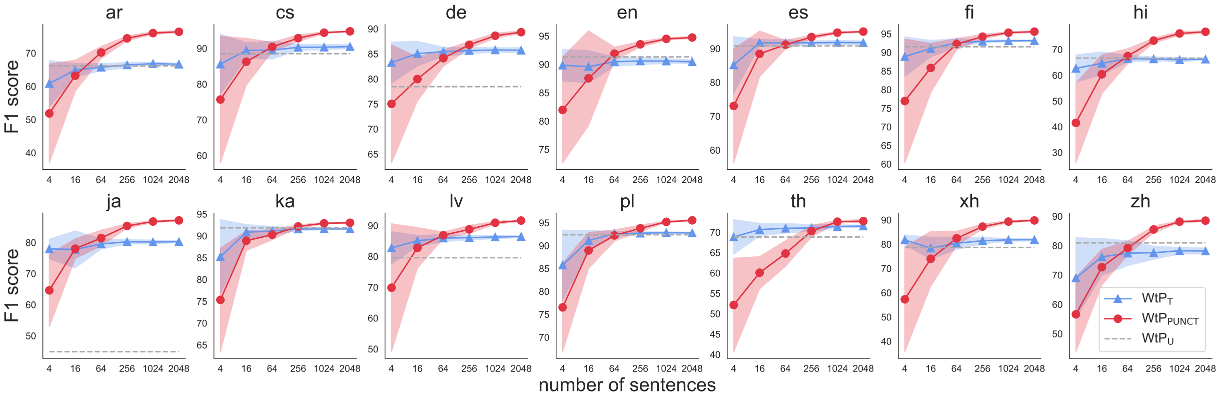

Figure 3: F1 score vs. number of sentences used for adaptation on the OPUS100 datasets. Numbers are averaged over 20 different sentence samples of the respective sizes. Shaded regions indicate one standard deviation.

PyThaiNLP v3.1.1.. Results are shown in Table 4. While scores are lower than for most other languages due to the more challenging nature of sentence segmentation in Thai, WtP consistently outperforms PyThaiNLP. We also find that transfer from WtP_PUNCT tuned on OPUS100 to ORCHID and UD is limited; see Appendix D.

Few-Shot Adaptation. We now analyse how many sentences are needed to successfully adapt WtP to a target corpus. Since we observed the most substantial need for adaptation on OPUS100, we focus on the OPUS100 datasets. Figure 3 shows performance of WtP_T and WtP_PUNCT w.r.t. the number of sentences used for adaptation. Between 64 and 256 sentences, WtP_PUNCT consistently starts outperforming the self-supervised WtP_U baseline and WtP_T in all languages except Thai.[8] It still benefits from more data up to ~2k sentences. Using less than 64 sentences, WtP_T outperforms WtP_PUNCT, but in some cases does not surpass WtP_U. In Thai, adaptation via WtP_PUNCT is less effective. However, with enough data, it can still improve over WtP_T.

src	tgt	True	None	Naïve	Ersatz	WtP_PUNCT
ar	en	44.0	16.4	29.4	38.0	**40.8**
cs	en	37.7	30.6	25.5	36.6	**37.6**
de	en	37.5	31.8	31.5	36.8	**37.2**
en	eo	48.8	21.8	26.7	44.8	**47.2**
es	en	43.7	37.3	33.5	43.1	**43.2**
fi	en	28.9	20.8	20.1	28.3	**28.8**
hi	en	39.2	8.7	21.0	28.1	**32.7**
ja	en	18.6	3.2	9.0	7.9	**16.2**
ka	en	22.3	0.6	11.6	-	**21.5**
lv	en	54.1	30.3	43.4	51.3	**53.6**
pl	en	29.1	22.4	18.7	28.5	**29.0**
th	en	27.1	4.4	21.4	-	**22.6**
xh	en	61.2	17.4	31.7	-	**56.8**
zh	en	48.4	27.4	35.6	44.8	**47.0**
Macro Avg.		39.1	22.8	26.7	35.3	**37.6**

Table 5: Impact of sentence segmentation on machine translation BLEU score. *True* indicates the ground truth segmentation. Average excludes languages with missing Ersatz scores. English is translated to Esperanto (eo).

Downstream Impact of Sentence Segmentation. Prior sentence segmentation tools limit themselves to intrinsic performance evaluation (Gillick, 2009; Sadvilkar and Neumann, 2020; Wicks and Post, 2021). But does proper sentence segmentation actually matter for downstream tasks? Usually sentence segmentation is done as the first step in a pipeline i.e. text is first split into sentences, then every sentence is passed one-by-one through some model to solve a downstream task. We quantify the impact of sentence segmentation on one downstream task: machine translation (MT). We obtain MT models for 14 languages trained on OPUS100 data from OPUS-MT (Tiedemann and Thottingal, 2020). We simulate a real-world scenario by partitioning the OPUS100 test data into paragraphs consisting of 10 sentences each. We sentence-segment the paragraphs using different tools, pass every sentence through the MT model, and conjoin the resulting translation. In addition, we compare against two baselines: (i) *None*, where the entire paragraph is passed to the MT model at once and (ii) *Naïve*, where the paragraph is segmented into 10 equally long sequences of words or characters.[9]

We evaluate the predicted translation via BLEU score (Papineni et al., 2002) implemented in Sacre-

[8] Assuming annotation speed of ~4 sentences per minute, it would take 15 minutes to 1 hour to annotate 64-256 sentences.

[9] Words if segmentation into words is trivial (i.e. the language uses whitespace between words), otherwise characters.

	English			Bengali			*Macro Avg.*
	C	P	Q	C	P	Q	
CANINE-S	66.6	80.2	72.3	35.7	61.2	29.9	57.7
WtP$_{\text{PUNCT}}$	61.3	76.0	73.3	**38.9**	**72.8**	39.2	60.3
WtP$_{\text{FINETUNE}}$	**69.8**	**82.7**	**77.9**	36.6	72.5	**40.9**	**63.4**

Table 6: Punctuation restoration F1 score across *comma* (C), *period* (P) and *question mark* (Q) of CANINE-S, WtP$_{\text{PUNCT}}$ and the fully finetuned WtP model (WtP$_{\text{FINETUNE}}$). Train and test splits as in Alam et al. (2020). Details in Appendix C.

BLEU (Post, 2018).[10] Results are shown in Table 5. We find that sentence segmentation is necessary for models trained on the sentence-level: Passing the entire paragraph to the model at once causes a drop of 16.3 BLEU score compared to the ground truth segmentation. Furthermore, the choice of sentence segmentation tool makes a difference: WtP$_{\text{PUNCT}}$ outperforms Ersatz by an average 2.3 BLEU points.

Previous work has found no clear difference between Ersatz, Moses, Punkt and SpaCy$_{\text{SENT}}$ for German-English Machine Translation on OPUS (Wicks and Post, 2022). We find that, although German-English translation exhibits one of the lowest differences, using WtP to match segmentation to the segmentation used during training can lead to consistent and sometimes large improvements across all languages in our evaluation. For qualitative analysis, see Appendix B.

Application to the Punctuation Restoration Task. Our auxiliary punctuation-prediction objective is closely related to the task of punctuation restoration (Păiş and Tufiş, 2022), where the position of punctuation characters is predicted in unpunctuated text. We evaluate the capacity of WtP for punctuation restoration on the English IWSLT dataset (Che et al., 2016; Cettolo et al., 2013), as well as a Bengali dataset provided by Alam et al. (2020). We compare against the pretrained CANINE-S (the starting point for the WtP models) fine-tuned on punctuation restoration data in the respective language. Results are shown in Table 6. CANINE-S with continued training via WtP outperforms the off-the-shelf CANINE-S by an average 5.7% F1 points.[11] Fitting a logistic regression on punctuation logits (WtP$_{\text{PUNCT}}$) outperforms the fully finetuned CANINE-S on Bengali, but not on English. While prior work approaches punctuation restoration by corrupting a curated small-scale corpus (Nguyen et al., 2019; Ueffing et al., 2013), we are the first to show that Web-scale pretraining via corruption can improve punctuation restoration.

Variation	WtP$_{\text{U}}$	WtP$_{\text{T}}$	WtP$_{\text{PUNCT}}$
Original	89.0	90.4	94.9
No language adapters	89.0	90.3	94.6
No punctuation-specific sampling	87.9	90.5	94.8
No aux. punctuation-prediction	88.3	90.0	-
+ punctuation corruption	88.4	89.8	-
Reduced pretraining data size			
75% subsample	89.1	90.6	94.9
50% subsample	89.2	90.6	94.8
25% subsample	89.1	90.4	94.8
Scaled amount of layers			
1 layer	88.8	90.1	94.4
3 layers (Original)	89.0	90.4	94.9
6 layers	89.3	90.8	95.1
9 layers	89.8	91.1	95.3
12 layers	89.9	91.2	95.5

Table 7: Ablation studies and sensitivity analysis w.r.t. amount of layers and amount of pretraining data.

Ablation Studies. We now quantify the impact of multiple design choices of WtP in Table 7. *No language adapters* expectedly decreases performance. We remove language adapters by using Pfeiffer et al. (2022)'s SHARED setting (keeping the language adapter layers but sharing parameters between languages) to match FLOPs. *No punctuation-specific sampling* i.e. not sampling paragraphs in punctuated languages such that a maximum of 10% do not end in punctuation decreases performance, especially of WtP$_{\text{U}}$. Removing the auxiliary objective (*No aux. punctuation-prediction*) does not allow adaptation via WtP$_{\text{PUNCT}}$, but also decreases performance of WtP$_{\text{U}}$ and WtP$_{\text{T}}$. Corrupting punctuation, but not adding the auxiliary loss in Equation (6) (*No aux. punctuation-prediction + punctuation corruption*) also decreases performance. This implies that, besides enabling adaptation via WtP$_{\text{PUNCT}}$, the auxiliary punctuation-prediction objective positively impacts the main newline-prediction objective.

Reducing the amount of pretraining data has comparatively little impact on performance, possibly because newline- and punctuation-prediction are not particularly knowledge-intensive tasks. Reducing the amount of layers to one decreases performance by a considerable amount. Scaling to 6, 9 and 12 layers continues to improve performance.

[10] nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.2.1

[11] For a fair comparison, we use the 12-layer WtP model.

6 Conclusion

We have introduced WtP, a method for multilingual sentence segmentation without relying on punctuation or sentence-segmented training data. We have demonstrated strong performance of our self-supervised sentence segmentation method across 85 languages and 3 different corpora, matching performance of the best prior (supervised) tools. We have further improved WtP in a data-efficient way by means of inexpensive supervised adaptation, which leads to state-of-the-art scores outperforming the best prior tools by an average 6.1% F1 points. We have also found that, though often an overlooked component of NLP systems, sentence segmentation can have a strong impact on downstream tasks by showing that matching the segmentation at inference to the segmentation used during training benefits sentence-level MT models.

Limitations

WtP performs comparatively worse in some low-resource languages (e.g. Welsh, Nepalese, Punjabi, Pushto). This may be attributed to quality issues of mC4 in these languages (Kreutzer et al., 2022). In addition, we find that the adapted WtP$_{\text{PUNCT}}$ classifiers generally do not transfer well across languages and dataset collections (Appendix D). Finally, although bias is less obvious in segmentation tasks than e.g. generation, WtP may be biased by performing disproportionately well on text by communities which are overrepresented in the training data, while performing worse on text from underrepresented communities. We try to minimize this form of bias by sampling text from all languages uniformly.

Acknowledgements

Ivan Vulić is supported by a personal Royal Society University Research Fellowship *'Inclusive and Sustainable Language Technology for a Truly Multilingual World'* (no 221137; 2022–).

Research supported with Cloud TPUs from Google's TPU Research Cloud (TRC). We thank Christoph Minixhofer for advice in the initial stage of this project. We also thank Sebastian Ruder and Srini Narayanan for helpful feedback on a draft of this paper.

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Tanvirul Alam, Akib Khan, and Firoj Alam. 2020. Punctuation restoration using transformer models for high- and low-resource languages. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 132–142, Online. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Yvette Graham, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, André Martins, Makoto Morishita, Christof Monz, Masaaki Nagata, Toshiaki Nakazawa, and Matteo Negri, editors. 2020. *Proceedings of the Fifth Conference on Machine Translation*. Association for Computational Linguistics, Online.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, André Martins, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Marco Turchi, and Karin Verspoor, editors. 2019. *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*. Association for Computational Linguistics, Florence, Italy.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2013. Report on the 10th iwslt evaluation campaign. In *Proceedings of the 10th International Workshop on Spoken Language Translation: Evaluation Campaign*.

Xiaoyin Che, Cheng Wang, Haojin Yang, and Christoph Meinel. 2016. Punctuation prediction for unsegmented transcript based on word vector. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

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, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

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

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

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.

Matthew Honnibal and Mark Johnson. 2015. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378, Lisbon, Portugal. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Dona M Kagan. 1980. Run-on and fragment sentences: An error analysis. *Research in the Teaching of English*, 14(2):127–138.

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.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Ortiz Suarez, Iroro Orife, Kelechi Ogueji, Andre Niyongabo Rubungo, Toan Q. Nguyen, Mathias Müller, André Müller, Shamsuddeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhalov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa de Silva, Sakine Çabuk Ballı, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. 2022. Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

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.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Eric Malmi, Yue Dong, Jonathan Mallinson, Aleksandr Chuklin, Jakub Adamek, Daniil Mirylenka, Felix Stahlberg, Sebastian Krause, Shankar Kumar, and Aliaksei Severyn. 2022. Text generation with text-editing models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorial Abstracts*, pages 1–7, Seattle, United States. Association for Computational Linguistics.

Robert C Moore. 2021. Indirectly supervised english sentence break prediction using paragraph break probability estimates. *arXiv preprint arXiv:2109.12023*.

Rungsiman Nararatwong, Natthawut Kertkeidkachorn, Nagul Cooharojananone, and Hitoshi Okada. 2018. Improving thai word and sentence segmentation using linguistic knowledge. *IEICE TRANSACTIONS on Information and Systems*, 101(12):3218–3225.

Binh Nguyen, Vu Bao Hung Nguyen, Hien Nguyen, Pham Ngoc Phuong, The-Loc Nguyen, Quoc Truong Do, and Luong Chi Mai. 2019. Fast and accurate capitalization and punctuation for automatic speech recognition using transformer and chunk merging. In *2019 22nd Conference of the Oriental COCOSDA International Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques (O-COCOSDA)*, pages 1–5. IEEE.

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.

Vasile Păiş and Dan Tufiş. 2022. Capitalization and punctuation restoration: a survey. *Artificial Intelligence Review*, 55(3):1681–1722.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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.

Wannaphong Phatthiyaphaibun, Korakot Chaovavanich, Charin Polpanumas, Arthit Suriyawongkul, Lalita Lowphansirikul, and Pattarawat Chormai. 2016. PyThaiNLP: Thai Natural Language Processing in Python.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*.

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.

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*.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. 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 3118–3135, Online. Association for Computational Linguistics.

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.

Chanatip Saetia, Ekapol Chuangsuwanich, Tawunrat Chalothorn, and Peerapon Vateekul. 2019. Semi-supervised thai sentence segmentation using local and distant word representations. *arXiv preprint arXiv:1908.01294*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Virach Sornlertlamvanich, Thatsanee Charoenporn, and Hitoshi Isahara. 1997. Orchid: Thai part-of-speech tagged corpus. *National Electronics and Computer Technology Center Technical Report*, pages 5–19.

Henry Sweet. 2014. *A new English grammar*, volume 1, page 155. Cambridge University Press.

Yi Tay, Vinh Q. Tran, Sebastian Ruder, Jai Gupta, Hyung Won Chung, Dara Bahri, Zhen Qin, Simon Baumgartner, Cong Yu, and Donald Metzler. 2022. Charformer: Fast character transformers via gradient-based subword tokenization. In *International Conference on Learning Representations*.

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.

Nicola Ueffing, Maximilian Bisani, and Paul Vozila. 2013. Improved models for automatic punctuation prediction for spoken and written text. In *Interspeech*, pages 3097–3101.

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.

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*, pages 843–854, Abu Dhabi. 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.

Nina Zhou, AiTi Aw, Nattadaporn Lertcheva, and Xuancong Wang. 2016. A word labeling approach to Thai sentence boundary detection and POS tagging. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 319–327, Osaka, Japan. The COLING 2016 Organizing Committee.

A Results in All Languages

We give an overview over all used languages for pretraining in Table 8. Results of WtP and prior methods for all languages are shown in Tables 9-14.

B Qualitative Analysis of the Impact on Machine Translation

An example paragraph of the German-English OPUS100 data is shown in Table 15. Passing the entire paragraph to the model at once (*None*) misses a considerable portion of the latter parts of the input text. Semantically uninformed chunking (*Naïve*) misses some parts where semantics are split across chunks. Segmentation with Ersatz results in longer sentences since text can not be split on non-punctuation characters, and adds a wrong boundary after 'p.'. Although $\text{WtP}_{\text{PUNCT}}$ does not exactly match the segmentation in OPUS100 (e.g. the first sentence, which could be considered undersegmented in the ground truth, is split into three parts), it identifies all correct sentence boundaries.

C Punctuation Restoration Details

For $\text{WtP}_{\text{PUNCT}}$, we fit a logistic regression on the punctuation logits to predict one of four classes for each character (*comma*, *period*, *question mark*, *none*). For $\text{WtP}_{\text{FINETUNE}}$, we replace the pretrained prediction head g_θ with a new 4-class prediction head and train the entire model with a batch size of 32 for 5 epochs at 256 characters sequence length. We use the AdamW optimizer with a triangular learning rate schedule peaking at 1e-5 at 30% of training steps. For CANINE-S, we again add a 4-class prediction head on top of the pretrained model and use the same hyperparameters as for $\text{WtP}_{\text{FINETUNE}}$. Hyperparameters were chosen by setting them to reasonable defaults (no hyperparameter tuning).

D Transferring WtP across Languages and Collections

We investigate the capacity of WtP models for cross-lingual transfer and transfer across collections by training WtP_{T} and $\text{WtP}_{\text{PUNCT}}$ using supervised data in one corpus, then evaluating performance on a different corpus and comparing against the WtP_{U} baseline. For cross-lingual transfer, we keep the collection the same between source and target; for cross-collection transfer, we keep the language the same. Results for cross-lingual transfer are shown in Figure 4, cross-collection transfer in Figure 5. We find that the threshold estimated by WtP_{T} can generally be transferred. This is consistent with our observation that WtP_{U} performs well in many languages although the threshold for WtP_{U} was selected using only one corpus (English UD). More sophisticated adaptation via $\text{WtP}_{\text{PUNCT}}$ generally hurts cross-lingual transfer, although some directions (e.g. th → ja, es → xh) exhibit strong positive transfer. Across collections, transferring from Ersatz and UD leads to moderate improvements, while transferring from OPUS100 strongly decreases performance.

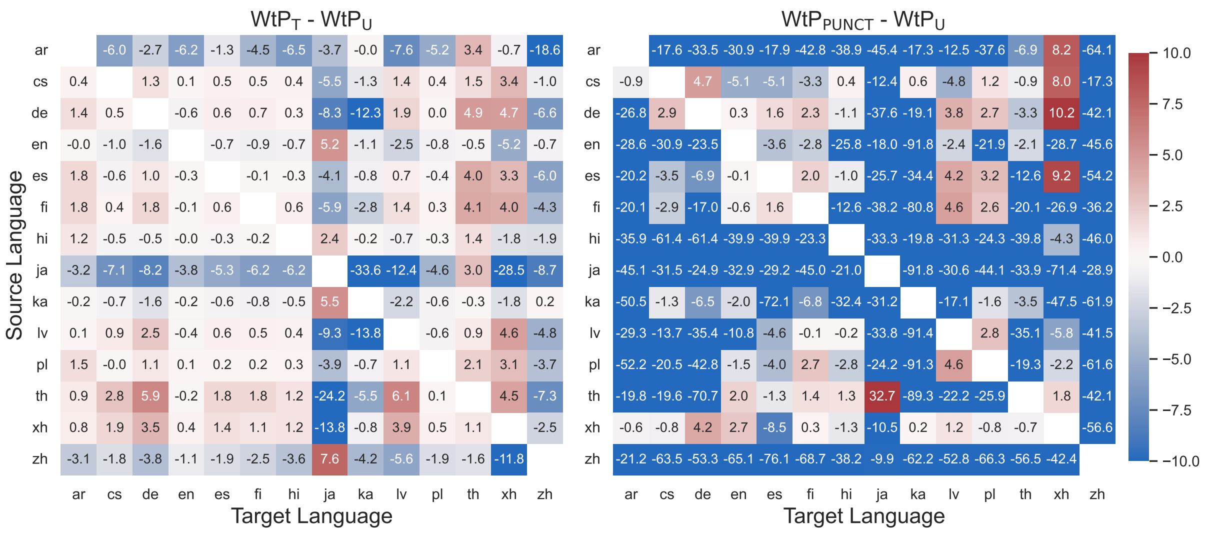

Figure 4: Performance of cross-lingual transfer. Values indicate the average difference of the adaptation method to the self-supervised WtP$_U$ baseline when trained on data in the source language and evaluated on data in the target language. We average scores across corpora, and use the same corpus collection for training and evaluation.

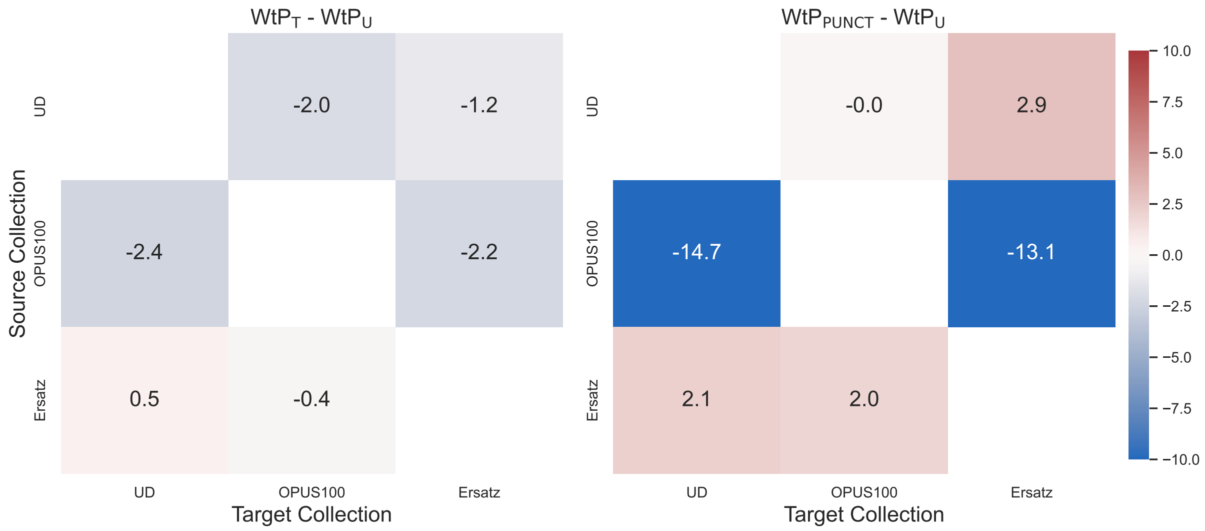

Figure 5: Performance of cross-collection transfer. Values indicate the average difference of the adaptation method to the self-supervised WtP$_U$ baseline when trained on data in the source collection and evaluated on data in the target collection. Scores are averaged across languages.

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

Table 8: List of the 85 languages used in pretraining, whether they generally use whitespace to separate sentences, and their corresponding evaluation dataset sizes in sentences. For UD, the treebank name is also shown.

		af	am	ar	az	be	bg	bn	ca	ceb	cs	cy	da	de	el	en
UD	SpaCy$_{\text{SENT}}$	98.6	-	73.4	-	-	90.7	**100.0**	95.3	-	83.7	-	85.7	89.9	90.5	89.3
	PySBD	-	-	29.5	-	-	74.5	-	-	-	-	-	72.4	79.9	91.6	75.5
	SpaCy$_{\text{DP}}$	-	-	-	-	-	-	-	**99.8**	-	-	-	94.7	**96.9**	94.4	91.7
	Ersatz	-	-	81.0	-	-	-	-	-	-	89.5	-	-	92.4	-	89.4
	Punkt	-	-	-	-	-	-	-	-	-	89.2	-	94.3	92.6	93.1	91.2
	WtP$_{\text{U}}$	98.0	-	82.1	-	89.8	98.2	94.1	98.4	**99.7**	92.5	99.2	95.2	95.7	97.4	95.0
	WtP$_{\text{T}}$	99.1	-	87.5	-	89.6	98.1	-	98.5	-	92.6	98.9	94.6	95.8	**97.7**	95.0
	WtP$_{\text{PUNCT}}$	**99.9**	-	**88.2**	-	**92.1**	**99.6**	-	99.8	-	**95.5**	**99.5**	**98.6**	96.7	**97.7**	**96.7**
OPUS100	SpaCy$_{\text{SENT}}$	30.7	6.6	51.4	70.6	-	91.5	78.6	86.2	-	84.6	-	87.6	70.0	82.5	86.8
	PySBD	-	6.2	39.1	-	-	72.9	-	-	-	-	-	70.3	66.6	62.5	59.8
	SpaCy$_{\text{DP}}$	-	-	-	-	-	-	-	87.5	-	-	-	90.7	74.5	91.1	89.4
	Ersatz	-	-	59.7	-	-	-	-	-	-	86.2	-	-	73.2	-	87.7
	Punkt	-	-	-	-	-	-	-	-	-	86.5	-	90.1	73.5	85.4	88.6
	WtP$_{\text{U}}$	75.8	60.4	66.2	76.6	73.1	93.7	79.5	88.7	-	88.5	69.8	89.2	78.5	91.7	91.3
	WtP$_{\text{T}}$	77.8	65.1	66.4	76.1	74.2	93.3	83.1	89.6	-	90.8	75.6	90.9	85.8	92.6	90.3
	WtP$_{\text{PUNCT}}$	**88.5**	**72.0**	**77.2**	**83.8**	**89.8**	**96.5**	**87.4**	**94.5**	-	**95.2**	**82.6**	**95.0**	**90.1**	**96.2**	**95.0**
Ersatz	SpaCy$_{\text{SENT}}$	-	-	89.4	-	-	-	-	-	-	84.1	-	-	89.9	-	89.8
	PySBD	-	-	47.9	-	-	-	-	-	-	-	-	-	95.5	-	74.2
	SpaCy$_{\text{DP}}$	-	-	-	-	-	-	-	-	-	-	-	-	96.3	-	98.3
	Ersatz	-	-	92.9	-	-	-	-	-	-	96.8	-	-	95.6	-	97.6
	Punkt	-	-	-	-	-	-	-	-	-	96.8	-	-	95.5	-	97.8
	WtP$_{\text{U}}$	-	-	87.8	-	-	-	-	-	-	93.7	-	-	95.7	-	96.8
	WtP$_{\text{T}}$	-	-	88.9	-	-	-	-	-	-	94.1	-	-	96.0	-	96.9
	WtP$_{\text{PUNCT}}$	-	-	**92.9**	-	-	-	-	-	-	**98.9**	-	-	**99.3**	-	**98.7**

Table 9: Sentence segmentation test F1 scores on languages af-en.

		eo	es	et	eu	fa	fi	fr	fy	ga	gd	gl	gu	ha	he	hi
UD	SpaCy$_{\text{SENT}}$	-	89.3	87.1	92.5	99.7	87.3	95.3	-	85.2	-	-	-	-	94.4	95.2
	PySBD	-	46.2	-	-	98.9	-	61.9	-	-	-	-	-	-	-	99.7
	SpaCy$_{\text{DP}}$	-	99.0	-	-	-	95.5	92.0	-	-	-	-	-	-	-	-
	Ersatz	-	97.5	93.1	-	-	92.9	97.3	-	-	-	-	-	-	-	99.5
	Punkt	-	98.6	93.7	-	-	92.8	97.2	-	-	-	-	-	-	-	-
	WtP$_{\text{U}}$	-	96.7	93.0	97.4	97.0	92.8	96.7	-	85.7	71.8	97.8	-	-	95.5	96.5
	WtP$_{\text{T}}$	-	97.1	93.2	97.6	98.0	93.0	97.1	-	91.3	72.0	**98.9**	-	-	96.3	97.1
	WtP$_{\text{PUNCT}}$	-	**99.7**	**98.2**	**99.9**	**99.9**	**98.2**	**98.8**	-	**98.1**	**81.2**	98.6	-	-	**97.2**	**99.9**
OPUS100	SpaCy$_{\text{SENT}}$	-	78.4	84.6	79.4	51.9	91.4	84.6	-	54.2	-	-	3.6	-	91.7	54.0
	PySBD	-	68.0	-	-	46.1	-	81.4	-	-	-	-	-	-	-	23.1
	SpaCy$_{\text{DP}}$	-	88.4	-	-	-	92.9	84.6	-	-	-	-	-	-	-	-
	Ersatz	-	90.0	87.3	-	-	92.9	86.4	-	-	-	-	20.9	-	-	58.5
	Punkt	-	90.2	87.8	-	-	93.5	86.0	-	-	-	-	-	-	-	-
	WtP$_{\text{U}}$	91.6	90.8	84.0	85.7	61.2	91.5	**87.9**	45.1	79.1	84.6	89.4	70.9	84.1	90.8	66.7
	WtP$_{\text{T}}$	91.2	92.1	88.6	87.0	61.2	93.1	-	61.8	78.6	84.9	89.8	71.0	89.5	90.1	66.1
	WtP$_{\text{PUNCT}}$	**95.7**	**95.4**	**94.9**	**92.2**	**73.7**	**96.1**	-	**88.6**	**87.9**	**92.8**	**94.4**	**77.8**	**92.1**	**94.1**	**77.5**
Ersatz	SpaCy$_{\text{SENT}}$	-	85.0	84.1	-	-	94.7	90.8	-	-	-	-	3.7	-	-	89.9
	PySBD	-	84.6	-	-	-	-	96.1	-	-	-	-	-	-	-	87.8
	SpaCy$_{\text{DP}}$	-	96.4	-	-	-	95.2	87.6	-	-	-	-	-	-	-	-
	Ersatz	-	96.7	**98.1**	-	-	95.9	96.3	-	-	-	-	94.4	-	-	**96.9**
	Punkt	-	96.6	97.6	-	-	95.7	96.1	-	-	-	-	-	-	-	-
	WtP$_{\text{U}}$	-	98.8	96.2	-	-	97.5	97.4	-	-	-	-	90.5	-	-	94.4
	WtP$_{\text{T}}$	-	97.8	95.8	-	-	97.3	96.8	-	-	-	-	89.6	-	-	94.7
	WtP$_{\text{PUNCT}}$	-	**99.5**	98.0	-	-	**99.4**	**98.6**	-	-	-	-	**96.9**	-	-	96.4

Table 10: Sentence segmentation test F1 scores on languages eo-hi.

		hu	hy	id	ig	is	it	ja	jv	ka	kk	km	kn	ko	ku	ky
UD	SpaCy$_{\mathrm{SENT}}$	90.2	0.3	94.3	-	94.0	93.2	96.3	-	-	-	-	-	-	-	-
	PySBD	-	92.7	-	-	-	74.6	97.9	-	-	95.6	-	-	-	-	-
	SpaCy$_{\mathrm{DP}}$	-	-	-	-	-	**99.6**	**98.0**	-	-	-	-	-	**99.9**	-	-
	Ersatz	-	-	-	-	-	-	93.4	-	-	95.6	-	-	-	-	-
	Punkt	-	-	-	-	-	95.4	-	-	-	-	-	-	-	-	-
	WtP$_{\mathrm{U}}$	96.1	96.3	**98.2**	-	86.9	94.3	93.9	**97.3**	-	**97.6**	-	-	99.3	-	-
	WtP$_{\mathrm{T}}$	96.4	96.3	-	-	89.7	94.3	95.8	-	-	83.0	-	-	99.4	-	-
	WtP$_{\mathrm{PUNCT}}$	**99.3**	**98.1**	-	-	**96.7**	99.5	98.0	-	-	97.2	-	-	**99.9**	-	-
OPUS100	SpaCy$_{\mathrm{SENT}}$	91.1	1.8	87.8	-	93.6	85.5	43.6	-	-	-	-	9.3	-	-	7.8
	PySBD	-	58.8	-	-	-	74.7	42.9	-	-	35.6	-	-	-	-	-
	SpaCy$_{\mathrm{DP}}$	-	-	-	-	-	85.7	42.2	-	-	-	-	-	47.4	-	-
	Ersatz	-	-	-	-	-	-	28.3	-	-	37.8	0.1	-	-	-	-
	Punkt	-	-	-	-	-	88.0	-	-	-	-	-	-	-	-	-
	WtP$_{\mathrm{U}}$	92.2	**86.3**	89.8	79.1	94.4	85.9	44.9	-	91.9	74.4	72.9	66.0	57.5	79.6	85.3
	WtP$_{\mathrm{T}}$	92.7	-	90.4	82.9	94.8	89.3	80.5	-	91.7	76.0	72.0	61.3	71.8	67.0	85.2
	WtP$_{\mathrm{PUNCT}}$	**96.5**	-	**94.5**	**90.7**	**96.9**	**94.0**	**87.4**	-	**93.2**	**92.5**	**79.3**	**78.5**	**82.6**	**84.8**	**90.9**
Ersatz	SpaCy$_{\mathrm{SENT}}$	-	-	-	-	-	-	84.7	-	-	-	-	-	-	-	-
	PySBD	-	-	-	-	-	-	87.7	-	-	64.7	-	-	-	-	-
	SpaCy$_{\mathrm{DP}}$	-	-	-	-	-	-	91.2	-	-	-	-	-	-	-	-
	Ersatz	-	-	-	-	-	-	85.9	-	-	99.6	31.7	-	-	-	-
	Punkt	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	WtP$_{\mathrm{U}}$	-	-	-	-	-	-	81.5	-	-	96.4	72.1	-	-	-	-
	WtP$_{\mathrm{T}}$	-	-	-	-	-	-	82.6	-	-	95.8	91.5	-	-	-	-
	WtP$_{\mathrm{PUNCT}}$	-	-	-	-	-	-	**94.8**	-	-	**99.8**	**92.0**	-	-	-	-

Table 11: Sentence segmentation test F1 scores on languages hu-ky.

		la	lt	lv	mg	mk	ml	mn	mr	ms	mt	my	ne	nl	no	pa
UD	SpaCy$_{\mathrm{SENT}}$	89.2	88.0	92.4	-	-	-	-	68.0	-	-	-	-	90.7	93.5	-
	PySBD	-	-	-	-	-	-	-	60.0	-	-	-	-	93.6	-	-
	SpaCy$_{\mathrm{DP}}$	-	92.5	-	-	-	-	-	-	-	-	-	-	95.1	-	-
	Ersatz	-	92.6	96.8	-	-	-	-	-	-	-	-	-	-	-	-
	Punkt	-	-	-	-	-	-	-	-	-	-	-	-	95.6	95.5	-
	WtP$_{\mathrm{U}}$	89.5	98.3	96.5	-	-	-	-	90.5	-	90.6	-	-	94.4	98.2	-
	WtP$_{\mathrm{T}}$	90.5	98.1	96.4	-	-	-	-	92.8	-	88.0	-	-	93.6	98.5	-
	WtP$_{\mathrm{PUNCT}}$	**97.3**	**99.5**	**99.1**	-	-	-	-	**97.9**	-	**94.4**	-	-	**97.2**	**99.5**	-
OPUS100	SpaCy$_{\mathrm{SENT}}$	-	67.2	58.0	-	90.4	39.9	-	84.4	-	-	-	15.3	92.2	92.1	-
	PySBD	-	-	-	-	-	-	-	86.1	-	-	27.2	-	18.2	-	-
	SpaCy$_{\mathrm{DP}}$	-	77.8	-	-	81.9	-	-	-	-	-	-	-	93.0	-	-
	Ersatz	-	77.3	77.6	-	-	-	-	-	-	-	-	-	-	-	-
	Punkt	-	-	-	-	-	-	-	-	-	-	-	-	**93.9**	94.8	-
	WtP$_{\mathrm{U}}$	-	78.3	79.6	90.0	93.0	81.3	**81.0**	89.1	87.7	63.0	70.5	70.6	92.2	94.7	56.3
	WtP$_{\mathrm{T}}$	-	85.3	86.5	92.1	93.0	82.4	-	89.0	88.5	81.1	75.5	70.2	-	94.8	63.3
	WtP$_{\mathrm{PUNCT}}$	-	**90.7**	**91.9**	**95.5**	**96.0**	**87.3**	-	**93.7**	**94.2**	**89.0**	**82.8**	**76.1**	-	**96.4**	**78.4**
Ersatz	SpaCy$_{\mathrm{SENT}}$	-	74.3	89.8	-	-	-	-	-	-	-	-	-	-	-	-
	PySBD	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	SpaCy$_{\mathrm{DP}}$	-	77.6	-	-	-	-	-	-	-	-	-	-	-	-	-
	Ersatz	-	95.1	98.7	-	-	-	-	-	-	-	-	-	-	-	-
	Punkt	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	WtP$_{\mathrm{U}}$	-	96.9	97.2	-	-	-	-	-	-	-	-	-	-	-	-
	WtP$_{\mathrm{T}}$	-	96.6	97.3	-	-	-	-	-	-	-	-	-	-	-	-
	WtP$_{\mathrm{PUNCT}}$	-	**99.2**	**99.4**	-	-	-	-	-	-	-	-	-	-	-	-

Table 12: Sentence segmentation test F1 scores on languages la-pa.

		pl	ps	pt	ro	ru	si	sk	sl	sq	sr	sv	ta	te	tg	th
UD	SpaCy$_{\text{SENT}}$	93.0	-	86.2	98.6	76.6	-	85.3	92.7	**100.0**	74.4	90.0	92.3	-	-	-
	PySBD	85.0	-	-	-	67.7	-	86.6	-	-	-	-	-	-	-	-
	SpaCy$_{\text{DP}}$	98.5	-	**98.4**	94.1	80.3	-	-	-	-	-	88.0	-	-	-	-
	Ersatz	97.5	-	-	98.3	78.3	-	-	-	-	-	-	91.0	-	-	-
	Punkt	97.4	-	92.0	-	78.2	-	-	-	-	-	94.2	-	-	-	-
	WtP$_{\text{U}}$	94.9	-	96.1	82.5	86.1	-	96.2	96.0	**100.0**	97.9	95.1	97.2	-	-	**69.5**
	WtP$_{\text{T}}$	95.8	-	95.7	94.0	87.7	-	96.0	96.5	-	98.2	95.3	97.9	-	-	-
	WtP$_{\text{PUNCT}}$	**99.4**	-	98.3	**99.4**	**93.4**	-	**98.1**	**99.1**	-	**99.8**	**96.9**	**98.8**	-	-	-
OPUS100	SpaCy$_{\text{SENT}}$	89.4	-	90.1	90.7	72.4	75.8	88.1	89.7	87.6	91.6	90.8	40.8	62.5	-	-
	PySBD	17.6	-	-	-	65.9	-	29.5	-	-	-	-	-	-	-	-
	SpaCy$_{\text{DP}}$	92.9	-	90.4	92.2	74.2	-	-	-	-	-	91.5	-	-	-	-
	Ersatz	92.2	1.6	-	92.8	68.7	-	-	-	-	-	-	45.2	-	-	-
	Punkt	92.8	-	92.4	-	75.9	-	-	-	-	-	92.9	-	-	-	-
	WtP$_{\text{U}}$	92.4	64.2	90.7	89.5	**82.1**	80.4	90.7	92.0	89.7	94.3	91.3	66.1	78.7	81.5	68.8
	WtP$_{\text{T}}$	92.8	71.6	92.1	90.0	-	80.8	93.1	93.3	90.8	94.8	93.1	66.5	78.7	83.8	71.5
	WtP$_{\text{PUNCT}}$	**96.0**	**76.7**	**95.8**	**96.9**	-	**86.0**	**96.2**	**95.4**	**95.8**	**96.7**	**96.2**	**75.1**	**84.5**	**91.9**	**72.9**
Ersatz	SpaCy$_{\text{SENT}}$	77.6	-	-	89.6	88.3	-	-	-	-	-	-	88.9	-	-	-
	PySBD	46.1	-	-	-	55.4	-	-	-	-	-	-	-	-	-	-
	SpaCy$_{\text{DP}}$	94.4	-	-	94.4	93.7	-	-	-	-	-	-	-	-	-	-
	Ersatz	95.1	93.7	-	95.9	94.3	-	-	-	-	-	-	95.6	-	-	-
	Punkt	94.3	-	-	-	93.7	-	-	-	-	-	-	-	-	-	-
	WtP$_{\text{U}}$	94.8	85.0	-	97.8	97.6	-	-	-	-	-	-	94.6	-	-	-
	WtP$_{\text{T}}$	92.8	91.6	-	97.1	97.7	-	-	-	-	-	-	95.0	-	-	-
	WtP$_{\text{PUNCT}}$	**98.0**	**96.0**	-	**99.4**	**99.4**	-	-	-	-	-	-	**98.2**	-	-	-

Table 13: Sentence segmentation test F1 scores on languages pl-th.

		tr	uk	ur	uz	vi	xh	yi	yo	zh	zu
UD	SpaCy$_{\text{SENT}}$	94.9	90.7	99.1	-	-	-	-	77.5	96.7	-
	PySBD	-	-	99.1	-	-	-	-	-	98.9	-
	SpaCy$_{\text{DP}}$	-	97.0	-	-	-	-	-	-	98.2	-
	Ersatz	97.6	-	-	-	-	-	-	-	88.8	-
	Punkt	96.3	-	-	-	-	-	-	-	-	-
	WtP$_{\text{U}}$	94.3	92.6	92.4	-	67.8	-	-	**84.5**	98.1	-
	WtP$_{\text{T}}$	94.3	93.1	96.1	-	91.4	-	-	-	98.2	-
	WtP$_{\text{PUNCT}}$	**99.1**	**98.4**	**99.4**	-	**97.9**	-	-	-	**99.8**	-
OPUS100	SpaCy$_{\text{SENT}}$	91.6	86.5	40.0	-	-	-	-	14.1	64.1	-
	PySBD	-	-	30.8	-	-	-	-	-	69.8	-
	SpaCy$_{\text{DP}}$	-	89.7	-	-	-	-	-	-	69.4	-
	Ersatz	92.7	-	-	-	-	-	-	-	54.7	-
	Punkt	93.6	-	-	-	-	-	-	-	-	-
	WtP$_{\text{U}}$	93.4	89.1	53.8	78.2	90.7	78.7	74.7	**76.9**	81.0	73.2
	WtP$_{\text{T}}$	93.6	89.8	53.0	80.4	90.9	81.9	75.7	-	77.8	83.9
	WtP$_{\text{PUNCT}}$	**95.7**	**94.7**	**68.1**	**85.9**	**94.9**	**90.4**	**81.8**	-	**89.2**	**90.9**
Ersatz	SpaCy$_{\text{SENT}}$	85.5	-	-	-	-	-	-	-	90.6	-
	PySBD	-	-	-	-	-	-	-	-	92.7	-
	SpaCy$_{\text{DP}}$	-	-	-	-	-	-	-	-	95.8	-
	Ersatz	96.3	-	-	-	-	-	-	-	87.6	-
	Punkt	92.9	-	-	-	-	-	-	-	-	-
	WtP$_{\text{U}}$	93.4	-	-	-	-	-	-	-	93.5	-
	WtP$_{\text{T}}$	93.4	-	-	-	-	-	-	-	93.7	-
	WtP$_{\text{PUNCT}}$	**98.4**	-	-	-	-	-	-	-	**97.8**	-

Table 14: Sentence segmentation test F1 scores on languages tr-zu.

Source Paragraph		Higgins bat mich um einen Gefallen. Und ich fragte jemand anderen um einen Gefallen. Sie gravierten meinen Namen rein. | Günstige und Luxus Hotels in Schillig: | Stand: 07.11.201513:00:28 | Alle Wettbewerbsrunden sind öffentlich. | Nicht jeder bekommt eine zweite Chance, Bruder. | 01/06/2011 :Auto Moto Rally: Runde Nsele 04 und 05 Juni 2011. | Also, komm rein. Aber du musst ruhig sein, ja? | 32008 L 0057: Richtlinie 2008/57/EG des Europäischen Parlaments und des Rates vom 17. Juni 2008 über die Interoperabilität des Eisenbahnsystems in der Gemeinschaft (ABl. L 191 vom 18.7.2008, S. 1).“ | Gebiet: Tschechische Republik | Was war daran auszusetzen?
Target Paragraph		Higgins asked me a favor, I asked someone else a favor, they slapped my name on it. | Accommodations in Schillig | Stand: 07.11.201513:02:59 | All rounds of the competition are open to the public. | Not everybody gets a second chance to do what’s right, bro. | 01/06/2011 :Auto Moto Rally: Round of Nsele 04 and 05 June 2011. | So, come on in, but keep it quiet, okay? | Directive 2008/57/EC of the European Parliament and of the Council of 17 June 2008 on the interoperability of the rail system within the Community (OJ L 191, 18.7.2008, p. 1).’; | Area: Czech Republic | What was wrong with it?
None	Segmentation	Higgins bat mich um einen Gefallen. Und ich fragte jemand anderen um einen Gefallen. Sie gravierten meinen Namen rein. Günstige und Luxus Hotels in Schillig: Stand: 07.11.201513:00:28 Alle Wettbewerbsrunden sind öffentlich. Nicht jeder bekommt eine zweite Chance, Bruder. 01/06/2011 :Auto Moto Rally: Runde Nsele 04 und 05 Juni 2011. Also, komm rein. Aber du musst ruhig sein, ja? 32008 L 0057: Richtlinie 2008/57/EG des Europäischen Parlaments und des Rates vom 17. Juni 2008 über die Interoperabilität des Eisenbahnsystems in der Gemeinschaft (ABl. L 191 vom 18.7.2008, S. 1).“ Gebiet: Tschechische Republik Was war daran auszusetzen?
	Prediction (BLEU=47.1)	Higgins asked me for a favor. And I asked someone else for a favor. They engraved my name in. Cheap and luxury hotels in Schillig: As of: 07.11.201513:00:28 All competition rounds are public. Not everyone gets a second chance, brother. 01/06/2011 :Auto Moto Rally: Round Nsele 04 and 05 June 2011. So, come in. But you have to be quiet, yes? 32008 L 0057: Directive 2008/57/EC of the European Parliament and of the Council of 17 June 2008 on the interoperability of the railway system in the Community (OJ L 191 of 18.7.2008, p. 1).
Naïve	Segmentation	Higgins bat mich um einen Gefallen. Und ich fragte | jemand anderen um einen Gefallen. Sie gravierten meinen Namen rein. | Günstige und Luxus Hotels in Schillig: Stand: 07.11.201513:00:28 Alle | Wettbewerbsrunden sind öffentlich. Nicht jeder bekommt eine zweite Chance, Bruder. | 01/06/2011 :Auto Moto Rally: Runde Nsele 04 und 05 | Juni 2011. Also, komm rein. Aber du musst ruhig sein, | ja? 32008 L 0057: Richtlinie 2008/57/EG des Europäischen Parlaments | und des Rates vom 17. Juni 2008 über die Interoperabilität | des Eisenbahnsystems in der Gemeinschaft (ABl. L 191 vom | 18.7.2008, S. 1).“ Gebiet: Tschechische Republik Was war daran auszusetzen?
	Prediction (BLEU=46.6)	Higgins asked me for a favor, and I asked | Someone else for a favor, they engraved my name in. | Cheap and Luxury Hotels in Schillig: As of: 07.11.201513:00:28 All | Competition rounds are public. Not everyone gets a second chance, brother. | 01/06/2011 :Auto Moto Rally: Round Nsele 04 and 05 | So, come in, but you have to be quiet, | yes? 32008 L 0057: Directive 2008/57/EC of the European Parliament | and the Council of 17 June 2008 on interoperability | of the rail system in the Community (OJ L 191, | 18.7.2008, p. 1).- Area: Czech Republic What had to be done about it?
Ersatz	Segmentation	Higgins bat mich um einen Gefallen. | Und ich fragte jemand anderen um einen Gefallen. | Sie gravierten meinen Namen rein. | Günstige und Luxus Hotels in Schillig: Stand: 07.11.201513:00:28 Alle Wettbewerbsrunden sind öffentlich. | Nicht jeder bekommt eine zweite Chance, Bruder. | 01/06/2011 :Auto Moto Rally: Runde Nsele 04 und 05 Juni 2011. | Also, komm rein. | Aber du musst ruhig sein, ja? | 32008 L 0057: Richtlinie 2008/57/EG des Europäischen Parlaments und des Rates vom 17. Juni 2008 über die Interoperabilität des Eisenbahnsystems in der Gemeinschaft (ABl. | L 191 vom 18.7.2008, S. 1).“ Gebiet: Tschechische Republik Was war daran auszusetzen?
	Prediction (BLEU=52.1)	Higgins asked me for a favor. | And I asked someone else for a favor. | They engraved my name in. | Cheap and Luxury Hotels in Schillig: As of: 07.11.201513:00:28 All competition rounds are public. | Not everyone gets a second chance, brother. | 01/06/2011 :Auto Moto Rally: Round Nsele 04 and 05 June 2011. | So, come on in. | But you have to be quiet, right? | 32008 L 0057: Directive 2008/57/EC of the European Parliament and of the Council of 17 June 2008 on the interoperability of the rail system in the Community (OJ L 347, 20.12.2008, p. | OJ L 191, 18.7.2008, p. 1).
WtP$_{\text{PUNCT}}$	Segmentation	Higgins bat mich um einen Gefallen. | Und ich fragte jemand anderen um einen Gefallen. | Sie gravierten meinen Namen rein. | Günstige und Luxus Hotels in Schillig: | Stand: 07.11.201513:00:28 | Alle Wettbewerbsrunden sind öffentlich. | Nicht jeder bekommt eine zweite Chance, Bruder. | 01/06/2011 :Auto Moto Rally: Runde Nsele 04 und 05 Juni 2011. | Also, komm rein. | Aber du musst ruhig sein, ja? | 32008 L 0057: Richtlinie 2008/57/EG des Europäischen Parlaments und des Rates vom 17. Juni 2008 über die Interoperabilität des Eisenbahnsystems in der Gemeinschaft (ABl. L 191 vom 18.7.2008, S. 1).“ | Gebiet: Tschechische Republik | Was war daran auszusetzen?
	Prediction (BLEU=59.8)	Higgins asked me for a favor. | And I asked someone else for a favor. | They engraved my name in. | Cheap and Luxury Hotels in Schillig: | Situation as at: 07.11.201513:00:28 | All competitions are open to the public. | Not everyone gets a second chance, brother. | 01/06/2011 :Auto Moto Rally: Round Nsele 04 and 05 June 2011. | So, come on in. | But you have to be quiet, right? | Directive 2008/57/EC of the European Parliament and of the Council of 17 June 2008 on the interoperability of the rail system in the Community (OJ L 191, 18.7.2008, p. 1). | Area: Czech Republic | What was wrong with that?

Table 15: Example paragraph of the German-English OPUS100 data with segmentations and translations following different strategies. Data in OPUS100 is shuffled, so the paragraph is not coherent; this has been shown by Wicks and Post (2021) to have little impact on segmentation performance. The pipe (’|’) indicates sentence boundaries.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 'Limitations'

☑ A2. Did you discuss any potential risks of your work?
Section 'Limitations'

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4, Section 5

☑ B1. Did you cite the creators of artifacts you used?
Section 4, Section 5

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
The models we release are licensed under the Apache 2.0 license, which is visible at the URL where they are distributed (anonymized in the Reviewer's version of the paper).

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Was not specified.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
We do not create any datasets. Being constructed from web crawls, personal and sensitive information may be present in the pretraining dataset (mC4). However, since our model is only trained to segment text, personal or sensitive information can not be produced by the model.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 4, Section 5, Appendix A

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 4, Section 5, Appendix A

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C ☑ Did you run computational experiments?

Section 4, Section 5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4, Appendix C

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 5

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 5

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.