MAGNET 🧲: Improving the Multilingual Fairness of Language Models with Adaptive Gradient-Based Tokenization

Orevaoghene Ahia[1] **Sachin Kumar**[2,3] **Hila Gonen**[1] **Valentin Hofmann**[2]
Tomasz Limisiewicz[4] **Yulia Tsvetkov**[1] **Noah A. Smith**[1,2]
[1]University of Washington [2]Allen Institute for AI [3]The Ohio State University
[4]Charles University
oahia@cs.washington.edu

Abstract

In multilingual settings, non-Latin scripts and low-resource languages are usually disadvantaged in terms of language models' utility, efficiency, and cost. Specifically, previous studies have reported multiple modeling biases that the current tokenization algorithms introduce to non-Latin script languages, the main one being over-segmentation. In this work, we propose MAGNET—multilingual adaptive gradient-based tokenization—to reduce over-segmentation via adaptive gradient-based subword tokenization. MAGNET learns to predict segment boundaries between byte tokens in a sequence via sub-modules within the model, which act as internal boundary predictors (tokenizers). Previous gradient-based tokenization methods aimed for uniform compression across sequences by integrating a single boundary predictor during training and optimizing it end-to-end through stochastic reparameterization alongside the next token prediction objective. However, this approach still results in over-segmentation for non-Latin script languages in multilingual settings. In contrast, MAGNET offers a customizable architecture where byte-level sequences are routed through language-script-specific predictors, each optimized for its respective language script. This modularity enforces equitable segmentation granularity across different language scripts compared to previous methods. Through extensive experiments, we demonstrate that in addition to reducing segmentation disparities, MAGNET also enables faster language modelling and improves downstream utility.

1 Introduction

Despite the proliferation of generative language models (LMs) in English, their non-English counterparts are far from being widely adopted. While multilingual LMs offer several advantages such as resource efficiency and cross-lingual generalization, the performance disparities across languages remain a significant challenge. Previous work has largely attributed these disparities to training data imbalances across languages [43, 34, 29, 24]. Recent work, however, highlights that *tokenization*— the way input text is segmented—can considerably degrade not only model performance but also training and inference costs on account of overly fragmenting certain languages and scripts [3, 33]. Subword segmentation algorithms used to build LM tokenizers [28, 39, 22, 40] typically segment the training corpus relying on frequency statistics alone. Due to data imbalances, they obtain high compression in high-resource languages, while majority of languages are over-fragmented. This issue disproportionately affects non-Latin scripts covering languages spoken by billions of people, which are not only less frequent in such corpora, but can require up to $4\times$ more bytes to represent the same information.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

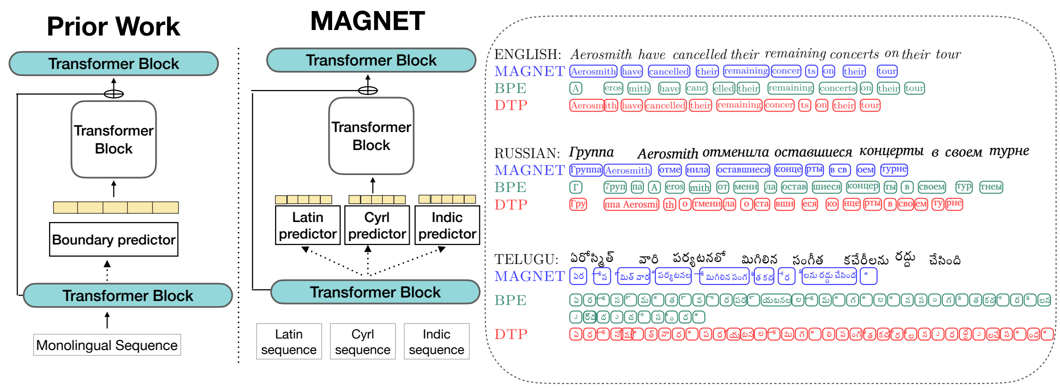


Figure 1: MAGNET routes byte-level sequences via language-script specific boundary predictors. These predictors infer boundaries leading to equitable segmentation across languages. Prior work infers boundaries with a single predictor across languages and leads to over-segmentation.

To address these challenges, prior work has instead proposed to build tokenizer-free models by directly training on character or byte sequences [45, 4]. Operating on smaller or finer-grained segments leads to significantly higher compute- and memory-intense modeling, caused by much longer sequences. To alleviate this issue, recent work introduced a tokenization "layer" within the model architecture by pooling fixed or dynamic length contiguous character representations into smaller sets of patch representations [30, 9, 41, 31, 14, 15], resulting in models optimized with a sequence "compression rate" in mind. While these can improve efficiency, they are mostly suited for character-level modeling for scripts whose characters can be mapped to single Unicode codepoint. Due to the extensive number of codepoints in Unicode, character-based vocabularies can be extremely large for multilingual models. Since many of those characters may never appear during training, "out-of-vocabulary" issues similar to those experienced with subword models may arise if we only included codepoints in the training data [28].[1] When extending these methods to training byte-level multilingual models, we observe that the disparities in fragmentation rates across languages persist . For instance, the English text "Fellow wrestlers also paid tribute to Luna." and its Telugu equivalent, "తోటి మల్ల యుద్ధకారులు కూడా లూనాకు నివాళులు అర్పించారు.", contain 43 and 148 UTF-8 bytes, respectively. A fixed compression ratio for both languages will result in the Telugu text getting over fragmented with requiring close to 3$\times$ more tokens than English.

In this work, we propose MAGNET (multilingual adaptive gradient-based tokenization) to reduce this disparity in tokenizer-free multilingual LMs. Our goal is to obtain end-to-end multilingual language modeling with gradient-based subword tokenization that results in high and similar sequence compression across languages with varying scripts. We leverage hourglass transformers [30, 31] to efficiently route byte-level sequences through language-script specific internal boundary predictors trained to infer word boundaries between byte tokens in a sequence. These boundary predictors are trained end-to-end through stochastic reparameterisation [20, 27]. The inferred boundaries are then used to pool representations of contiguous tokens in the same segment, after which the pooled (narrow) representation is passed into the rest of the transformer block. Unlike previous gradient-based tokenization approaches that apply the same compression rate to all languages in the pretraining data by incorporating a single boundary predictor within their model architectures, MAGNET employs modularity. It incorporates multiple language-script specific predictors to achieve equitable segmentation granularity across different language scripts. We test the effectiveness of MAGNET on equitable fragmentation, model efficiency, and downstream task performance across nine typologically different languages, comparing to byte-level models without compression and existing gradient-based tokenization models [31]. Our extensive experiments demonstrate that our approach MAGNET results in more equitable tokenization when compared to subword-tokenizers, byte-level tokenizers and previous gradient-based tokenization models. This in turn leads to faster modelling and competitive performance across downstream-tasks.[2]

[1] Another disparity with character-level tokenizers is that Chinese-Japanese-Korean scripts use a high number of Unicode codepoints.

[2] Code and data are publicly available at https://github.com/orevaahia/magnet-tokenization

2 MAGNET: Multilingual Adaptive GradieNt-basEd Tokenization

Our goal is to build a multilingual byte-level language model with equitable segmentation across languages. We propose to achieve this by dedicating a separate module within the model for each writing script, to serve as an internal tokenizer for languages that use that script. Our proposed model, called MAGNET, builds on *hourglass transformers* [30, 31], an architecture that was introduced to efficiently handle long sequences in tokenizer-free models. We make several simple but important modifications to this architecture in order to obtain equitable segmentation across languages, while maintaining a high quality of multilingual modeling. In what follows, we explain the main concepts of hourglass transformers, and then introduce the modifications we make to accommodate equitable multilingual modeling.

2.1 Background: Hourglass Transformers

The hourglass transformer [31, 30] is a hierarchical architecture for efficiently handling long sequences. The architecture has three main components, each consisting of one or more transformer layers: A **tokenization submodule** which takes as input a byte sequence and outputs a segmentation, a **language modeling submodule** that takes as input the predicted segments or tokens and is then trained to perform next token prediction, and an **upsampling module** that takes as input the hidden representations of the segmentations and converts them back to a byte sequence on which a typical language modeling loss can be applied. Considering this model as a blackbox, it still performs byte-level language modeling, however, it requires significantly less compute thanks to the tokenization submodule.

Gradient-based Tokenization This submodule performs two steps. First, the given input sequence $x_1, \ldots, x_N$ (where each x_t is a byte in our case) is encoded using a small transformer network (with causal attention) to produce a sequence of hidden vectors $h_1^T, \ldots, h_N^T$. Next, a *boundary predictor* takes as input each h_t and predicts a scalar value between 0 and 1, indicating the probability of position t to be the end of a segment. It is implemented as

$$\hat{b}_t = p(b_t = 1) = \sigma(\text{MLP}_\phi(h_t)), \tag{1}$$

where MLP indicates a multi-layer perceptron and σ is the sigmoid function. To convert the soft probabilities to hard segment predictions, a Bernoulli distribution is sampled from, defined by $\hat{b}_t$. Since the sampling operation will make the process non-differentiable, hard Gumbel-sigmoid is used, a stochastic reparameterization of the Bernoulli distribution, following Nawrot et al. [31]:

$$b_t = \text{sigmoid}\left[\log \frac{\hat{b}_t u}{(1-\hat{b}_t)(1-u)}^{\frac{1}{\tau}}\right], \quad u \sim \text{Uniform}(0,1) \tag{2}$$

where τ is a hyper-parameter. Since this module is differentiable, the segmentations are learned during training of the full model.[3] This module is referred to as "gradient-based tokenization."

Language Modeling Given a sequence of segment boundaries $b_t \in \{0, 1\}$ from the boundary predictor, this submodule first pools the hidden states belonging to the same segment by averaging them to form a sequence of representations $h_1^P, \ldots, h_k^P$.[4] Let $t_1, \ldots, t_k$ indicate the positions at which a boundary is sampled, i.e., for any contiguous pair t_j, t_{j+1}, the sequence $x_{t_j+1} \ldots x_{t_{j+1}}$ forms a "token" ending at position t_{j+1}.[5] The input representation of this "token" is defined as $hP_j = \frac{1}{t_{j+1}-t_j} \sum_{t=t_j+1}^{t_{j+1}} hT_t$. These representations are then passed through the middle block of transformer layers (with causal attention) to obtain another sequence of hidden representations $h_1^M, \ldots, h_k^M$. From the perspective of a subword-tokenizaton based language model, this module is equivalent to the transformer blocks without the input and output embedding layers.

Upsampling This module converts h_l^M to probabilities over a byte vocabulary. This involves, first, upsampling the output of the middle block to the original resolution by duplication followed by skip

[3]Nawrot et al. [31] also explore learning the segmentations using supervision from predefined word or subword boundaries. However, it is not a viable solution for all languages and does not resolve the unfairness issues.

[4]P, M, and T denote representations in the middle transformer block, after pooling and at the token level.

[5]The first token is defined as $x_0 \ldots x_{t_1}$ and the last token as $x_{t_k+1} \ldots x_N$.

connection: $h_t^U = h^M_{\lceil \frac{t-k+1}{k} \rceil} + h_t^T$.[6] These vectors are further passed through a small transformer network followed by an unembedding layer and a softmax to get a probability distribution over which language modeling loss (cross entropy) can be computed. To prevent the boundary predictor from collapsing and trivially predicting each position t as a boundary, Nawrot et al. [31] propose adding a regularizer to the LM objective: $-\log \text{Binomial}(\beta; l, k)$ where,

$$\text{Binomial}(\beta; N, k) = \binom{N}{k} \beta^k (1-\beta)^{N-k}, \quad \text{and} \quad k = \sum_N b_t. \tag{3}$$

Here $\beta \in [0, 1]$ is a hyperparameter, k defines the number of predicted segments or tokens. Intuitively, this loss nudges the model to find a k close to βl which is the mode of the Binomial distribution. In other words, β allows to control the compression rate of the input sequence to approximately $\frac{1}{\beta}\times$. Setting it as 1 will lead to every position being predicted as boundary whereas setting it to 0 will cause no boundaries to be predicted.

2.2 Adaptive Gradient-Based Tokenization via Multiple Boundary Predictors

To encode the same information, different languages require different number of bytes, owing to their different inherent efficiencies [3, 25, 33] as well as restrictions imposed by Unicode mappings, where non-Latin languages (e.g., Indian languages) may require up to 4 bytes per character.[7][8] In multilingual models, setting the same compression rate (via β) for all languages will lead to text in some languages getting segmented into much longer sequences,[9] see Equation (3). This disparity contributes to higher compute and memory costs for such languages as well as poorer model performance for downstream tasks [3]. This issue parallels subword tokenizers where languages with higher segmentation rates get disadvantaged due to longer context-length requirements to perform the same task and poorer in-context learning performance since the same context length fits fewer training instances than other languages leading to unfairness [5, 3, 25, 33].

To make tokenization more equitable, we propose MAGNET, which efficiently learns to segment sequences across languages and language scripts with similar granularity. As part of creating equitable segmentation, we aim to efficiently maximize sequence compression, without having a negative impact on downstream performance across languages.

Introducing multiple gradient-based tokenizers To achieve this, we propose a modification to the model architecture that enables the processing of multiple language scripts. Each script has its own boundary predictor trained with distinct Binomial priors β determined based on the scripts' Unicode encoding and also tailored to a desired compression rate. This allows us to achieve similar fragmentation rates across languages, due to variations in compression. The input sequence is tagged with its script[10] and we infer the segmentation by routing it through the appropriate boundary predictor. The remainder of the model architecture remains the same.

Determining β for equitable tokenizaton We use the binomial priors β for each boundary predictor to control the rates of the resulting segmentations for the different scripts. Since we want to impose equitable lengths across languages, we set the different β according to the following process. First, we choose an *anchor language* L for each script in our training corpus and define a quantity *byte-to-word ratio* $\bar{R}$ for this script as follows. Let $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_D\}$ be a sample of text sequences in language L from our training corpus with $|\mathbf{x}_i|$ denoting the byte-length and $\text{count}_{\text{words}}(\mathbf{x}_i)$ the number of words[11] in sequence $\mathbf{x}_i$. We define the average byte-to-word ratio $\bar{R}$ over $\mathcal{X}$ as:

$$\bar{R} = \tfrac{1}{D} \textstyle\sum_{i=1}^{D} \frac{|\mathbf{x}_i|}{\text{count}_{\text{words}}(\mathbf{x}_i)} \tag{4}$$

[6]The hidden vectors are shifted by one in order to perform next token prediction.

[7]https://en.wikipedia.org/wiki/UTF-8

[8]Modeling characters directly may alleviate this issue; however, character-level multilingual models can explode vocabulary sizes.

[9]As is the case generally in multilingual modeling, also without boundary predictors.

[10]Determining the script of given text sequence is trivial, we assume every sequence contains a single script.

[11]For the purposes of this work, words are defined by whitespace boundaries.

We then set the prior β_S for the corresponding script S to be $1/\bar{R}$. Our final training objective over a single instance **x** is as follows:

$$\sum_{i=1}^{N} -\log p_\theta(x_i|x_{<i}) - \lambda \sum_{S} \mathbb{I}(\text{script}(\mathbf{x}) = S) \log \text{Binomial}(\beta_S; N, k)$$

where $\mathbb{I}$ is the indicator function and script$(\cdot)$ is a function assigning a writing script to a sequence of bytes **x**, such assignment can be easily obtained based on codepoint definitions in Unicode.

3 Experimental Setup

3.1 Language Modeling

Pretraining Data We pretrain all models on nine languages (English, Spanish, French, Russian, Ukranian, Belarusian, Telugu, Bengali and Hindi) divided into three groups, written with distinct scripts: Latin, Cyrillic, and Indic (Brahmic) scripts. Our choice of selection is based on the linguistic diversity of these languages and the availability of data for downstream evaluation. Our pretraining data is obtained from the OSCAR dataset [32]. We present the statistics for each language in Appendix C.

Baselines We compare MAGNET against Dynamic Token Pooling [31], which infers boundaries with a fixed binomial prior β for every sequence, irrespective of the language script. This model is referred to as DTP in the rest of the paper. DTP has a single boundary predictor; we train two versions of this baseline with the binomial prior β as 0.2 and 0.1 respectively yielding $5\times$ and $10\times$ compression respectively. We also compare against a byte-level decoder language model. To ensure fair model comparisons, this model has a similar architecture as DTP, but without any sequence compression.

MAGNET configurations We compute the byte-word-ratios', choosing English, Russian and Telugu as anchor languages for each the language script, based on initial explorations. The FLORES [16] dataset is used for this purpose and the resulting ratios are approximately $5\times$, $10\times$, and $20\times$ for English, Russian, and Telugu, respectively. Based on these ratios, we train five MAGNET models with different binomial prior combinations maintaining the ratio but adjusting the multipliers. First, to optimize for word-level boundary segmentation, we use the original byte-to-word ratio configuration, i.e., $5\times$ compression for Latin, $10\times$ compression for Cyrillic and $20\times$ compression for Indic languages within the same model. The second configuration; $(1, 2, 4)\times$ is the average bytes-to-character ratio for English, Russian and Telugu. Hence, using this configuration optimizes for fair byte-level-modelling with character-level granularity. The third configuration $(3, 6, 12)\times$ is based on a hypothesis that it would lead to fair subword-segmentation boundaries. Finally, since we apply a very high compression on Indic languages, we empirically test two additional configurations $(5, 10, 13)\times$ and $(5, 10, 15)\times$ with a reduced compression rate for Indic languages.

Table 1: Binomial prior choice for each MAGNET and DTP model configuration. These combinations of binomial priors determine the compression rate of sequences per language script. While DTP uses fixed priors for all languages, MAGNET is dynamic and script-specific.

	Binomial Prior		
Configuration	**Latin**	**Cyrillic**	**Indic**
DTP $5\times$	0.2	0.2	0.2
DTP $10\times$	0.1	0.1	0.1
MAGNET $(1, 2, 4)\times$	1	0.5	0.25
MAGNET $(3, 6, 12)\times$	0.33	0.17	0.083
MAGNET $(5, 10, 13)\times$	0.2	0.10	0.076
MAGNET $(5, 10, 15)\times$	0.2	0.10	0.066
MAGNET $(5, 10, 20)\times$	0.2	0.10	0.05

Subword Tokenizer Training To compare the segmentation derived from MAGNET to traditional subword tokenizers, we create byte-level byte pair encoding (BPE) vocabularies containing 50K, 100K and 250K subword units on our pretraining data. We employ α-sampling to train the tokenizers, typically used to improve representation of low-resource languages [10]. That is, we sample documents for each language according to a multinomial distribution with probabilities $\{q_i\}_{i=1...N}$, where: $q_i = \frac{p_i^\alpha}{\sum_{j=1}^{N} p_j^\alpha}$ with $p_i = \frac{n_i}{\sum_{k=1}^{N} n_k}$. This increases tokens for low-resource languages and has been shown to reduce bias towards high-resource ones. We consider $\alpha = 0.5, 0.3$ consistent with [10, 12].

Downstream Datasets To demonstrate the effectiveness of MAGNET, we evaluate by finetuning our trained models on several question answering and classification tasks. Specifically, we evaluate on XQuAD (question answering) [6], XNLI (natural language inference) [11], PAWS-X (paraphrase detection) [46] from XTREME [19], and SIB 200 (the topic classification) [2]. We provide a detailed language coverage across all tasks in Table 5. In addition, to test how adaptation capabilities of MAGNET, we evaluate on dialectal tasks, specifically ILI [48] the Indo-Aryan Language Identification (ILI) shared task and HaSCoSVa-2022 [7], hatespeech detection on Spanish dialects. We provide finetuning details in Appendix D.

3.2 Analyzing segmentation across models

The objective of this analysis is to compare segmentation granularity across different approaches. That is, we measure whether the same amount of information is conveyed through similar token counts across various languages. Following previous work [33, 3], we conduct this analysis with the parallel corpus FLORES-200 [16] focusing on the nine languages in our pretraining data.

For byte-level models, segmentation is done by converting each sentence to raw UTF-8 bytes and computing the average number of bytes per sentence. With the subword tokenizer, each sentence is segmented using the tokenizer we trained in 3.1, and the average number of resulting tokens computed across all sentences. As for gradient-based methods like DTP and our MAGNET, we feed each sentence into the model and retrieve a sequence of boundary predictions from the boundary predictor layers. The count of positive predictions determines the number of tokens per sentence.

4 Results

The goal of MAGNET is to learn equitable segmentation during training while maintaining high quality downstream performance. Ideally, we expect that MAGNET results in higher compression for the non-Latin script languages, hence balancing segmentation granularity across all languages. This, in turn, should improve modeling efficiency by reducing computational costs at training and inference time.

4.1 MAGNET results in equitable segmentation across language scripts.

We analyze the segmentation granularity, contrasting our method with byte-level, subword tokenization, and DTP as described in §3.2. Our results in Figure 2 show that MAGNET models produce similar segmentation rates for all languages. The improvement is particularly noticeable in non-Latin script languages that are most susceptible to over-segmentation with the baselines.

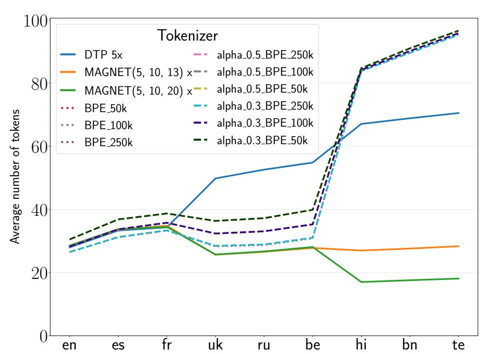


Figure 2: Average number of tokens after segmenting the FLORES dataset. Subword tokenizers and DTP result in over-segmentation in non-Latin script languages, while MAGNET closes the gap.

First, we compare byte-level segmentation to MAGNET with the $(1, 2, 4)\times$ segmentation configuration. As described in §3.1, Indic languages have approximately four byte code-points to one character, Cyrillic languages have approximately two, and many Latin languages have a one-to-one mapping. Therefore, we expect that training MAGNET with the $(1, 2, 4)\times$ configuration will result in equitable byte-level modeling across all of these languages. Appendix Figure 7a shows that MAGNET $(1, 2, 4)\times$ results in a $3\times$ drop in the average number of tokens for the Indic languages, and close to $2\times$ drop for Cyrillic, while the Latin languages are not affected. Next, we compare segmentation between DTP $5\times$, MAGNET at $(5, 10, 13)\times$, MAGNET at $(5, 10, 20)\times$ and subword-tokenizers at vocabulary sizes 50k, 100k and 250k with and without alpha sampling (see §3.1). We find that MAGNET models result in the most equitable segmentation across all languages. In fact, we measure a drop close to $5\times$ in the average number of tokens for the Indic languages compared to DTP and the subword tokenizers. Notably, we also see that a subword tokenizer with a large vocabulary size is required to achieve a lower segmentation rate on Cyrillic languages, whereas for Indic languages, even with a large vocabulary and α sampling, we observe a pronounced disparity. This contrasts with findings from previous work that α sampling alleviates tokenization disparities [10]. Overall, these results suggest that MAGNET learns equitable segmentation across languages with diverse scripts, while fixed segmentation models like DTP and byte-based subword tokenizers are sub-optimal and very likely to result in over-segmentation.

4.2 MAGNET maintains performance on downstream tasks.

Our goal is to enforce equitable segmentation while maintaining model performance across tasks. In Table 2, we present results for the best-performing MAGNET model compared to DTP and byte-level models (we provide comparisons to all MAGNET models in §5.1). Overall, We find that MAGNET models perform better than DTP but are competitive with the byte-level models while being considerably faster and requiring less compute. MAGNET$(3, 6, 12)\times$ performs best on PAWS-X and SIB, while $(1, 2, 4)\times$ performs best on XNLI and XQUAD.

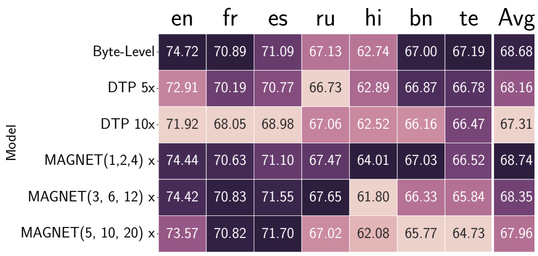


Figure 3: Language-specific accuracy on the XNLI task across byte-level, and all DTP and MAGNET models. Results are mostly competitive between byte-level and MAGNET models on Latin languages and Russian.

We report language-specific results on the XNLI dataset in Figure 3. We see better results with the MAGNET models even in some cases where the models in comparison optimize for a similar segmentation. For instance, on Spanish, MAGNET at $(5, 10, 15)\times$ outperforms DTP at $5\times$. On Indic languages, MAGNET models are generally competitive with the DTP models. We provide more analysis on the trade-offs between downstream performance and segmentation in §5.1. Results on dialectal tasks are reported in §2 We see competitive results across all models on all the tasks, suggesting that there is also no negative impact as a result of adapting the models to their respective dialects.

Table 2: The average performance (accuracy) on downstream tasks on all languages across different models. We present results for the best-performing MAGNET model: $((3, 6, 12)\times$ for PAWS-X and SIB), $((1, 2, 4)\times$ for XQUAD and XNLI). Bold indicates the best overall performance.Table 6 provides more detailed language-specific results.

Model	XNLI	PAWS-X	SIB	XQUAD
Byte-level	68.68	82.18	71.05	**44.62**
DTP5×	68.16	81.29	69.17	43.31
DTP10×	67.31	75.99	67.83	35.83
MAGNET	**68.74**	**85.41**	**71.43**	44.61

(a) In-language tasks.

Hascova	ILI
87.38	89.24
86.87	89.17
87.62	88.72
87.25	**89.27**

(b) Dialectal tasks.

4.3 MAGNET results in more efficient models.

Comparing inference times across all models, we expect that models which optimize for fixed compression like DTP would be only efficient for Latin languages because of their lower byte-to-word ratio. Hence, we anticipate that our routing strategy with the MAGNET models would result in an efficiency gain for non-Latin script languages. In Figure 4, we plot the inference time per language in XQUAD, relative to the inference time of the byte-level models. We show that MAGNET has a shorter inference time than the byte-level models, comparable to DTP for English and Russian and slightly lower for Hindi and Russian. If we assume the optimal compression rates for English and Russian to be $5\times$ and $10\times$, respectively, using a DTP model with a fixed compression rate for both languages requires training two separate monolingual models to obtain the ideal compression rate for each. However, training a single MAGNET $(5, 10, 20)\times$ model that dynamically achieves $5\times$ compression rate for English and $10\times$ compression rate for Russian results in a lower inference time for both.

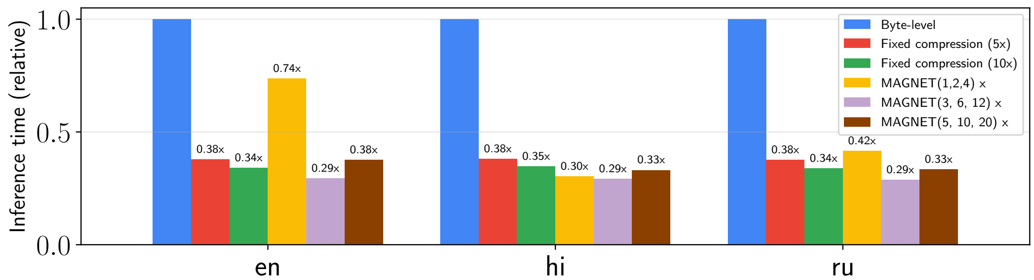


Figure 4: Inference time per language in XQUAD, relative to the byte-level model. MAGNET's inference time is shorter than the byte-level model and comparable to DTP for most of the languages.

5 Analysis and Discussion

5.1 Trade-off between downstream performance and equitable segmentation

Previous studies have reported correlations between compression and model performance [14]. We empirically investigate these tradeoffs by comparing downstream performance across different MAGNET configurations defined in Table 1. In the results reported in Table 3, we find that the configurations that perform best are MAGNET$(1, 2, 4)\times$ and MAGNET$(3, 6, 12)\times$. These configurations are equivalent to fair byte-level and subword modelling across all languages. We also report the average task performance per script at various compression rates in Figure 5 . Here we are not looking to compare performance across language scripts, but rather to assess performance across different compression rates within each language script. Our results show that there is little to no drop in performance for Latin and Cyrillic languages as compression increases moderately. However, for Indic languages, we see an average of 5% drop in performance as compression increases.

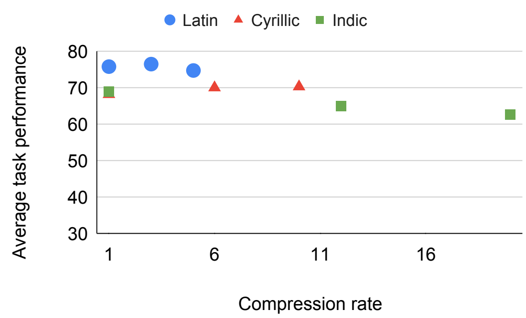


Figure 5: Average task performance vs compression trade off across language scripts.

Table 3: Results (accuracy) from ablations across all MAGNET configurations.

Model	XNLI	PAWSX	SIB	XQUAD
MAGNET$(1, 2, 4)\times$	**68.74**	83.30	71.02	**44.61**
MAGNET $(3, 6, 12)\times$	68.35	**85.41**	**71.43**	43.72
MAGNET $(5, 10, 20)\times$	67.50	80.44	71.13	41.06
MAGNET $(5, 10, 13)\times$	66.28	79.85	70.64	41.92
MAGNET $(5, 10, 15)\times$	67.96	84.79	68.79	42.81

(a) In-language tasks

Hascova	ILI
86.75	88.62
87.13	88.57
87.25	**89.27**
87.88	88.35
88.37	88.67

(b) Dialectal tasks

5.2 What is the granularity of segmentation across different compression rates?

In §3.1, we highlight that the binomial prior is essential for determining the granularity of the segmentation derived from the boundary predictor. To intrinsically validate that MAGNET indeed learns segmentations of similar lengths across different languages, we manually analyze examples from the SIB corpus, comparing them with DTP at 5× and 10×. As shown in Table 4 DTP at 5× produces word-level segmentation for all Latin languages while producing subword-level segmentation for Cyrillic and Indic languages. At 10×, we see word-level segmentation for Cyrllic languages, phrase-level segmentation for Latin languages and a mix of subword and word-level segmentation on Indic languages. To achieve word-level segmentation for all languages, DTP requires training three separate models. However MAGNET alleviates this requirement by producing a similar segmentation granularity across all the languages. For comparison, the segmentation granularity of the BPE tokenizer is highly sub-optimal for Indic languages as shown in Appendix Table 4. While the BPE tokenizer produces word-level segmentations for Latin and Cyrillic languages, it produces character-level segmentation for Indic languages. MAGNET, on the other hand, finds a good balance of segmentation granularity across languages.

Table 4: Segmentation of English, Spanish, Hindi, Russian, Ukranian and Telugu examples from the SIB corpus. While other models produce different segmentation granularity across languages, MAGNET consistently produces similar segmentation granularity across languages.

<table>
<tr><th>Lang</th><th>Text</th><th>DTP 5x</th><th>DTP 10x</th><th>MAGNET$(5, 10, 20)\times$</th><th>BPE 100k</th></tr>
<tr><td>en</td><td>This will allow players to control actions and movements in video games by moving the device through the air.</td><td>This ‖ will ‖ allow ‖ players ‖ to ‖ con ‖ trol ‖ action ‖ s ‖ and ‖ movemen ‖ ts ‖ in ‖ video ‖ games ‖ by ‖ moving ‖ the ‖ device ‖ through ‖ the ‖ air.</td><td>This will ‖ allow players ‖ to control ‖ actions ‖ and movements ‖ in video ‖ games ‖ by moving ‖ the device ‖ through ‖ the air.</td><td>This ‖ will ‖ allow ‖ players ‖ to ‖ con ‖ trol ‖ actions ‖ and ‖ movemen ‖ ts ‖ in ‖ video ‖ games ‖ by ‖ moving ‖ the ‖ device ‖ through ‖ the ‖ air.</td><td>This ‖ will ‖ allow ‖ players ‖ to ‖ control ‖ actions ‖ and ‖ movements ‖ in ‖ video‖ games‖ by‖ moving‖ the ‖ device ‖ through ‖ the ‖ air ‖.</td></tr>
<tr><td>es</td><td>Esto permitirá a los jugadores controlar las acciones y los movimientos en los videojuegos, a través del movimiento del dispositivo por el aire.</td><td>Es ‖ to ‖ permitir ‖ á ‖ a los ‖ jugadores ‖ con ‖ trolar ‖ las ‖ accion ‖ es ‖ y ‖ los ‖ movimientos ‖ en ‖ los ‖ videojuegos, ‖ a ‖ través ‖ del ‖ movimiento ‖ del ‖ dispositivo ‖ por ‖ el ‖ aire.</td><td>Esto permitirá ‖ a los ‖ jugadores ‖ controlar ‖ las acciones ‖ y los ‖ movimientos ‖ en los ‖ videojuegos ‖ , a través ‖ del movimiento ‖ del dispositivo ‖ por el aire.</td><td>Es ‖ to ‖ permitir ‖ á ‖ a ‖ los ‖ jugadores ‖ con ‖ trolar ‖ las ‖ acciones ‖ y ‖ los ‖ movimien ‖ tos ‖ en ‖ los ‖ videojuegos, ‖ a ‖ través ‖ de ‖ l movimien ‖ to ‖ de ‖ l dispositivo ‖ por ‖ e ‖ l aire.</td><td>Esto‖ permitirá‖ a‖ los‖ jugadores‖ controlar‖ las‖ acciones‖ y‖ los‖ movimientos‖ en‖ los‖ videojuegos‖,‖ a‖ través‖ del‖ movimiento‖ del‖ dispositivo‖ por‖ el‖ aire‖.</td></tr>
<tr><td>te</td><td>పరికరాన్ని గాల్లో కదిలించడం ద్వారా వీడియో గేమ్స్‌లో యాక్షన్లు మరియు కదలికలను నియంత్రించడానికి ఇది ఆటగాళ్లకు వీలు కల్పిస్తుంది</td><td>ప ‖ రి ‖ కరా ‖ న్ని ‖ గా ‖ ల్లో ‖ కది ‖ లి ‖ ంచ ‖ డం ‖ ద్వా ‖ రా ‖ వీ ‖ డి ‖ యో ‖ గే ‖ మ్స ‖ లో ‖ యా ‖ క్ష ‖ న్లు ‖ మరి ‖ యు ‖ కద ‖ లి ‖ కల ‖ ను ‖ ని ‖ యంత్రి ‖ ంచ ‖ డా ‖ ని ‖ కి ‖ ఇది ‖ ఆట ‖ గా ‖ ళ్ల ‖ కు ‖ వీ ‖ లు ‖ కల్పి ‖ స్త ‖ ుం ‖ ది.</td><td>పరికరాన్ని గాల్లో కదిలించడ ‖ ం ద్వారా ‖ వీడియో గ ‖ ేమ్స్‌లో యాక్షన్ల ‖ ు మరియు కదలికలన ‖ ు నియంత్రించడానికి ‖ ఇది ‖ ఆటగాళ్లక ‖ ు వీలు కల్పిస్త ‖ ుంది</td><td>పరికరాన్ని గాల్లో కదిలించడ ‖ ం ద్వారా వీడియో గేమ్స్‌లో యాక్షన్లు మరియు కదలికలన ‖ ు నియంత్రించడానికి ఇది ఆటగాళ్లకు వీల ‖ ు కల్పిస ‖ ్తుంది</td><td>పరి ‖ కరా ‖ న ‖ ్ని ‖ గా ‖ ల ‖ ్లో ‖ కది ‖ లిం ‖ చడ ‖ ం ‖ ద ‖ ్వా ‖ రా ‖ వీ ‖ డి ‖ యో ‖ గే ‖ మ ‖ ్స ‖ ‌లో ‖ ‖ యా ‖ క ‖ ్ష ‖ న ‖ ్లు ‖ మరి ‖ యు ‖ కదలి ‖ కల ‖ ను ‖ ని ‖ యం ‖ త ‖ ్రిం ‖ చడా ‖ ని ‖ కి ‖ ఇది ‖ ‖ ఆటగా ‖ ళ ‖ ్ల ‖ కు ‖ వీ ‖ లు ‖ కల ‖ ్పి ‖ స ‖ ్తుం ‖ ది</td></tr>
<tr><td>hi</td><td>यह डिवाइस को हवा में मूव करके खिलाड़ियों को एक्शन और मूवमेंट कंट्रोल करने की अनुमति देगा.</td><td>य || ह || डि || वा || इस || को || हव || ◌ा || मे || ◌ं || मू || व || कर || के || खि || ला || ड़ि || यो || ◌ं || को || एक || ◌्श || न || और || मू || व || मे || ◌ंट || कं || ट || ◌्र || ◌ो || ल || कर || न || ◌े || की || अ || न || ◌ु || मत || ◌ि || दे || ग || ा.</td><td>य || ह डिवाइस क || ◌ो ह || वा म || ◌ें म || ◌ूव क || रके ख || ◌िलाड़ियों क || ◌ो ए || क || ◌्शन और म || ◌ूवम || ◌ेंट क || ◌ंट्र || ◌ोल क || रने क || ◌ी || अन || ◌ुमति द || ◌ेगा.</td><td>यह डिवाइस क || ◌ो हवा म || ◌ें म || ◌ूव करके खिलाड़िय || ◌ों क || ◌ो एक्शन और म || ◌ूवमेंट कंट || ◌्रोल करन || ◌े की अन || ◌ुमति द || ◌ेगा.</td><td>यह || ड ||◌ि ||व ||◌ा ||इस || क ||◌ो || हव ||◌ा || म ||◌ें || म ||◌ू ||व || करक ||◌े || ख ||◌ि ||ल ||◌ा ||ड़ ||◌ि ||य ||◌ों || क ||◌ो || एक ||◌् ||शन || और || म ||◌ू ||वम ||◌ें ||ट || क ||◌ं ||ट ||◌् ||र ||◌ो ||ल || करन ||◌े || क ||◌ी || अन ||◌ु ||मत ||◌ि || द ||◌े ||ग ||◌ा</td></tr>
<tr><td>ru</td><td>Посредством движения устройства в воздухе игроки смогут управлять действиями и движениями в видеоиграх.</td><td>П || о || сре || дств || о || м || дви || жени || я || устро || йств || а || в || во || здухе || и || гро || ки || смо || гут || упра || вля || ть || де || йств || и || ями || и || дви || жени || ями || в || ви || део || и || гра || х.</td><td>Посред || ство || м || движен || ия || устрой || ства || в воздухе || игроки || смогут || управлять || действи || ями || и движен || иями || в видеоигра || х</td><td>По || средством || движе || ния || устройства || в воздухе || и || гроки || смогут || управлять || действ || и || ями || и движе || ниями || в видеои || грах</td><td>Пос || ред || ством || движения || устройства || в || воздухе || игроки || смогут || управлять || действия || ми || и || движ || ениями || в|| видео || иг || рах ||.</td></tr>
<tr><td>uk</td><td>Це забезпечить гравцям контроль над діями та рухами у відеоіграх, рухаючи пристрій у повітрі.</td><td>Подор || ож н || а мі || сце || можн || а зр || учно по || єдна || ти з пр || огулянкою озе || ром || у човні.</td><td>Це || забезпечить || гравцям || контро || ль || над діями || та || рухами || у відеоігра || х || , рухаючи || пристрій || у повітрі.</td><td>Це || забезпечить || гравцям || контроль || над діями || та рухами || у відеоіграх, || рухаю || чи || пристрій || у повітрі.</td><td>Це || забезпеч ||ить || грав ||цям || контроль || над || діями || та || рух ||ами || у || відео ||іг ||рах ||, || рух ||аючи || пристрій || у || повітрі ||.</td></tr>
</table>

5.3 Does segmentation significantly change after finetuning?

We investigate the effects of fine-tuning on the resulting segmentation boundaries across various downstream tasks. Essentially, we inspect how the boundary prediction changes after fine-tuning for each downstream task. We find that there are no differences in segmentation before and after fine-tuning despite updating the parameters of the boundary predictors. While there are a few instances where the segmentation of the fine-tuned model is different than that of the pretrained model, there is no clear evidence of the segmentation changing drastically after fine-tuning. In Table 7 in the Appendix, we present two examples from the SIB dataset where there is a slight change in segmentation. We found no indication that any observed changes contribute to or hurt task performance.

6 Related Work

Overcoming segmentation disparities in subword tokenization In multilingual settings, subword tokenizers have proven to be prone to over-segmentation, due to the data-driven nature of the BPE algorithm [39]. Previous work [13, 12, 44] has attempted to address data imbalance issues in subword tokenizers by over-sampling low-resource languages. Our work shows that this only alleviates the bias on certain scripts and doesn't solve the problem. Other studies [1, 36, 17] have also shown that tokenization in transformers remains biased in favor of high-resource languages. Wang et al. [42] enforce models to use smaller subword units in high-resource languages to make segmentation fairer. Some works [23, 8] suggest training multilingual tokenizers on language clusters to mitigate segmentation disparities, however, this leads to expanded vocabularies. Despite these attempts, it is evident that the training objectives of subword tokenizers do not effectively align with those of language modeling.

Tokenizer-free language models Language modelling over bytes [45, 4] and pixels [37, 38, 26] has become desirable, as it removes complicated preprocessing pipelines in modelling. Xue et al. [45] introduced ByT5, a tokenizer-free variant of T5 [35] that processes text at the byte level. However byte-level encoding over-fragments non-Latin script languages resulting in overly long sequences. Since byte or character sequences usually result in longer sequences, previous work [30, 9, 41, 47, 31, 15] on tokenizer-free LMs has introduced novel model architectures to mitigate the computational overhead of processing raw character or byte text directly. These methods [9, 41, 47, 30] end up segmenting raw sequences into fixed/dynamic-size patches, which is not suitable for modelling over non-Latin scripts.

7 Conclusion

In this work, we introduce MAGNET, a gradient-based tokenization method to learn equitable segmentation across languages scripts in byte-level multilingual models. MAGNET dynamically routes byte-level sequences through language-script-specific internal boundary predictors trained to infer word boundaries through stochastic reparameterisation. We show that MAGNET enables us to learn token representations with the same granularity across languages compared to vanilla byte-level models and previous gradient-based tokenization approaches. Our analysis demonstrates that while there are indeed downstream performance trade-offs as a result of MAGNET inducing high compression on non-Latin script languages, we are still able to maintain downstream performance quality. Overall, our results hold promise for future research on equitable segmentation and text processing more generally.

Acknowledgments

We would like to thank the UW NLP community for valuable discussions of this work. We are grateful to Farhan Samir and Alisa Liu for discussions on experiments and analysis. This work was supported in part by NSF IIS 2113530. We also gratefully acknowledge support from the National Science Foundation under CAREER Grant No. IIS2142739, NSF grants No. IIS2125201 and IIS2203097, and gift funding from Google, MSR, and OpenAI.

References

[1] J. Ács. Exploring bert's vocabulary. *Blog Post*, 2019.

[2] D. Adelani, H. Liu, X. Shen, N. Vassilyev, J. Alabi, Y. Mao, H. Gao, and E.-S. Lee. SIB-200: A simple, inclusive, and big evaluation dataset for topic classification in 200+ languages and dialects. In Y. Graham and M. Purver, editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 226–245, St. Julian's, Malta, Mar. 2024. Association for Computational Linguistics. URL `https://aclanthology.org/2024.eacl-long.14`.

[3] O. Ahia, S. Kumar, H. Gonen, J. Kasai, D. Mortensen, N. Smith, and Y. Tsvetkov. Do all languages cost the same? tokenization in the era of commercial language models. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9904–9923, Singapore, Dec. 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.614. URL `https://aclanthology.org/2023.emnlp-main.614`.

[4] R. Al-Rfou, D. Choe, N. Constant, M. Guo, and L. Jones. Character-level language modeling with deeper self-attention. In *AAAI Conference on Artificial Intelligence*, 2018. URL `https://api.semanticscholar.org/CorpusID:52004855`.

[5] C. Arnett, P. D. Riviere, T. A. Chang, and S. Trott. Different tokenization schemes lead to comparable performance in spanish number agreement. *ArXiv*, abs/2403.13754, 2024. URL `https://api.semanticscholar.org/CorpusID:268536952`.

[6] M. Artetxe, S. Ruder, and D. Yogatama. On the cross-lingual transferability of monolingual representations. *CoRR*, abs/1910.11856, 2019.

[7] G. Castillo-lópez, A. Riabi, and D. Seddah. Analyzing zero-shot transfer scenarios across Spanish variants for hate speech detection. In Y. Scherrer, T. Jauhiainen, N. Ljubešić, P. Nakov, J. Tiedemann, and M. Zampieri, editors, *Tenth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2023)*, pages 1–13, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.vardial-1.1. URL `https://aclanthology.org/2023.vardial-1.1`.

[8] H. W. Chung, D. Garrette, K. C. Tan, and J. Riesa. Improving multilingual models with language-clustered vocabularies. In B. Webber, T. Cohn, Y. He, and Y. Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4536–4546, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.367. URL `https://aclanthology.org/2020.emnlp-main.367`.

[9] J. H. Clark, D. Garrette, I. Turc, and J. Wieting. Canine: Pre-training an efficient tokenization-free encoder for language representation. *Transactions of the Association for Computational Linguistics*, 10:73–91, 2022. doi: 10.1162/tacl_a_00448. URL `https://aclanthology.org/2022.tacl-1.5`.

[10] A. CONNEAU and G. Lample. Cross-lingual language model pretraining. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper_files/paper/2019/file/c04c19c2c2474dbf5f7ac4372c5b9af1-Paper.pdf`.

[11] A. Conneau, R. Rinott, G. Lample, A. Williams, S. R. Bowman, H. Schwenk, and V. Stoyanov. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018.

[12] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. Unsupervised cross-lingual representation learning at scale. In D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July 2020.

Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.747. URL https://aclanthology.org/2020.acl-main.747.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio, editors, *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, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

[14] W. Fleshman and B. V. Durme. Toucan: Token-aware character level language modeling. *ArXiv*, abs/2311.08620, 2023. URL https://api.semanticscholar.org/CorpusID:265213263.

[15] N. Godey, R. Castagné, É. de la Clergerie, and B. Sagot. MANTa: Efficient gradient-based tokenization for end-to-end robust language modeling. In Y. Goldberg, Z. Kozareva, and Y. Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2859–2870, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.207. URL https://aclanthology.org/2022.findings-emnlp.207.

[16] N. Goyal, C. Gao, V. Chaudhary, P.-J. Chen, G. Wenzek, D. Ju, S. Krishnan, M. Ranzato, F. Guzmán, and A. Fan. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538, 2022. doi: 10.1162/tacl_a_00474. URL https://aclanthology.org/2022.tacl-1.30.

[17] J. Hayase, A. Liu, Y. Choi, S. Oh, and N. A. Smith. Data mixture inference: What do bpe tokenizers reveal about their training data?, 2024. URL https://arxiv.org/abs/2407.16607.

[18] D. Hendrycks and K. Gimpel. Gaussian error linear units (gelus), 2023.

[19] J. Hu, S. Ruder, A. Siddhant, G. Neubig, O. Firat, and M. Johnson. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *CoRR*, abs/2003.11080, 2020.

[20] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rkE3y85ee.

[21] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL https://api.semanticscholar.org/CorpusID:6628106.

[22] T. Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. In I. Gurevych and Y. Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1007. URL https://aclanthology.org/P18-1007.

[23] D. Liang, H. Gonen, Y. Mao, R. Hou, N. Goyal, M. Ghazvininejad, L. Zettlemoyer, and M. Khabsa. XLM-V: Overcoming the vocabulary bottleneck in multilingual masked language models. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13142–13152, Singapore, Dec. 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.813. URL https://aclanthology.org/2023.emnlp-main.813.

[24] T. Limisiewicz, D. Malkin, and G. Stanovsky. You can have your data and balance it too: Towards balanced and efficient multilingual models. In L. Beinborn, K. Goswami, S. Muradoğlu, A. Sorokin, R. Kumar, A. Shcherbakov, E. M. Ponti, R. Cotterell, and E. Vylomova, editors, *Proceedings of the 5th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 1–11, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.sigtyp-1.1. URL https://aclanthology.org/2023.sigtyp-1.1.

[25] T. Limisiewicz, T. Blevins, H. Gonen, O. Ahia, and L. Zettlemoyer. Myte: Morphology-driven byte encoding for better and fairer multilingual language modeling. *ArXiv*, abs/2403.10691, 2024. URL https://api.semanticscholar.org/CorpusID:268512851.

[26] J. Lotz, E. Salesky, P. Rust, and D. Elliott. Text rendering strategies for pixel language models. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10155–10172, Singapore, Dec. 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.628. URL https://aclanthology.org/2023.emnlp-main.628.

[27] C. J. Maddison, A. Mnih, and Y. W. Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=S1jE5L5gl.

[28] S. J. Mielke, Z. Alyafeai, E. Salesky, C. Raffel, M. Dey, M. Gallé, A. Raja, C. Si, W. Y. Lee, B. Sagot, and S. Tan. Between words and characters: A brief history of open-vocabulary modeling and tokenization in nlp. *ArXiv*, abs/2112.10508, 2021. URL https://api.semanticscholar.org/CorpusID:245335281.

[29] B. Muller, A. Anastasopoulos, B. Sagot, and D. Seddah. When being unseen from mBERT is just the beginning: Handling new languages with multilingual language models. In K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.38. URL https://aclanthology.org/2021.naacl-main.38.

[30] P. Nawrot, S. Tworkowski, M. Tyrolski, L. Kaiser, Y. Wu, C. Szegedy, and H. Michalewski. Hierarchical transformers are more efficient language models. In M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, editors, *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1559–1571, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-naacl.117. URL https://aclanthology.org/2022.findings-naacl.117.

[31] P. Nawrot, J. Chorowski, A. Lancucki, and E. M. Ponti. Efficient transformers with dynamic token pooling. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6403–6417, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.353. URL https://aclanthology.org/2023.acl-long.353.

[32] P. J. Ortiz Su'arez, B. Sagot, and L. Romary. Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019, pages 9 – 16, Mannheim, 2019. Leibniz-Institut f"ur Deutsche Sprache. doi: 10.14618/ids-pub-9021. URL http://nbn-resolving.de/urn:nbn:de:bsz:mh39-90215.

[33] A. Petrov, E. La Malfa, P. H. S. Torr, and A. Bibi. Language model tokenizers introduce unfairness between languages. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.15425.

[34] J. Pfeiffer, I. Vulić, I. Gurevych, and S. Ruder. UNKs everywhere: Adapting multilingual language models to new scripts. In M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10186–10203, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.800. URL https://aclanthology.org/2021.emnlp-main.800.

[35] C. Raffel, N. M. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2019. URL https://api.semanticscholar.org/CorpusID:204838007.

[36] P. Rust, J. Pfeiffer, I. Vulić, S. Ruder, and I. Gurevych. How good is your tokenizer? on the monolingual performance of multilingual language models. In C. Zong, F. Xia, W. Li, and R. Navigli, editors, *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, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.243. URL https://aclanthology.org/2021.acl-long.243.

[37] P. Rust, J. F. Lotz, E. Bugliarello, E. Salesky, M. de Lhoneux, and D. Elliott. Language modelling with pixels. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=FkSp8VW8RjH.

[38] E. Salesky, N. Verma, P. Koehn, and M. Post. Multilingual pixel representations for translation and effective cross-lingual transfer. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13845–13861, Singapore, Dec. 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.854. URL https://aclanthology.org/2023.emnlp-main.854.

[39] R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. In K. Erk and N. A. Smith, editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162.

[40] X. Song, A. Salcianu, Y. Song, D. Dopson, and D. Zhou. Fast WordPiece tokenization. In M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2089–2103, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.160. URL https://aclanthology.org/2021.emnlp-main.160.

[41] Y. Tay, V. Q. Tran, S. Ruder, J. Gupta, H. W. Chung, D. Bahri, Z. Qin, S. Baumgartner, C. Yu, and D. Metzler. Charformer: Fast character transformers via gradient-based subword tokenization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=JtBRnrlOEFN.

[42] X. Wang, S. Ruder, and G. Neubig. Multi-view subword regularization. In K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 473–482, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.40. URL https://aclanthology.org/2021.naacl-main.40.

[43] S. Wu and M. Dredze. Are all languages created equal in multilingual BERT? In S. Gella, J. Welbl, M. Rei, F. Petroni, P. Lewis, E. Strubell, M. Seo, and H. Hajishirzi, editors, *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.repl4nlp-1.16. URL https://aclanthology.org/2020.repl4nlp-1.16.

[44] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. In K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.41. URL https://aclanthology.org/2021.naacl-main.41.

[45] L. Xue, A. Barua, N. Constant, R. Al-Rfou, S. Narang, M. Kale, A. Roberts, and C. Raffel. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306, 2022. doi: 10.1162/tacl_a_00461. URL https://aclanthology.org/2022.tacl-1.17.

[46] Y. Yang, Y. Zhang, C. Tar, and J. Baldridge. PAWS-X: A Cross-lingual Adversarial Dataset for Paraphrase Identification. In *Proc. of EMNLP*, 2019.

[47] L. YU, D. Simig, C. Flaherty, A. Aghajanyan, L. Zettlemoyer, and M. Lewis. MEGABYTE: Predicting million-byte sequences with multiscale transformers. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=JTmO2V9Xpz.

[48] M. Zampieri, P. Nakov, N. Ljubešić, J. Tiedemann, S. Malmasi, and A. Ali, editors. *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, Santa Fe, New Mexico, USA, Aug. 2018. Association for Computational Linguistics. URL https://aclanthology.org/W18-3900.

Appendix

A Limitations

The primary limitation of our work is the restricted resources available for the extensive experiments we carried out. This constrained the number of languages included in our pretraining data, the size of the pretraining data itself, and the size of the models. Nonetheless, we hypothesize that our current results will hold true when replicated with larger models, as MAGNET is likely to provide even greater benefits when integrated into large models. We leave this to future work to explore further. Another limitation is the performance-compression trade-offs associated with MAGNET, as some languages are sensitive to high compression rates. However we note that this is not universal across all tasks. In fact, we argue that MAGNET offers users the flexibility to optimize for their desired benefits. Finally, MAGNET also inherits some limitations from previous gradient-based tokenization methods [31, 30, 41] and the vast majority of segmentation methods. This approach may not be suitable for Semitic languages, where morphemes are discontinuous and vowels are interspersed between consonant roots for inflection or sometimes omitted. Computing byte-to-word ratios would be harder in CJK (Chinese, Japanese, and Korean) languages, as they do not mark word boundaries with space. However we note that our choice of languages was not influenced by space separation, but rather by linguistic diversity and the availability of data for downstream evaluation. MAGNET is very flexible and applicable to all languages that can be expressed with UTF-8, we use byte-word-ratio as a simple proxy to train our boundary predictors to learn equitable tokenization . We note that byte-word-ratio is not a compulsory proxy and for such languages other proxies can be used

B Broader Impacts Statement

In this work, we contribute to promoting equitable segmentation in multilingual language models across various language scripts. Our approach holds promise for enhancing the utility and efficiency of multilingual language models, particularly benefiting low-resourced and non-Latin script languages spoken by billions worldwide. We acknowledge limitations of our work in Appendix A, and strongly advise against unintended usage of the models. We will release our code and models to facilitate further research in this direction.

C Dataset Statistics

C.1 Pretraining data

Our pre-training data is obtained from the OSCAR dataset [32]. Due to resource constraints, we only pretrain our models on a subset of this dataset. The distribution of tokens across languages in displayed in Figure 6.

C.2 Downstream data

Table 5: Downstream language and task coverage

Dataset	Task	Languages
XNLI	Natural language inference	en, fr, es, ru, hi, bn , te
XQUAD	Question answering	en, es, ru, hi
PAWS-X	Adversarial paraphrase identification	en, fr, es
SIB-200	Topic classification	en, fr, es, ru, uk, be, bn, te, hi

D Technical Details

Data Preprocessing For all our datasets, we preprocess all text to raw UTF-8 bytes. In the MAGNET models, we add a unique script identifier to the front of every sequence that guides the models to route the sequence to the respective script boundary predictor.

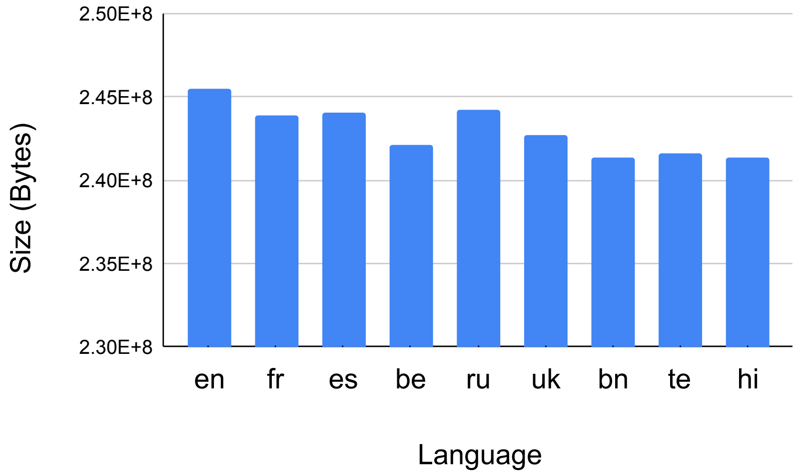


Figure 6: Language statistics in the pretraining data.

D.1 Model Hyperparameters

For all our experiments, we use 14-layer hourglass transformers with 2 layers in the first block, 10 layers in the second block and 2 layers in the final block. For every transformer layer, the hidden dimension is 768, the intermediate feed-forward dimension is 3072. Each self-attention layer consists of 12 heads. We use a post-norm architecture, GELU activation function [18] in feedforward layers and the relative attention parametrisation from Transformer XL. This brings our model's size to $\approx$ 126M parameters. The boundary predictor is a 2-layer MLP that takes in a hidden state as input and outputs a scalar prediction at each time step. We use the Adam optimizer [21] with (β_1, β_2) and ϵ parameters as (0.9, 0.98) and 1e-6, respectively. We use a learning rate of 5e-5, a warmup ratio of 0.1 and 38,000 training steps, a batch size of 512 distributed across 4 A40 GPUs. Each batch consists of examples concatenated up to the maximum sequence length of 512.

D.2 Finetuning

We finetune the pretrained models by appending a linear classification layer and update all parameters during training. The boundary predictors' parameters are also updated to ensure that the predicted segmentations are well adapted to each task. We finetune for 5 epochs with a batch size of 32 and the same learning rate and warm-up ratio that we used for pretraining. We report accuracy averaged over 2 runs with different random seeds.

E Supplementary Results

E.1 Equitable Segmentation at Byte-Level Granularity

In Figure 7, we present plots comparing segmentation granularity between (1.) byte-level segmentation and MAGNET$(1, 2, 4)\times$. (2) Between DTP $5\times$, MAGNET at $(5, 10, 13)\times$, MAGNET at $(5, 10, 20)\times$ and subword-tokenizers at vocabulary sizes 50k, 100k and 250k with and without alpha sampling (see section 3.1)

E.2 Downstream Tasks

We present language-level results (accuracy) across all tasks and models in Table 6

Table 6: Language-level results across all tasks

Model	en	fr	es	ru	hi	bn	te	Avg
Byte-Level	74.72	70.89	71.77	67.89	62.89	67.93	66.90	69.12
DTP 5×	73.18	70.19	69.78	66.96	61.79	67.11	66.12	66.74
DTP 10×	71.92	68.05	68.13	67.00	61.26	65.79	65.49	66.38
MAGNET$(1, 2, 4)$ x	74.44	70.63	71.99	67.48	64.73	67.44	66.88	69.17
MAGNET$(3, 6, 12)$ x	74.41	70.83	70.78	67.66	61.57	65.32	65.14	67.53
MAGNET$(5, 10, 20)$ x	73.71	70.23	71.15	66.35	61.72	65.12	65.01	67.13
MAGNET$(5, 10, 13)$ x	73.97	70.38	70.58	66.85	62.42	66.11	65.46	67.97
MAGNET$(5, 10, 15)$ x	73.57	70.82	71.36	67.02	61.89	65.43	64.96	67.86

(a) XNLI

Model	en	fr	es	Avg
Byte-Level	85.70	80.45	80.40	82.18
DTP 5×	84.13	80.20	79.55	81.29
DTP 10×	76.13	76.10	75.75	75.99
MAGNET$(1, 2, 4)$×	87.63	81.73	80.55	83.30
MAGNET$(3, 6, 12)$×	87.60	83.98	84.65	85.41
MAGNET$(5, 10, 20)$×	82.05	78.83	80.45	80.44
MAGNET$(5, 10, 13)$×	81.40	78.28	79.10	79.59
MAGNET$(5, 10, 15)$×	87.78	83.05	83.55	84.79

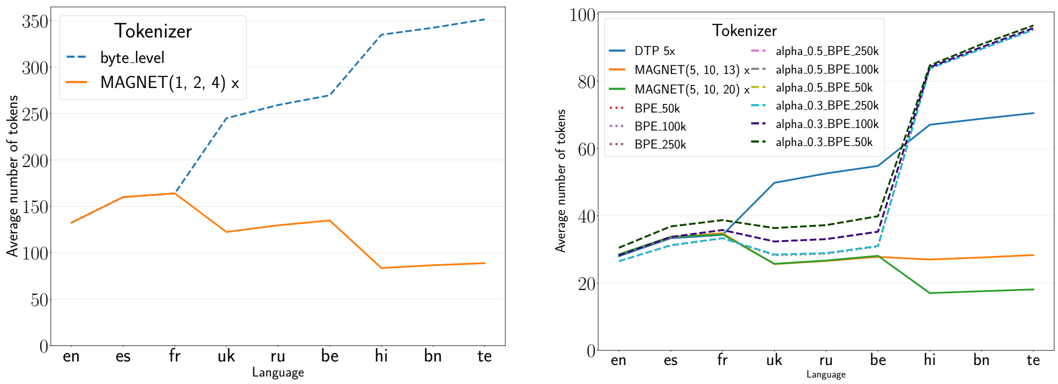
(b) PAWSX

en	es	ru	hi	Avg
55.61	53.225	40.49	29.18	44.62
51.40	49.85	41.62	30.39	43.31
40.68	37.75	38.09	26.79	35.82
55.94	51.78	41.75	28.98	44.61
55.32	52.46	41.29	25.82	43.72
53.12	50.29	39.61	21.23	41.06
53.13	50.16	38.31	26.07	41.92
53.14	51.88	39.82	26.41	42.81

(c) XQUAD

Model	en	es	fr	ru	be	uk	bn	te	hi	Avg
Byte-Level	76.96	72.06	71.57	75.98	71.07	72.55	63.00	68.63	67.65	71.05
DTP 5×	73.28	73.03	69.60	70.34	73.77	70.83	62.01	66.18	63.48	69.17
DTP 10×	73.28	68.38	68.14	67.89	70.83	67.89	63.24	66.67	64.21	67.83
MAGNET$(1, 2, 4)$×	75.25	73.04	72.55	75.00	70.09	70.59	63.24	66.43	73.04	71.02
MAGNET$(3, 6, 12)$×	79.17	77.94	75.69	75.00	75.74	73.29	56.62	59.31	56.13	69.87
MAGNET$(5, 10, 20)$×	80.15	78.92	77.70	75.98	76.47	74.02	49.75	53.90	51.96	68.76
MAGNET$(5, 10, 13)$×	79.17	79.66	73.78	71.57	73.78	71.33	59.07	65.68	61.76	70.64
MAGNET$(5, 10, 15)$×	80.64	78.92	74.80	72.30	76.23	72.31	56.37	52.94	54.66	68.79

(d) SIB-200



(a) MAGNET compared to byte-level model

(b) MAGNET compared to DTP and BPE

Figure 7: Average number of tokens after segmenting the FLORES dataset. Evidently subword tokenizers and DTP result in over-segmentation in non-Latin script languages, while MAGNET closes the gap.

Table 7: English and Russian instances from the SIB dataset showing slight changes in segmentation before and after fine-tuning. Segmentation for these examples was performed using the MAGNET (5x, 10x, 20x).

<table>
<tr><th>Lang</th><th>Text</th><th>Finetuning Segmentation</th><th>Pretraining Segmentation</th></tr>
<tr><td>en</td><td>Last month a presidential commission recommended the prior CEP’s resignation as part of a package of measures to move the country towards new elections.</td><td>Last || month || a || presiden || tial || commission || recommen || ded || the || prior || CEP’s || resign || a || tion || as || part || of || a || package || of || measures || to || move || the || country || towards || new || e || lec || tions.</td><td>Last || mon || th || a || presiden || tial || comm || ission || recommended || the || prior || CEP’s || resign || ation || as || part || of || a || package || of || measures || to || move || the || country || towards || new || e || lections.</td></tr>
<tr><td>ru</td><td>В прошлом месяце президентская комиссия рекомендовала предыдущему Временному избирательному совету уйти в отставку в качестве части пакета мер для движения страны к новым выборам.</td><td>В про || шлом || месяце || президе || нтс || кая || коми || сси || я || рекоме || ндовала || предыд || у || щему || Време || нному || избиратель || ному || совету || уйти || в отставку || в качестве || части || пакета || мер || для движе || ния || страны || к новым || выборам</td><td>В про || шлом || месяце || президе || нтс || кая || комиссия || рекоме || ндовала || предыду || щему || Време || нному || избиратель || ному || совету || уйти || в отставку || в качестве || части || пакета || мер || для движе || ния || страны || к новым || выборам.</td></tr>
</table>

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We demonstrate through extensive experiments and analysis that our approach leads to equitable segmentation rates across language scripts.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provide a "Limitations" section that highlights the shortcomings of our approach such as "extension to Semitic languages" and compute-constraints.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [No]

Justification: The paper doesn't include and theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide a detailed description of the model architecture, both visually and in-writing. We also provide details about our entire experimental-setup.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide a link to our code and data.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide details about our entire experimental setup inclusing model hyper parameters.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report results averaged over 2 runs with different random seeds.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provided details about the computational resources used in this project

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our work adheres to the NeurIPS Code of Ethics, there are no harmful impacts that could result from our methods.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: In the appendix we write a broader impact statement outlining the positive impacts of our work and discouraging malicious use.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our work poses no such risks

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets and assets are properly cited and attributed to the respective creators.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification:This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.