Searching for a Unified Approach to Morpheme Segmentation: Benchmarking Supervised and LLM Architectures Across 20 Languages

Anonymous ACL submission

Abstract

In this work, we compared various architectures of surface morpheme segmentation algorithms using data from 20 languages belonging to different language families and with varying data availability. We compare conventional subword and statistical baselines, CNN and LSTM models trained from scratch, fine-tuned multilingual encoders, prompted frontier LLMs, and LoRA-tuned decoder-only LLMs. To the best of our knowledge, this is the first systematic study of decoder-only LLMs for morpheme segmentation. Our results show that XLM-Roberta-large fine-tuned for character-level sequence labeling with whole-word context remains the strongest and most stable language-agnostic approach, achieving the highest average normalized score and improving previous SOTA results for five languages. At the same time, LoRA fine-tuning of Qwen3-32B yields competitive results and substantially outperforms XLM-Roberta-large for Dravidian and Indo-Aryan languages, demonstrating that decoder-only LLMs can be effectively adapted to the task. Prompting frontier LLMs, however, remains inconsistent. Finally, root-disjoint evaluation and error analysis show that generalization to unseen roots and dataset-specific segmentation conventions remain central challenges.

1 Introduction

Morpheme segmentation — the decomposition of words into their smallest meaning-bearing units — is important both for linguistic applications, such as language education (Kuo and Anderson, 2006; Volskaya et al., 2018; Xie et al., 2019), and for NLP engineering, including the construction of morphologically informed tokenizers. Recent work has shown that such tokenizers can improve language modeling or downstream performance for typologically diverse languages (García et al., 2025; Kildeberg et al., 2025; Hudspeth et al., 2025; Salehi and Jacobs, 2025; Nzeyimana and Niyongabo Rubungo, 2022; Asgari et al., 2025; Narzary et al., 2025; Teklehaymanot et al., 2025). At the same time, manually compiled morpheme dictionaries are inherently incomplete: for example, even the largest Russian resources cover only 100–150K unique lemmas, whereas large corpora contain at least 300K (Morozov et al., 2025). This makes automatic segmentation an important complement to manually compiled resources.

The best-performing automatic approaches are supervised neural models, including CNNs, LSTMs, Transformers, and fine-tuned BERT-like models. They have been evaluated for more than 25 languages from the Indo-European, Uralic, Turkic, Dravidian, and Bantu families (Sorokin, 2022; Batsuren et al., 2022; Peters and Martins, 2022; Pranjić et al., 2024; Morozov et al., 2025; Marquard et al., 2025; Supriya et al., 2025; Olbrich and Žabokrtský, 2026). However, direct comparison across studies remains difficult because datasets differ in size, annotation principles, segmentation depth, and evaluation metrics, raising the question of whether strong segmentation can be achieved beyond language- and dataset-specific settings.

A natural way to address this challenge is to search not for a universal theory of morpheme annotation, but for a language-agnostic modeling methodology: an approach that can be applied to typologically diverse languages without manual architectural adaptation. Thus, the approaches should not depend on the availability of language-specific pre-trained models (which may not exist for low-resource languages) and the availability of a detailed description of the segmentation principles (since such a description is absent for most available datasets). Previous comparative work suggests that such a methodology may be feasible. The SIGMORPHON 2022 Shared Task showed that a Transformer-based system can perform strongly across several languages and segmentation set-

tings (Batsuren et al., 2022; Peters and Martins, 2022), while Olbrich and Žabokrtský (2026) reported competitive CNN-based results across seven languages. Recent work has also explored pretrained multilingual encoders: LLMSegm (Pranjić et al., 2024) fine-tunes Glot500-m (Imani et al., 2023) for morpheme-boundary classification in low-resource settings. Nevertheless, existing studies leave two important gaps. First, they do not examine whether modern decoder LLMs, which have achieved strong performance across a wide range of NLP tasks, can be effectively adapted to morpheme segmentation. Second, the number and typological diversity of the languages considered in multilingual comparisons remain limited.

Seeking to address these gaps, we consider four groups of surface morpheme segmentation (Cotterell et al., 2016) methods: (i) CNN and LSTM architectures trained from scratch; (ii) fine-tuned pretrained multilingual BERT-like encoders; (iii) Low-Rank Adaptation (LoRA) (Hu et al., 2021) of open-weight decoder-only LLMs; and (iv) prompted frontier API-based LLMs. We aim to determine which of these promising approaches generalizes best to a large set of languages, including low-resource ones.

Our main contributions are as follows:

1. We provide a broad multilingual comparison of surface morpheme segmentation methods across 20 languages. For ten languages — Armenian, Croatian, Hindi, Kannada, Malayalam, Marathi, Persian, Swedish, Tajik, and Udmurt — we report, to the best of our knowledge, the first evaluation of neural surface morpheme segmentation. We show that XLM-Roberta-large (Conneau et al., 2020), fine-tuned for character-level classification, remains the strongest and most stable approach. Word-level segmentation accuracy ranges from 64% to 95%, and exceeds 90% for 6 out of 20 languages. For five languages — Belarusian, English, French, Italian, and Russian — our model outperforms previously reported SOTA results.

2. We present the first systematic evaluation of generative decoder LLMs for morpheme segmentation under both prompting and parameter-efficient fine-tuning. Prompting frontier API-based models is substantially weaker than supervised approaches and highly inconsistent: they achieved competitive or even superior results for a specific subset of languages (e.g., Armenian, Hindi, and Swedish), yet exhibited poor performance on others (e.g., Czech, English, and French). In contrast, LoRA-tuned Qwen3-32B yielded competitive results, substantially outperforming XLM-Roberta-large on Dravidian and Indo-Aryan languages, as well as on datasets annotated exclusively for inflectional morphemes.

3. We isolate the out-of-vocabulary-roots (OOV roots) problem using root-disjoint splits and show that all model families degrade on words whose roots are absent from training data, confirming that random-split accuracy can overestimate practical generalization.

2 Datasets

Most papers on morpheme segmentation focus on individual languages or language groups. Dataset creators employ diverse storage formats, tagsets, and, most crucially, differing segmentation principles. While the Universal Segmentations project (Žabokrtský et al., 2022) attempts to address this, only a few datasets have been converted to this format. Therefore, the selection and unification of datasets posed a particular challenge for our research.

Datasets for 12 languages were taken from the Universal Segmentations project[1] (original data source for each language is given in Appendix B): Armenian, Croatian, English, French, Hindi, Italian, Kannada, Malayalam, Marathi, Persian, Tajik, and Udmurt. We also used datasets for Belarusian (Morozov et al., 2025), Czech (Vidra et al., 2019), Swedish (Volodina et al., 2021), Russian (Morozov et al., 2024), and four Nguni languages: Zulu, Xhosa, Ndebele, and Swati (Gaustad and Puttkammer, 2022).

The selected datasets are extremely diverse in terms of size, content, and annotation types (see Table 1). After removing duplicate entries, the dataset sizes ranged from ca. 5,000 (Swati) to ca. 820,000 (Czech) segmentations. Some datasets contain segmentations for lemmata, while others contain segmentations of word forms. Another fundamental difference between the datasets is the chosen segmentation paradigm. In some datasets (in particular, for Armenian, Tajik, and Udmurt

[1] https://ufal.mff.cuni.cz/universal-segmentations

Language	ISO	Script	Type	Morpheme types	Size	Unique morph.	Avg mpw
Armenian	hye	Armenian	Wordforms	PR, ST, SU, H	530,328	77,862	2.01
Belarusian	bel	Cyrillic	Lemmata	PR, R, SU, E, IN, H	35,219	7,148	3.77
Croatian	hrv	Latin	Lemmata	PR, ST, SU, E, IN	15,373	4,830	3.60
Czech	ces	Latin	Lemmata	PR, R, SU	819,447	109,208	4.65
English	eng	Latin	Wordforms	PR, R, SU	68,624	25,788	1.87
French	fra	Latin	Wordforms	PR, R, SU	15,954	12,245	1.67
Hindi	hin	Devanagari	Wordforms	R, SU	8,275	7,589	1.12
Italian	ita	Latin	Lemmata	PR, R, SU, IN, H	11,081	6,323	2.49
Kannada	kan	Kannada	Wordforms	PR, R, SU, IN	32,217	17,789	1.90
Malayalam	mal	Malayalam	Wordforms	PR, R, SU	41,848	30,613	1.65
Marathi	mar	Devanagari	Wordforms	R, SU, IN	41,226	22,883	1.76
Ndebele	nbl	Latin	Wordforms	-	8,698	3,213	3.45
Persian	fas	Perso-Arabic	Wordforms	-	45,049	17,967	2.11
Russian	rus	Cyrillic	Lemmata	PR, R, SU, E, IN, PO, H	75,645	7,728	4.12
Swati	ssw	Latin	Wordforms	-	4,999	3,028	2.36
Swedish	swe	Latin	Lemmata	PR, R, SU, IN	12,525	4,566	2.60
Tajik	tgk	Cyrillic	Wordforms	PR, ST, SU, H	153,179	26,287	1.94
Udmurt	udm	Cyrillic	Wordforms	ST, SU, IN, H	258,599	39,678	1.99
Xhosa	xho	Latin	Wordforms	-	10,703	3,065	3.37
Zulu	zul	Latin	Wordforms	-	11,221	3,121	3.52

Table 1: Brief characteristics of the datasets. **Avg mpw** stands for average morpheme number per word. Morpheme types: PR — prefix, ST — stem, R — root, SU — suffix, H — connector/hyphen, E — ending/inflection, IN — interfix/connecting vowel or consonant, PO — postfix.

languages, the original source of which is the Uniparser family[2]), only inflectional suffixes are identified, and the annotation was performed automatically based on rules proposed by linguists, while others use deep manual segmentation, including the identification of historical affixes (e.g., for Russian and Croatian).

For Nguni languages we utilized the surface segmentation adaptation by Moeng et al. (2021) to ensure consistency, as native speaker verification was not available. A limitation of these datasets is the lack of morpheme type annotation. Morpheme type annotation is also missing from the Persian dataset in the Universal Segmentations collection. Therefore, only 15 of the datasets used contain morpheme type annotation.

3 Methods

Morpheme segmentation methods range from rule-based and statistical algorithms to neural sequence-labeling models and approaches based on pretrained language models. In this work, we compare four groups of methods. First, we include three conventional baselines encompassing subword tokenization and semi-supervised segmentation: the WordPiece tokenizer (Devlin et al., 2019) derived from `bert-base-multilingual-cased` (Devlin et al., 2019), the unigram language model (ULM) tokenizer (Kudo, 2018) extracted from `XLM-Roberta-large` (Conneau et al., 2020), and the semi-supervised Morfessor2.0 algorithm (Smit et al., 2014). These methods represent widely adopted statistical approaches and serve as reference points to quantify the gains achieved by fully supervised and pretrained neural models. Second, we consider task-specific neural models trained from scratch, including CNN and LSTM architectures (Morozov et al., 2024; Olbrich and Žabokrtský, 2026). Third, we evaluate fine-tuned pretrained encoders, which have shown strong results in prior work (Morozov et al., 2025; Marquard et al., 2025). Finally, we investigate generative decoder LLMs, which, to the best of our knowledge, have not previously been systematically evaluated for this task. We consider them in two distinct settings: parameter-efficient fine-tuning of open-weight models and prompting frontier API-based models.

3.1 CNN and LSTM Models

Among the models that do not rely on pretrained transformers, architectures based on convolutional neural networks (CNNs) and LSTMs are the most prevalent (Sorokin and Kravtsova, 2018; Morozov et al., 2024, 2025; Olbrich and Žabokrtský, 2026). In these approaches, the problem is formulated as a character-level classification task: each character is assigned a composite P-T label. Here, P denotes the position of the character within the morpheme and takes one of four values (B stands

[2] https://uniparser-morph.readthedocs.io/en/latest/

for beginning, M for middle, E for ending, and S for a single-character morpheme). T indicates the morpheme type (the set of possible types depends on the specific dataset). CNN and LSTM networks have demonstrated high performance in multiple languages and can be considered strong baselines for this task. For the CNN, we used a 10-layer character-level ResNet with a hidden size of 256, a filter size of 3, and a dropout rate of 0.1. For the LSTM, we utilized a two-layer character-level BiLSTM with a hidden dimension of 256 and a dropout rate of 0.1. Both models were trained for 30 epochs using the cross-entropy loss function. The hyperparameters were determined during preliminary experiments in English, Italian, and Russian.

3.2 BERT-like Models

Several studies have demonstrated that segmentation performance can be improved by utilizing pretrained BERT-like models (Sorokin, 2022; Pranjić et al., 2024; Morozov et al., 2025; Marquard et al., 2025). This is particularly crucial for processing OOV roots. Various approaches exist for incorporating BERT-like models, such as augmenting word feature representations with BERT embeddings in a CNN-based architecture (Sorokin, 2022). However, we consider the most promising approach to be fine-tuning a BERT-like model for character-level classification, similar to the CNN and LSTM baselines. In this setup, the model takes a character-tokenized word as input. The model is then tasked with predicting the composite P-T labels for all single-character tokens. Morozov et al. (2025) further improved this approach by appending the entire word to the input sequence. This whole-word token is assigned a special '0' label, which is ignored during loss calculation and evaluation.

We restricted our scope to multilingual models to avoid disadvantaging low-resource languages that might lack high-quality language-specific pretrained models. In preliminary experiments (see Appendix F.1), we compared `XLM-Roberta-large` (Conneau et al., 2020) with 550M parameters (hereinafter XLMR), `bert-base-multilingual-cased` (Devlin et al., 2019) with 177M parameters, and `distilbert-base-multilingual-cased` (Sanh et al., 2020) with 134M parameters. All models were fine-tuned for 30 epochs with a learning rate of 4×10^{-6}. For datasets smaller than 100,000 samples, we used a batch size of 16; for larger datasets (Armenian, Czech, Tajik, and Udmurt), a batch size of 256 was used to accelerate training due to GPU time constraints. In most cases, the XLMR model with the entire word appended to the input sequence achieved the best performance. Consequently, we adopted this configuration for all subsequent experiments.

3.3 Low-Rank Adaptation of Instruction-based LLMs

In the SIGMORPHON 2022 Shared Task (Batsuren et al., 2022), a sequence-to-sequence (seq2seq) approach utilizing a transformer trained from scratch achieved the best performance (Peters and Martins, 2022). Although adapting this approach to surface morpheme segmentation proved unsuccessful — the generated string frequently mismatched the source word (Morozov et al., 2025), — the applicability of instruction-tuned decoder-only models for morpheme segmentation remains unexplored. To address this, we investigated parameter-efficient fine-tuning of decoder LLMs with LoRA. We formulated morpheme segmentation as a single-turn chat generation task. Each training instance was represented in the model's chat format as a three-message dialogue: a system message specifying the task and the required output format, a user message containing only the target word, and an assistant message containing the gold segmentation. At inference time, the same system message and the target word were provided, and the model was expected to generate only the segmentation in the format `morpheme:TYPE/morpheme:TYPE/...`. The system prompt is shown in Appendix G.1.

Due to computational constraints, we limited our model selection to the widely adopted Qwen family of open-weights models, specifically evaluating Qwen2.5-32B-Instruct[3] (Team, 2024), Qwen3-4B[4], and Qwen3-32B[5] (Team, 2025). We tested full-precision versions of these models. All models were fine-tuned for three epochs with a learning rate of 2×10^{-4}, a LoRA α of 32, and a LoRA dropout of 0.05. For datasets with fewer than 100,000 samples, we used a training batch size of 8; for larger datasets (Armenian, Czech, Tajik, and Udmurt), we increased it to 32 to accelerate training. An inference batch size of 32 was used across all setups. In preliminary experiments (see Appendix F.3), the full-precision Qwen3-32B model achieved the best

[3] https://huggingface.co/Qwen/Qwen2.5-32B-Instruct

[4] https://huggingface.co/Qwen/Qwen3-4B

[5] https://huggingface.co/Qwen/Qwen3-32B

performance and was therefore adopted for all subsequent experiments.

3.4 Prompting Instruction-based LLMs

Frontier generative LLMs have shown strong performance on many NLP tasks and can be evaluated without task-specific fine-tuning via prompting. This makes them a relevant comparison point for morpheme segmentation, especially in settings where local training is impractical. However, to the best of our knowledge, there has been no systematic investigation into the applicability of prompt-based LLMs for the morpheme segmentation task.

To identify the most promising approach, we conducted a series of preliminary experiments evaluating various prompting strategies across three state-of-the-art LLMs released in the fall of 2025: Gemini 3 Pro Preview (Google), GPT-5.1 (OpenAI), and Claude 4.5 Sonnet (Anthropic). The models were tasked with generating a list of (morpheme, morpheme type) pairs. Examples of the prompts utilized are provided in Appendices G.2 (zero-shot approach) and G.3 (Corpus-to-Prompt Distillation approach). As detailed in Appendix F.4, few-shot prompting Gemini 3 Pro Preview with 50 random segmentations outperformed the other configurations in most cases. Consequently, this approach was adopted for all subsequent experiments.

3.5 Evaluation

Segmentation quality was evaluated using two metrics proposed by Sorokin and Kravtsova (2018): F-measure (hereinafter F1) for morpheme boundaries, and word-level accuracy (hereinafter WA). In general, for a generative approach, as in the case of LLMs, these metrics may not be indicative if the concatenation of the generated morphemes does not match the original word, as, for example, when testing the sequence-to-sequence character-level Transformer DeepSPIN-3 model (Peters and Martins, 2022) in (Morozov et al., 2025). However, in our experiments, the models very rarely violated the surface segmentation paradigm, meaning that the concatenation of morphemes almost always turned out to be equal to the word itself. Otherwise, we repeated the generation until a segmentation that met the conditions was generated.

For all approaches except LLM prompting, we evaluated segmentation performance using 5-fold cross-validation. Due to space limitations, for the cross-validation setups we are only presenting averaged values. The detailed results for each language are available in the repository[6]. Following Morozov et al. (2025), we also studied the robustness of supervised approaches in root-disjoint train-test split scenario (Subsection 4.1).

Due to financial constraints, the prompting approach was tested on a reduced test set of 500 words for each language. The test words were randomly selected from the first fold of the split used for supervised models, while the few-shot examples were selected from the remaining four folds.

4 Results

In the final experiments, we evaluated eight distinct models: three baselines (WordPiece, ULM, and Morfessor 2.0); CNN and LSTM models; XLM-Roberta-large fine-tuned with the entire word appended (XLMR+word); Qwen3-32B fine-tuned via LoRA; and Gemini 3 Pro Preview utilizing few-shot prompting (Gemini3+few). The results are presented in Table 2. The best results for each language are highlighted in bold. Additionally, the "ANSS" (Average Normalized Segmentation Score) column reports a normalized metric: the score expressed as a percentage of the maximum performance achieved for each language, averaged across all 20 languages.

As anticipated, the baselines yielded very low scores. In contrast, the character-level CNN and LSTM architectures, despite lacking massive pretraining, achieved high scores (exceeding 90% F1 on most languages) consistently with the results obtained previously by Morozov et al. (2025) and Olbrich and Žabokrtský (2026). The performance of these approaches indicates that relatively lightweight, character-focused architectures can successfully capture surface morphological patterns. The low computational cost makes these models attractive candidates for future experiments with morphology-informed tokenizers.

The XLMR+word model proved to be the most effective and stable approach, yielding the best performance on 12 out of the 20 languages, achieving the highest ANSS for both F1 (99.6%) and Word Accuracy (98.3%), and establishing itself as the state-of-the-art baseline for the majority of languages in the benchmark, including diverse families such as Slavic, Romance, and Bantu. Notably, the model's performance on Belarusian, English,

[6] https://anonymous.4open.science/r/morpheme-segmentation-050C

Metric	F1	WA	F1	WA	F1	WA	F1	WA	F1	WA	F1	WA	F1	WA
Language	**Armenian**		**Belarusian**		**Croatian**		**Czech**		**English**		**French**		**Hindi**	
WordPiece	45.0	0.7	57.6	2.7	53.5	1.8	48.8	0.6	60.1	16.9	58.2	15.7	45.6	7.8
ULM	57.0	6.5	59.9	3.5	56.9	2.0	55.7	0.8	64.6	21.0	62.2	19.4	60.5	27.7
Morfessor2.0	75.9	42.0	59.2	3.2	58.7	2.4	52.3	1.8	72.9	35.0	61.1	12.5	52.7	12.8
CNN	94.0	83.6	98.1	89.0	97.9	90.7	99.1	92.7	97.7	93.4	88.0	68.8	97.4	94.9
LSTM	93.9	84.5	98.2	89.0	98.1	91.3	99.1	92.8	97.6	93.0	88.2	68.9	97.6	95.2
XLMR+word	93.8	83.3	**98.8**	**92.4**	**98.2**	**92.7**	99.2	93.8	**98.2**	**95.3**	**89.8**	**74.5**	**98.4**	**97.1**
Qwen3-32B	93.9	87.2	98.3	90.8	97.1	89.1	**99.3**	**94.3**	97.8	94.9	88.1	73.6	97.9	**97.1**
Gemini3+few	**96.1**	**92.8**	94.7	71.8	96.6	86.0	84.8	44.8	92.0	82.2	83.9	62.4	97.6	95.2
Language	**Italian**		**Kannada**		**Malayalam**		**Marathi**		**Ndebele**		**Persian**		**Russian**	
WordPiece	55.5	7.4	42.9	2.2	26.5	0.4	46.8	2.9	52.1	3.9	62.1	9.2	56.7	2.9
ULM	62.1	14.5	58.7	16.5	40.2	9.5	61.8	17.3	55.3	5.6	71.6	22.6	64.2	5.6
Morfessor2.0	63.2	13.4	61.9	21.3	61.5	20.7	68.6	24.8	58.6	6.4	73.3	25.5	67.3	7.2
CNN	92.0	76.5	89.8	76.9	**89.4**	77.8	92.7	80.1	91.6	63.0	94.2	81.1	98.9	91.8
LSTM	91.9	75.4	89.0	74.8	88.1	75.3	92.5	79.4	91.7	63.1	94.3	81.5	98.8	91.4
XLMR+word	**92.5**	**78.5**	90.4	78.6	89.2	77.5	92.0	80.6	**92.1**	**64.6**	**95.8**	**86.1**	**99.2**	**94.3**
Qwen3-32B	88.7	72.9	**90.8**	**81.0**	88.5	**80.3**	**94.6**	**85.0**	90.5	60.4	95.5	85.7	99.1	93.7
Gemini3+few	86.1	60.6	83.2	69.8	84.3	73.2	90.4	79.0	88.8	55.7	93.6	79.0	95.2	76.8
Language	**Swati**		**Swedish**		**Tajik**		**Udmurt**		**Xhosa**		**Zulu**		**ANSS, %**	
WordPiece	53.7	6.6	63.3	16.0	49.2	2.2	52.4	3.1	51.2	2.6	50.9	2.5	54.1	6.4
ULM	54.3	8.3	72.0	27.0	52.1	3.9	52.5	6.0	55.9	5.3	56.3	5.1	61.5	13.4
Morfessor2.0	56.8	8.8	75.1	28.1	82.6	54.9	75.9	45.9	60.5	9.4	59.7	8.0	68.1	22.3
CNN	95.6	84.0	95.3	75.3	**95.2**	86.9	**95.6**	88.4	93.3	72.3	93.8	73.3	99.1	95.8
LSTM	**95.7**	**84.3**	95.5	75.8	94.7	84.9	95.4	88.2	93.7	74.1	94.3	75.1	99.0	95.6
XLMR+word	95.4	83.4	**97.4**	**84.5**	94.7	86.5	95.0	87.9	**93.9**	**75.2**	**94.6**	**76.6**	99.6	98.3
Qwen3-32B	92.7	77.9	95.4	80.9	94.6	**87.1**	94.9	**89.6**	92.3	70.7	93.3	73.1	98.8	97.1
Gemini3+few	93.0	73.4	94.8	83.4	94.2	86.1	92.9	84.4	90.9	62.4	89.9	58.4	95.6	86.2

Table 2: Surface morpheme segmentation quality across 20 languages. Scores are averaged over cross-validation folds for all models except Gemini3+few, which is evaluated on a single sample.

French, Italian, and Russian surpassed the previously reported state-of-the-art for morpheme segmentation (see Table 3). These results allow us to consider this model the most promising candidate for a universal, language-agnostic morpheme segmentation architecture.

Language	Previous SOTA	XLMR+word
Belarusian	90.5†	92.4
English	94‡	95.3
French	68‡	74.5
Italian	72‡	78.5
Russian	93.6†	94.3

Table 3: Comparison of the XLMR+word model with the previous state-of-the-art (SOTA) results in Word Accuracy. Previous SOTA scores are reported in (Morozov et al., 2025) (†) and (Olbrich and Žabokrtský, 2026) (‡).

The main limitations of XLMR+word were observed on the Uniparser family datasets (Armenian, Tajik, and Udmurt), as well as on Dravidian and Indo-Aryan languages[7]. Partly, this may be explained by model overfitting, which we observed in additional experiments with reduced training samples (see Appendix F.2). While there is no clear leader for the Uniparser datasets, the Qwen3-32B model fine-tuned using LoRA emerged as the absolute winner for Dravidian and Indo-Aryan languages and demonstrated highly competitive results, securing the second-best overall ANSS. Its superiority over other approaches for Kannada, Malayalam, and Marathi exceeded 2 percentage points in Word Accuracy. Thus, LoRA fine-tuning effectively adapts decoder-only Qwen3-32B model for the morpheme segmentation task. However, the significantly higher training costs compared to XLMR+word relegate this model to second place overall.

Gemini3+few showed mixed results: high quality compared to other algorithms for datasets from the Uniparser family and some other languages such as Swedish or Persian, but mediocre quality in many other cases, e.g., for Czech and Italian languages. Thus, it can be argued that prompting modern LLMs — at least using basic prompting strategies — cannot serve as a universal morpheme segmentation approach, as it remains significantly inferior to fully supervised methods.

[7]For Armenian, Malayalam, Swati, Tajik, and Udmurt, XLMR+word was not the best BERT-like model during preliminary experiments (see Appendix F.1).

Metric	F1	WA	F1	WA	F1	WA	F1	WA	F1	WA	F1	WA	F1	WA	F1	WA
Language	**Armenian**		**Belarusian**		**Croatian**		**Czech**		**English**		**French**		**Hindi**		**Italian**	
WordPiece	45.0	0.7	57.9	2.7	53.6	1.8	48.8	0.6	60.1	16.9	58.2	15.7	45.6	7.6	55.4	7.4
ULM	57.0	6.5	59.7	3.5	57.0	2.0	55.7	0.8	64.6	21.0	62.2	19.4	60.5	27.6	62.1	14.7
Morfessor2.0	68.0	19.4	54.1	1.4	57.3	3.1	53.8	1.2	62.1	8.0	57.8	5.3	51.2	7.6	59.4	4.3
CNN	87.4	71.9	94.4	69.0	94.6	77.2	96.3	70.4	89.3	74.2	83.6	59.8	95.6	92.0	85.8	63.0
LSTM	86.8	69.9	94.4	69.0	94.6	77.0	96.1	68.6	89.2	73.0	83.7	59.5	96.0	92.5	86.0	63.6
XLMR+word	**88.8**	75.4	**96.2**	**77.8**	**95.9**	**83.7**	**96.6**	**72.7**	**92.7**	**82.6**	**86.8**	**68.4**	**97.5**	**96.1**	**88.6**	**71.2**
Qwen3-32B	**88.8**	**76.4**	95.1	74.3	94.5	80.1	95.8	68.0	90.9	80.6	84.4	66.7	97.0	**96.1**	81.6	58.7
Language	**Kannada**		**Malayalam**		**Marathi**		**Russian**		**Swedish**		**Tajik**		**Udmurt**		**ANSS, %**	
WordPiece	42.8	2.1	26.4	0.4	46.8	2.9	57.3	3.2	61.6	18.0	49.2	2.2	52.4	3.1	55.4	7.4
ULM	58.7	16.6	40.3	9.6	61.8	17.3	63.9	5.7	67.6	27.1	52.1	3.8	52.5	6.0	63.8	15.6
Morfessor2.0	59.6	12.8	59.7	15.5	62.7	13.2	61.6	1.8	63.5	3.5	69.5	20.9	63.2	11.5	63.8	15.6
CNN	81.6	63.9	84.2	68.8	84.9	64.0	94.5	68.4	92.2	68.0	89.1	73.7	84.5	67.3	97.4	90.6
LSTM	81.5	63.2	83.7	67.5	84.8	63.6	94.8	69.1	92.6	68.7	88.4	71.6	84.2	66.1	97.3	89.8
XLMR+word	83.3	67.9	**85.8**	71.9	87.2	71.4	**96.7**	**79.9**	**94.4**	**75.9**	**90.7**	78.0	**86.8**	**72.7**	99.6	98.7
Qwen3-32B	**83.7**	**69.7**	**85.8**	**75.5**	**91.3**	**78.5**	95.0	74.6	90.2	69.9	90.6	**78.3**	85.2	70.5	98.4	96.3

Table 4: Surface morpheme segmentation quality across 20 languages for root/stem-based fold splitting. Scores are averaged over cross-validation folds.

4.1 OOV Roots Processing

Following Morozov et al. (2025), we evaluate robustness to OOV roots using a root-disjoint split, where words sharing a root or stem cannot appear in both training and test data. We exclude Persian and the four Nguni languages because they lack morpheme type annotation. Results are shown in Table 4, with the best results for each language highlighted in bold.

In this setting, XLMR+word remains the strongest model, achieving the best score in 10 out of 15 languages, while Qwen3-32B wins in four languages and is especially strong for Dravidian and Indo-Aryan datasets. All supervised models, however, degrade substantially compared to the random split, confirming that OOV-root generalization remains a central challenge. The drop in WA is typically at least six percentage points. A notable exception is Hindi, which we attribute to a combination of two factors: the predominance of single-morpheme words in the Hindi dataset and a generally low morpheme frequency (on average, each morpheme appears only about 1.2 times in the corpus). Consequently, for Hindi, the root-disjoint split differs little from a random split. The ANSS metric also highlights a widened performance gap for the CNN and LSTM architectures compared to the random split: while they previously trailed by only 2–3 percentage points, the gap increases to 6–8 points in the root-disjoint setting.

Finally, the decline in segmentation quality for Belarusian, Czech, and Russian is consistent with previously reported findings (Morozov et al., 2025). Furthermore, consistent with the random splitting scenario, our XLMR+word approach outperforms prior monolingual models on Belarusian and Russian by an additional 2–3 percentage points, setting a new robust baseline even in the zero-shot root setting.

5 Error Analysis

To better understand the differences between the strongest approaches, we analyze the errors of XLMR+word and Qwen3-32B. These models are not only among the best-performing systems in our experiments, but also represent substantially different paradigms: XLMR+word performs character-level sequence labeling with access to the whole input word, whereas Qwen3-32B generates the complete segmentation string autoregressively.

We aggregate by macro F1 predictions over all cross-validation folds and group languages according to both genealogical relatedness and dataset origin. We consider East Slavic (Russian, Belarusian), Dravidian (Malayalam, Kannada), Indo-Aryan (Hindi, Marathi), Nguni (Ndebele, Swati, Xhosa, Zulu), UniParser-based (Udmurt, Tajik, Armenian), and MorphoLex-based (English, French) groups; Italian, Swedish, Persian, Czech, and Croatian are analyzed separately. For each incorrect prediction, we assign an error type (see Table 5 and, for details, Appendix F.5).

The analysis shows that XLMR+word and Qwen3-32B often fail on the same segmentations: depending on the language, the Jaccard similarity between their sets of incorrectly segmented words ranges from 0.25 in English to 0.58 in Kannada and Malayalam (for details, see Appendix F.5). Also, for each language group, most errors can be attributed to one or several major types. These patterns

Group	XLMR+word	Qwen3-32B
East Slavic	root/suff over-split	root/suff over-split
Dravidian	root–suffix boundary shift, complex error	complex error, root–suffix boundary shift
Indo-Aryan	root over-split, missed root–suffix boundary	root over-split, missed root–suffix boundary
Nguni	prefix-chain over/under-split	prefix-chain over/under-split
UniParser	stem–suffix boundary shift	stem–suffix boundary shift
MorphoLex	root over-split, root–suffix boundary shift	root–suffix boundary shift, missed root–suffix boundary
Italian	root/suff over-split, root/suff under-split	root/suff over-split, root–suffix boundary shift
Swedish	type-only, root over-split	type-only, root over-split
Persian	morpheme over-split	morpheme over-split
Czech	root over-split, missed root–suffix boundary	missed root–suffix boundary, root over-split
Croatian	stem boundary shift, false initial prefix	stem boundary shift

Table 5: Dominant error clusters for XLMR+word and Qwen3-32B.

are broadly similar across the two models, although the distributions do not coincide exactly. For East Slavic, Indo-Aryan, MorphoLex, Italian, Persian, and Czech, a large proportion of errors made by both models is associated with excessively granular segmentation. Boundary shifts within morpheme pairs are also frequent, especially in the UniParser, MorphoLex, and Croatian groups. Importantly, the morphemes produced by over-segmentation and boundary-shift errors are often attested elsewhere in the corresponding gold data. Using weighted averages over languages, with the number of relevant errors as weights, we find that this effect is particularly strong for Qwen3-32B. In 79% of over-segmentation cases and 92% of boundary-shift cases, all resulting morphemes occur in the dataset morpheme inventory. The same tendency is also observed for XLMR+word, with 55% and 91%, respectively (for details, see Appendix F.5). This indirectly suggests that a substantial portion of errors may reflect inconsistencies in dataset annotation, such as non-systematic separation of historical suffixes from roots.

We further identify three groups with more specific error profiles. First, in the Dravidian languages, many errors are complex, combining boundary shifts with over- or under-segmentation. Second, in the Nguni languages, most errors involve over- or under-segmentation of prefix-chains. Finally, Swedish is dominated by type errors involving one-letter morphemes *-a-* and *-e-*, which may be annotated in the training data either as linking elements or as suffixes.

Overall, the dominant error types appear to be systematic and are often associated with recurring language- or dataset-specific patterns rather than isolated model failures. At the same time, our analysis does not claim to be exhaustive, and a detailed error analysis should be conducted in the future with the involvement of native speakers.

6 Conclusion

We presented a broad evaluation of surface morpheme segmentation methods across 20 typologically diverse languages and heterogeneous annotation resources. To the best of our knowledge, we presented the first systematic evaluation of decoder-only LLMs for this task under both prompting and LoRA fine-tuning. Our results show that fine-tuning XLM-Roberta-large for character-level sequence labeling with whole-word context remains the strongest and most stable language-agnostic approach: it achieves the highest Average Normalized Segmentation Score, i.e., the best average performance relative to the top score for each language, establishes new SOTA for five languages, and is the most robust model under root-disjoint evaluation. At the same time, our experiments demonstrate that decoder-only LLMs can be effectively adapted to morpheme segmentation: LoRA fine-tuning of Qwen3-32B yields competitive results and substantially outperforms XLMR+word for Dravidian and Indo-Aryan languages, although at a much higher computational cost. In contrast, prompting frontier LLMs produces inconsistent results and cannot yet be considered a reliable general-purpose segmentation strategy.

The OOV-root experiment confirms that generalization to unseen lexical material remains a central challenge for all supervised approaches, even when aggregate random-split scores are high. Our error analysis further shows that remaining errors are systematic, including root–suffix granularity, prefix-chain segmentation, stem–suffix boundary placement, and morpheme-type ambiguity. These findings suggest that broadly applicable morpheme segmentation approach is feasible, but further progress will require stronger modeling of unseen roots and more consistent across-language data annotation.

7 Limitations

Our study has several limitations. First, the experiments with prompted frontier LLMs were conducted on reduced test samples due to financial constraints. Although the performance gap between prompting and supervised models is large for most languages, larger-scale evaluation would be needed for languages where prompted models achieved competitive results. In addition, we tested only a limited set of prompting strategies. More sophisticated approaches, such as retrieval-augmented few-shot prompting with examples selected by similarity to the target word, may improve performance.

Second, our LoRA fine-tuning experiments were limited to the Qwen family of open-weight decoder LLMs. While Qwen3-32B proved competitive for several language groups, especially Dravidian and Indo-Aryan languages, evaluating other decoder-only model families would be necessary to determine whether these findings generalize beyond a single model line.

Third, the datasets used in this study differ substantially in size, annotation depth, morpheme type inventories, and segmentation principles. Some resources, such as the UniParser-based datasets, annotate only inflectional morphology and were produced automatically from linguistic rules, whereas others provide much deeper derivational segmentation. Hindi contains a large proportion of single-morpheme words, which makes its evaluation less comparable to datasets with richer segmentation. For Persian and the Nguni languages, morpheme type annotation is unavailable, limiting the range of comparable analyses.

Fourth, although our error analysis shows that many errors are systematic and often may reflect dataset-specific conventions, it remains only a first approximation. A more complete analysis would require native speakers or language experts for all languages, especially for cases where predicted segmentations may be linguistically plausible alternatives to the gold annotation.

Finally, our work evaluates intrinsic segmentation quality rather than the impact of the resulting segmentations on downstream NLP systems or morphologically informed tokenizers. While morpheme segmentation is motivated by both linguistic and engineering applications, future work should test whether the observed differences between segmentation models translate into improvements in language modeling, downstream tasks, or tokenizer efficiency.

References

Sachi Angle, Pruthwik Mishra, and Dipti Misra Sharma. 2018. Automated error correction and validation for POS tagging of Hindi. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, Hong Kong. Association for Computational Linguistics.

Ebrahim Ansari, Zdeněk Žabokrtský, Hamid Haghdoost, and Mahshid Nikravesh. 2019. Persian morphologically segmented lexicon 0.5. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL).

Timofey Arkhangelskiy and Maria Medvedeva. 2016. Developing morphologically annotated corpora for minority languages of Russia. In *Proceedings of Corpus Linguistics Fest*, volume 1607 of *CEUR Workshop Proceedings*, pages 1–6, Bloomington, Indiana, USA.

Ehsaneddin Asgari, Yassine El Kheir, and Mohammad Ali Sadraei Javaheri. 2025. Morphbpe: A morpho-aware tokenizer bridging linguistic complexity for efficient llm training across morphologies. *Preprint*, arXiv:2502.00894.

Shriya Atmakuri, Bhavya Shahi, Ashwath B Rao, and SN Muralikrishna. 2018. A comparison of features for POS tagging in Kannada. *International Journal of Engineering & Technology*, 7(4.24):18–22.

Khuyagbaatar Batsuren, Gábor Bella, Aryaman Arora, Viktor Martinovic, Kyle Gorman, Zdeněk Žabokrtský, Amarsanaa Ganbold, Šárka Dohnalová, Magda Ševčíková, Kateřina Pelegrinová, Fausto Giunchiglia, Ryan Cotterell, and Ekaterina Vylomova. 2022. The SIGMORPHON 2022 shared task on morpheme segmentation. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 103–116, Seattle, Washington. Association for Computational Linguistics.

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

Ryan Cotterell, Tim Vieira, and Hinrich Schütze. 2016. A joint model of orthography and morphological segmentation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, pages 664–669, San Diego, California. Association for Computational Linguistics.

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.

Alba Táboas García, Piotr Przybyła, and Leo Wanner. 2025. Exploring morphology-aware tokenization: A case study on Spanish language modeling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 30493–30506, Suzhou, China. Association for Computational Linguistics.

Tanja Gaustad and Martin J. Puttkammer. 2022. Linguistically annotated dataset for four official south african languages with a conjunctive orthography: Isindebele, isixhosa, isizulu, and siswati. *Data in Brief*, 41:107994.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Marisa Hudspeth, Patrick J. Burns, and Brendan O'Connor. 2025. Contextual morphologically-guided tokenization for latin encoder models. *Preprint*, arXiv:2511.09709.

Ayyoob Imani, Peiqin Lin, Amir Hossein Kargaran, Silvia Severini, Masoud Jalili Sabet, Nora Kassner, Chunlan Ma, Helmut Schmid, André Martins, François Yvon, and Hinrich Schütze. 2023. Glot500: Scaling multilingual corpora and language models to 500 languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1082–1117, Toronto, Canada. Association for Computational Linguistics.

D. M. Iskandarova. 2021. The tajik national corpus as a tool for linguistic research (in russ.). *Kazan Science*, 1:94–97.

Jisha P Jayan, Asha S Nair, and Govindaru V. 2017. Malayalam VerbFrames. In *Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*, pages 236–244, Kolkata, India. NLP Association of India.

Victoria Khurshudyan, Timofey Arkhangelskiy, Misha Daniel, Vladimir Plungian, Dmitri Levonian, Alex Polyakov, and Sergei Rubakov. 2022. Eastern Armenian national corpus: State of the art and perspectives. In *Proceedings of the Workshop on Processing Language Variation: Digital Armenian (DigitAm) within the 13th Language Resources and Evaluation Conference*, pages 28–37, Marseille, France. European Language Resources Association.

Mikkel Wildner Kildeberg, Emil Allerslev Schledermann, Nicolaj Larsen, and Rob van der Goot. 2025. From smør-re-brød to subwords: Training llms on danish, one morpheme at a time. *Preprint*, arXiv:2504.01540.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Li-jen Kuo and Richard C. Anderson. 2006. Morphological awareness and learning to read: A cross-language perspective. *Educational Psychologist*, 41(3):161–180.

Hugo Mailhot, Maximiliano A. Wilson, Joël Macoir, S. Hélène Deacon, and Claudia Sánchez-Gutiérrez. 2020. MorphoLex-FR: A derivational morphological database for 38,840 French words. *Behavior Research Methods*, 52(3):1008–1025.

Cael Marquard, Simbarashe Mawere, and Francois Meyer. 2025. Neural morphological tagging for nguni languages. In *Proceedings of the Sixth Workshop on African Natural Language Processing (AfricaNLP 2025)*, pages 210–220, Vienna, Austria. Association for Computational Linguistics.

Tumi Moeng, Sheldon Reay, Aaron Daniels, and Jan Buys. 2021. Canonical and surface morphological segmentation for nguni languages. *Preprint*, arXiv:2104.00767.

Dmitry Morozov, Lizaveta Astapenka, Anna Glazkova, Timur Garipov, and Olga Lyashevskaya. 2025. BERT-like models for Slavic morpheme segmentation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6795–6815, Vienna, Austria. Association for Computational Linguistics.

Dmitry Morozov, Timur Garipov, Olga Lyashevskaya, Svetlana Savchuk, Boris Iomdin, and Anna Glazkova. 2024. Automatic morpheme segmentation for Russian: Can an algorithm replace experts? *Journal of Language and Education*, 10(4):71–84.

Sanjib Narzary, Maharaj Brahma, Sukumar Nandi, and Bidisha Som. 2025. Bpe and morphologically segmented phrase based statistical machine translation system for indian languages to resource constrained language bodo. *Multimedia Tools and Applications*, 84(25).

Antoine Nzeyimana and Andre Niyongabo Rubungo. 2022. KinyaBERT: a morphology-aware Kinyarwanda language model. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5347–5363, Dublin, Ireland. Association for Computational Linguistics.

Michal Olbrich and Zdeněk Žabokrtský. 2026. Morphological segmentation with neural networks: Performance effects of architecture, data size, and cross-lingual transfer in seven languages. In *Text, Speech, and Dialogue*, pages 275–286, Cham. Springer Nature Switzerland.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Ben Peters and Andre F. T. Martins. 2022. Beyond characters: Subword-level morpheme segmentation. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 131–138, Seattle, Washington. Association for Computational Linguistics.

Marko Pranjić, Marko Robnik-Šikonja, and Senja Pollak. 2024. LLMSegm: Surface-level morphological segmentation using large language model. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10665–10674, Torino, Italia. ELRA and ICCL.

Hanumant Redkar, Sandhya Singh, Nandini Ghag, Jai Paranjape, Nilesh Joshi, Malhar Kulkarni, and Pushpak Bhattacharyya. 2016. Verbframator:semi-automatic verb frame annotator tool with special reference to Marathi. In *Proceedings of the 13th International Conference on Natural Language Processing*, pages 299–304, Varanasi, India. NLP Association of India.

Ali Salehi and Cassandra L. Jacobs. 2025. Subword tokenization strategies for kurdish word embeddings. *Preprint*, arXiv:2511.14696.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *Preprint*, arXiv:1910.01108.

Svetlana O Savchuk, Timofey Arkhangelskiy, Anastasiya A Bonch-Osmolovskaya, Ol'ga V Donina, Yuliya N Kuznetsova, Ol'ga N Lyashevskaya, Boris V Orekhov, and Mariya V Podryadchikova. 2024. Russian National Corpus 2.0: New opportunities and development prospects. *Voprosy Jazykoznanija*, 2024(2):7–34.

Peter Smit, Sami Virpioja, Stig-Arne Grönroos, and Mikko Kurimo. 2014. Morfessor 2.0: Toolkit for statistical morphological segmentation. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 21–24, Gothenburg, Sweden. Association for Computational Linguistics.

Krešimir Šojat, Matea Srebačić, Marko Tadić, and Tin Pavelić. 2014. CroDeriV: a new resource for processing Croatian morphology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3366–3370, Reykjavik, Iceland. European Language Resources Association (ELRA).

Alexey Sorokin. 2022. Improving morpheme segmentation using bert embeddings. In *Analysis of Images, Social Networks and Texts*, pages 148–161, Cham. Springer International Publishing.

Alexey Sorokin and Anastasia Kravtsova. 2018. Deep convolutional networks for supervised morpheme segmentation of Russian language. In *Artificial Intelligence and Natural Language*, pages 3–10, Cham. Springer International Publishing.

Musica Supriya, Dinesh Acharya Udupi, Ashalatha Nayak, and Arjuna Srirangapatna Raghavendra. 2025. Developing a hybrid morphological analyzer for low-resource languages. *Applied Sciences*, 15(10).

Claudia H. Sánchez-Gutiérrez, Hugo Mailhot, S. Hélène Deacon, and Maximiliano A. Wilson. 2018. MorphoLex: A derivational morphological database for 70,000 English words. *Behavior Research Methods*, 50(4):1568–1580.

Luigi Talamo, Chiara Celata, and Pier Marco Bertinetto. 2016. Derivatario: An annotated lexicon of italian derivatives. *Word Structure*, 9(1):72–102.

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

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Hailay Kidu Teklehaymanot, Dren Fazlija, and Wolfgang Nejdl. 2025. MoVoC: Morphology-aware subword construction for Ge'ez script languages. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 13131–13144, Suzhou, China. Association for Computational Linguistics.

Jonáš Vidra, Zdeněk Žabokrtský, Magda Ševčíková, and Lukáš Kyjánek. 2019. DeriNet 2.0: Towards an all-in-one word-formation resource. In *Proceedings of the Second International Workshop on Resources and Tools for Derivational Morphology*, pages 81–89, Prague, Czechia. Charles University, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics.

Elena Volodina, Yousuf Ali Mohammed, and Therese Lindström Tiedemann. 2021. CoDeRooMor: A new dataset for non-inflectional morphology studies of Swedish. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 178–189, Reykjavik, Iceland (Online). Linköping University Electronic Press, Sweden.

Anastasiya S. Volskaya, Tatyana A. Korneyeva, and Tatyana D. Markova. 2018. Word-formation, morphemic, etymological analysis in school. *Amazonia Investiga*, 7(15):190–195.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Ruibo Xie, Jie Zhang, Xinchun Wu, and Thi Phuong Nguyen. 2019. The relationship between morphological awareness and reading comprehension among chinese children. *Frontiers in Psychology*, Volume 10 - 2019.

Zdeněk Žabokrtský, Niyati Bafna, Jan Bodnár, Lukáš Kyjánek, Emil Svoboda, Magda Ševčíková, and Jonáš Vidra. 2022. Towards universal segmentations: UniSegments 1.0. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1137–1149, Marseille, France. European Language Resources Association.

A Risks

Our research is primarily fundamental, therefore we do not consider any risks to be significant. However, we acknowledge several ethical considerations.

First, there is the risk of performance disparity. Our results show that segmentation accuracy varies significantly across languages (from 64% to 95%). Deploying these models in downstream applications, such as search engines or educational tools, may provide a lower quality of service for speakers of low-resource or typologically complex languages, potentially exacerbating the digital divide.

Second, we highlight the risk of linguistic misinformation. In educational contexts, a model that incorrectly segments a word or mislabels a morpheme type may mislead students. Therefore, the potential integration into educational applications should be accompanied by a detailed study and classification of the errors made by the models.

Third, the environmental impact of this research should be noted. Fine-tuning large-scale multilingual models such as XLM-Roberta-large and Qwen3-32B across 20 languages with 5-fold cross-validation requires significant energy consumption.

Finally, our LLM experiments rely on proprietary APIs. This introduces a challenge for long-term reproducibility, as the underlying models are frequently updated or deprecated by their providers. Furthermore, users applying our methodology to sensitive or proprietary text must be aware of the privacy implications of transmitting data to third-party API providers.

B Scientific Artifacts

Our work uses external datasets, pre-trained models and software libraries. All artifacts we used were used within the terms of their licensing. We also have prepared a repository containing our methods, cross-validation results and prompts[8].

B.1 Datasets

The datasets we utilized do not contain any personal information. The datasets may contain a small number of examples of obscene and offensive vocabulary. However, all the data used are based on academic dictionaries and projects, so we believe that such words are an integral part of the language, and morpheme segmentation models should be able to work with such words.

19 out of the 20 datasets used in the experiments are distributed under various free licenses. The exception was the dataset for the Russian language, provided by the Russian National Corpus (Savchuk et al., 2024) after signing the appropriate license agreement.

1. Datasets from Universal Segmentations[9] (Žabokrtský et al., 2022):
   (a) Armenian (Uniparser) (Khurshudyan et al., 2022)
   (b) Croatian (CroDeriV) (Šojat et al., 2014)
   (c) English (MorphoLex) (Sánchez-Gutiérrez et al., 2018)
   (d) French (MorphoLex) (Mailhot et al., 2020)
   (e) Hindi (KCIS) (Angle et al., 2018)
   (f) Italian (DerIvaTario) (Talamo et al., 2016)
   (g) Kannada (KCIS) (Atmakuri et al., 2018)
   (h) Malayalam (KCIS) (Jayan et al., 2017)
   (i) Marathi (KCIS) (Redkar et al., 2016)

[8] https://anonymous.4open.science/r/morpheme-segmentation-050C

[9] https://ufal.mff.cuni.cz/universal-segmentations

(j) Persian (PerSegLex) (Ansari et al., 2019)
(k) Tajik (Uniparser) (Iskandarova, 2021)
(l) Udmurt (Uniparser) (Arkhangelskiy and Medvedeva, 2016)

2. Nguni datasets collected by Gaustad and Puttkammer (2022)[10] an transformed to surface segmentation dataset by Moeng et al. (2021)[11]

3. **Slounik** — Belarusian[12] (Morozov et al., 2025)

4. **DeriNet** — Czech[13] (Vidra et al., 2019)

5. **Morphodict-K** — Russian (Morozov et al., 2024)

6. **CoDeRooMor** — Swedish[14] (Volodina et al., 2021)

B.2 Pre-trained Models

We utilized six pre-trained models from HuggingFace[15]:

1. `XLM-Roberta-large`[16] (Conneau et al., 2020). This model was trained on 100 languages, has 550M parameters, and is available under MIT license.

2. `bert-base-multilingual-cased`[17] (Devlin et al., 2019). This model was trained on 104 languages, has 177M parameters, and is available under Apache 2.0 license.

3. `distilbert-base-multilingual-cased`[18] (Sanh et al., 2020). This model was trained on 104 languages, has 134M parameters, and is available under Apache 2.0 license.

4. `Qwen2.5-32B-Instruct`[19] (Team, 2024). This instruction-tuned causal language model has 32.5B parameters, supports over 29 languages, and is available under Apache 2.0 license.

5. `Qwen3-4B`[20] (Team, 2025). This causal language model was released as part of the Qwen3 family, supports over 100 languages and dialects, has 4.0B parameters, and is available under Apache 2.0 license.

6. `Qwen3-32B`[21] (Team, 2025). This causal language model was released as part of the Qwen3 family, supports over 100 languages and dialects, has 32.8B parameters, and is available under Apache 2.0 license.

We also used three proprietary generative LLMs: Gemini 3 Pro Preview by Google, GPT 5.1 by OpenAI, and Claude Sonnet 4.5 by Anthropic. Their parameters were not publicly disclosed. Access to the LLMs was provided via the OpenRouter API.

B.3 Software libraries

We used PyTorch (Paszke et al., 2019) to implement CNN and LSTM models, and PyTorch and Transformers (Wolf et al., 2020) for fine-tuning BERT-like models, as well as for executing the WordPiece and ULM baselines. For LoRA-tuning we used also PEFT[22] and TRL[23] libraries. We used the official implementation of Morfessor2.0 available on GitHub[24] for both semi-supervised training and inference. The LLMs were accessed through the OpenRouter API; the script for making these calls was implemented in Python using the `requests` library.

C Computational Experiments

An NVIDIA RTX 4090 (24GB) GPU was used to train the CNN and LSTM models. In total, training all models of these types took less than 10 hours of GPU time.

An NVIDIA A100 (40GB) GPU was used for fine-tuning BERT-like models for Armenian, Czech, Tajik, and Udmurt languages, and an NVIDIA RTX 4090 (24GB) GPU for the remaining languages. Training time varied depending on the dataset size, model, and GPU. In total, fine-tuning

[10] https://repo.sadilar.org/items/eb524a86-8953-4d2b-89d9-f40f5860e36c
[11] https://github.com/TumiMoeng/MORPH_SEGMENT
[12] https://huggingface.co/datasets/ruscorpora/morphodict-bel
[13] https://ufal.mff.cuni.cz/derinet
[14] https://spraakbanken.gu.se/en/resources/coderoomor
[15] https://huggingface.co/
[16] https://huggingface.co/FacebookAI/xlm-roberta-large
[17] https://huggingface.co/google-bert/bert-base-multilingual-cased
[18] https://huggingface.co/distilbert/distilbert-base-multilingual-cased
[19] https://huggingface.co/Qwen/Qwen2.5-32B-Instruct
[20] https://huggingface.co/Qwen/Qwen3-4B
[21] https://huggingface.co/Qwen/Qwen3-32B
[22] https://github.com/huggingface/peft
[23] https://github.com/huggingface/trl
[24] https://github.com/aalto-speech/morfessor

all models took approximately 400 hours on the A100 GPU and approximately 600 hours on the RTX 4090 GPU.

An NVIDIA A100 (80GB) GPU was used for LoRA of Qwen models. Training time varied depending on the dataset size, model, and GPU. In total, fine-tuning all models took approximately 4000 GPU hours.

Access to the LLMs was provided via the OpenRouter[25] REST API. In total, approximately 15,000 queries were executed. The total budget for the experiment was approximately 600 USD. For all models, we enabled the reasoning mode and selected the "high" setting for the reasoning budget. A temperature of 1.0 was used for all models. The average query size for the zero-shot experiment was 300 tokens, for the few-shot approach, 3,500 tokens, and for the two-stage approach, 1,200 tokens. The size of the generated text varied significantly due to the enabled reasoning and ranged from 100 to 15,000 tokens (see Table 6).

Language	Tokens	Language	Tokens
Armenian	2k	Marathi	2.2k
Belarusian	2.9k	Ndebele	3.7k
Croatian	2.1k	Persian	2.3k
Czech	4.3k	Russian	2k
English	1.8k	Swati	3.6k
French	2.4k	Swedish	2.3k
Hindi	1.2k	Tajik	2.6k
Italian	3.3k	Udmurt	1.6k
Kannada	3.2k	Xhosa	3.9k
Malayalam	2.8k	Zulu	3.8k

Table 6: Average model response length in tokens for each language in the dataset.

D Human Annotation

We used only pre-existing datasets, so human annotators were not involved in the research.

E AI Assistant

We used an AI assistant as a translator, proof-reader, during the preliminary analysis of datasets representing languages we do not speak. Some of the services used for searching scientific literature and datasets also rely on AI. We also used an AI assistant to refactor parts of the codebase published in the anonymized repository for improved readability. At the same time, we did not use AI assistants for originally coding or writing the original text of the article.

[25] https://openrouter.ai

F Preliminary Experiments and Extended Experimental Results

F.1 BERT-like Models

We followed the approach proposed by Morozov et al. (2025): a BERT-like model receives a character-by-character pre-tokenized word as input and classifies each character with BMES label. Since we were considering language-agnostic architectures, we focused only on multilingual BERT-like models. As such, we chose XLM-Roberta-large (Conneau et al., 2020) with 550M parameters, bert-base-multilingual-cased (Devlin et al., 2019) with 177M parameters (hereinafter mBERT), and distilbert-base-multilingual-cased (Sanh et al., 2020) with 134M parameters (hereinafter dBERT). Each of these models was trained on approximately one hundred languages, making them sufficiently universal and suitable for our purposes. Nevertheless, some of the languages we studied were absent from the models' training data. These languages include Udmurt and four Nguni languages (the exception is Xhosa, which was included in the XLM-Roberta-large training set). Testing on these languages allowed us to assess whether this approach would be competitive for low-resource languages that are not included in large multilingual datasets. Similarly to Morozov et al. (2025), we considered a modification of this approach where the entire word with the target label `'O'` is added to the beginning of the input sequence, while this word is ignored when calculating metrics (such models are designated by the suffix +word). Presumably, this modification allows the self-attention mechanism to attend to the global word context while classifying local characters. All models were trained for 30 epochs with a learning rate of 4×10^{-6}. For datasets smaller than 100,000 entries, we used a batch size of 16; for larger datasets (Armenian, Czech, Tajik, and Udmurt), due to GPU time limitations, a batch size of 256 was used to speed up training.

The results of the preliminary experiments are presented in Table 7. The best results for each language are highlighted in bold.

The XLMR+word model performed best for 11 of 20 languages in word-level accuracy and 8 of 20 languages in F-measure for morpheme boundaries. This allowed us to consider this model the best overall and include it in the main experiment.

Metric	F1	WA	F1	WA	F1	WA	F1	WA	F1	WA	F1	WA	F1	WA
Language	**Armenian**		**Belarusian**		**Croatian**		**Czech**		**English**		**French**		**Hindi**	
XLMR	**93.8**	83.5	98.5	90.9	**98.3**	92.5	99.2	93.5	**98.2**	94.9	88.5	70.6	97.5	95.1
XLMR+word	**93.8**	83.3	**98.8**	**92.4**	98.2	**92.7**	99.2	93.8	**98.2**	**95.3**	**89.8**	**74.5**	**98.4**	**97.1**
mBERT	93.7	83.9	98.5	91.4	**98.3**	92.5	**99.4**	**95.4**	98.1	94.7	88.5	70.5	97.6	95.3
mBERT+word	93.4	**84.6**	98.4	90.5	97.9	91.3	**99.4**	95.3	97.5	93.3	88.3	70.5	97.7	95.6
dBERT	93.5	83.3	98.4	90.7	98.2	92.1	**99.4**	95.3	98.0	94.5	88.3	69.9	97.4	95.1
dBERT+word	93.3	83.9	98.3	89.8	97.9	90.7	**99.4**	95.0	97.3	93.0	87.8	69.2	97.6	95.4
Language	**Italian**		**Kannada**		**Malayalam**		**Marathi**		**Ndebele**		**Persian**		**Russian**	
XLMR	91.7	76.2	**90.9**	**79.2**	**89.5**	**78.0**	91.9	80.2	**92.6**	66.3	94.9	83.2	**99.2**	93.6
XLMR+word	**92.5**	**78.5**	90.4	78.6	89.2	77.5	**92.0**	**80.6**	92.1	**64.6**	**95.8**	**86.1**	**99.2**	**94.3**
mBERT	92.2	77.7	89.4	76.1	88.0	75.4	91.3	78.6	92.4	66.3	94.9	83.6	99.1	93.3
mBERT+word	91.8	77.0	89.4	76.4	88.0	75.7	91.4	78.8	91.8	63.6	94.9	83.3	99.0	92.7
dBERT	92.0	76.9	89.3	75.8	87.9	75.3	91.3	78.5	92.1	**64.6**	94.9	83.2	99.0	93.0
dBERT+word	91.5	75.7	89.3	76.0	88.1	75.6	91.2	78.5	91.6	62.2	94.8	83.2	98.9	92.2
Language	**Swati**		**Swedish**		**Tajik**		**Udmurt**		**Xhosa**		**Zulu**			
XLMR	95.8	84.5	96.3	79.2	**94.9**	**87.2**	**95.2**	88.1	94.2	76.5	**94.9**	**77.8**		
XLMR+word	95.4	83.4	**97.4**	**84.5**	94.7	86.5	95.0	87.9	93.9	75.2	94.6	76.6		
mBERT	**96.0**	**85.2**	96.0	78.1	94.4	85.4	**95.2**	**88.5**	**94.3**	**77.0**	94.7	77.5		
mBERT+word	95.6	84.1	96.1	78.1	93.6	83.6	94.9	88.3	93.9	75.4	94.5	76.4		
dBERT	95.9	85.1	95.8	77.2	94.2	84.9	94.9	87.6	94.2	76.4	94.6	76.5		
dBERT+word	95.5	83.9	95.8	76.6	93.5	83.5	94.6	87.4	93.9	74.4	94.3	74.7		

Table 7: Results of BERT-like models preliminary experiments (average metric value)

However, the results of the best models in most cases differing very little. XLMR+word performed slightly worse than other approaches in Dravidian languages (Kannada and Malayalam), most Nguni languages (with the exception of Ndebele), languages from the Uniparser collection (Armenian, Tajik, and Udmurt), and Czech.

For the Czech language, mBERT results are close to those obtained by a monolingual fine-tuned model (Morozov et al., 2025). At the same time, for Belarusian and Russian, the XLMR+word results outperform similar monolingual approaches from the aforementioned work by 2-3 percentage points. Adding full words to the input sequences generally had a positive effect on the labeling quality for XLMR and a negative effect for mBERT and dBERT. However, adding the word reduced the quality of all models for almost all languages not included in the models' training data.

We were unable to establish a direct correlation between the size of different datasets and the quality of segmentation. To further investigate this question, we conducted an additional experiment with the XLMR+word model, testing it on reduced datasets (see Appendix F.2)).

We also repeated the BERT-like model experiments under the root-disjoint split scenario. In this setting, the advantage of XLMR+word was unequivocal: it achieved the best results for all 15 languages included in the experiment. The results also confirmed the effectiveness of providing the model with access to the full word, as the +word variant outperformed the corresponding baseline in the vast majority of cases.

F.2 Reducing Datasets

Table 9 reports the results of the XLMR+word model trained on reduced datasets. For each language, we trained the model on random subsets of 4K, 8K, and 30K training examples, and compared the results with the full-data setting. We performed the experiment both for the random split and for the root-disjoint split. This setup allows us to separate two factors that are otherwise conflated in a cross-lingual comparison: the amount of available supervision and the intrinsic difficulty of the dataset.

The results show that dataset size has a clear effect, but the effect is far from uniform across languages. In the random-split setting, increasing the training size usually improves performance: for example, Czech grows from 66.4 WA with 4K examples to 93.8 WA in the full-data setting, English from 77.2 to 95.3 WA, Russian from 80.0 to 94.3 WA, and Udmurt from 80.5 to 87.9 WA. Similar trends are observed for Belarusian, Croatian, French, Italian, Swedish, Xhosa, Zulu, and Ndebele. These results confirm that larger datasets are generally beneficial. This is consistent with results obtained by Olbrich and Žabokrtský (2026).

At the same time, the reduced-data experiment shows that dataset size alone does not explain cross-lingual differences in segmentation quality. Even with the same number of training examples, the

Metric	F1	WA	F1	WA	F1	WA	F1	WA	F1	WA	F1	WA	F1	WA	F1	WA
Language	**Armenian**		**Belarusian**		**Croatian**		**Czech**		**English**		**French**		**Hindi**		**Italian**	
XLMR	88.0	73.4	95.1	72.6	95.1	80.3	96.3	70.3	90.9	78.1	84.8	63.0	96.1	92.7	87.6	68.2
XLMR+word	**88.8**	**75.4**	**96.2**	**77.8**	**95.9**	**83.7**	**96.6**	**72.7**	**92.7**	**82.6**	**86.8**	**68.4**	**97.5**	**96.1**	**88.6**	**71.2**
mBERT	86.7	70.5	94.8	71.6	95.1	80.4	96.3	70.8	90.2	76.8	84.7	62.5	96.0	92.6	87.2	67.5
mBERT+word	87.4	72.3	94.9	71.7	95.3	81.2	96.4	71.4	90.2	77.0	85.2	64.1	96.6	94.0	87.3	68.2
dBERT	87.0	70.8	94.8	71.1	95.2	80.4	96.4	70.8	90.0	76.3	84.5	62.2	96.1	92.6	87.3	67.2
dBERT+word	87.7	72.7	94.9	71.6	95.3	81.1	96.4	71.1	89.8	75.9	84.7	63.0	96.2	93.2	86.8	66.4
Language	**Kannada**		**Malayalam**		**Marathi**		**Russian**		**Swedish**		**Tajik**		**Udmurt**			
XLMR	83.2	67.2	85.7	71.2	85.0	66.4	95.8	74.7	93.3	70.9	**90.7**	77.9	86.0	70.9		
XLMR+word	**83.3**	**67.9**	**85.8**	**71.9**	**87.2**	**71.4**	**96.7**	**79.9**	**94.4**	**75.9**	**90.7**	**78.0**	**86.8**	**72.7**		
mBERT	81.7	64.2	83.6	67.9	83.7	63.5	95.2	72.3	93.2	71.0	89.0	73.6	84.7	68.3		
mBERT+word	82.5	66.0	84.7	70.0	84.9	66.2	95.7	74.4	93.7	72.1	89.4	74.4	86.1	71.3		
dBERT	82.0	64.6	84.1	68.6	84.2	64.8	95.4	72.5	93.1	70.3	89.5	74.7	85.5	69.5		
dBERT+word	82.7	66.0	85.0	70.2	85.0	66.4	95.6	73.8	93.0	70.8	89.4	74.6	85.5	69.5		

Table 8: Results for root/stem-based splitting (average metric value)

	Armenian				**Czech**				**English**				**Malayalam**			
	4k	8k	30k	full*	4k	8k	30k	full*	4k	8k	30k	full	4k	8k	30k	full
random split																
F1	92.0	92.4	93.0	93.8	95.5	96.2	97.5	99.2	90.6	92.7	96.8	98.2	88.5	88.7	89.2	89.2
WA	81.6	83.1	84.5	83.3	66.4	71.2	80.3	93.8	77.2	82.1	91.9	95.3	76.6	77.0	77.6	77.5
split by roots																
F1	91.3	91.2	90.2	88.8	95.0	95.4	96.0	96.6	89.0	90.2	92.0	92.7	87.8	87.4	85.7	85.8
WA	80.3	80.4	78.8	75.4	62.9	65.7	69.2	72.7	73.5	76.5	81.0	82.6	75.3	74.7	71.8	71.9

	Marathi				**Persian**				**Russian**				**Tajik**			
	4k	8k	30k	full	4k	8k	30k	full	4k	8k	30k	full	4k	8k	30k	full*
random split																
F1	91.1	92.3	93.3	92.0	92.7	93.9	95.6	95.8	97.0	97.7	98.9	99.2	91.1	92.5	93.6	94.7
WA	78.1	80.0	82.0	80.6	76.6	79.9	85.4	86.1	80.0	84.6	91.7	94.3	77.8	81.3	84.2	86.5
split by roots																
F1	89.9	90.1	88.7	87.2	-	-	-	-	95.9	96.3	96.7	96.7	90.3	90.7	90.5	90.7
WA	75.9	76.1	72.8	71.4	-	-	-	-	75.0	77.0	79.4	79.9	76.1	77.6	77.5	78.0

	Udmurt				**Belarusian**			**Croatian**			**French**			**Italian**		
	4k	8k	30k	full*	4k	8k	full	4k	8k	full	4k	8k	full	4k	8k	full
random split																
F1	90.9	92.5	94.2	95.0	96.1	97.0	98.8	96.9	97.7	98.2	85.8	88.1	89.8	89.5	92.0	92.5
WA	80.5	84.0	87.8	87.9	77.2	82.4	92.4	87.2	90.5	92.7	65.9	70.8	74.5	71.5	77.4	78.5
split by roots																
F1	88.1	88.2	88.2	86.8	95.3	95.5	96.2	95.4	95.8	95.9	84.6	85.9	86.8	87.3	88.6	88.6
WA	75.0	75.8	75.9	72.7	73.2	74.7	77.8	81.3	83.0	83.7	63.5	66.5	68.4	67.8	71.1	71.2

	Kannada			**Swedish**			**Xhosa**			**Zulu**			**Hindi**		**Ndebele**	
	4k	8k	full	4k	8k	full	4k	8k	full	4k	8k	full	4k	full	4k	full
random split																
F1	88.5	89.6	90.4	96.0	97.1	97.4	92.5	93.7	93.9	93.3	94.3	94.6	98.4	98.4	90.9	92.1
WA	75.5	77.4	78.6	78.5	83.3	84.5	68.6	74.2	75.2	70.5	75.5	76.6	97.1	97.1	59.6	64.6
split by roots																
F1	86.0	85.3	83.3	94.2	-	94.4	-	-	-	-	-	-	97.7	97.5	-	-
WA	71.8	71.6	67.9	75.0	-	75.9	-	-	-	-	-	-	96.3	96.1	-	-

Table 9: Results of the XLMR+word model trained on reduced datasets. The columns indicate the number of training examples used in each setting. The full* setting denotes datasets for which batch size of 256 was utilized on the full training set.

variation in Word Accuracy remains substantial: with 4K training examples, the scores range from 59% to 87% if Hindi is excluded, and with 8K examples, from 71% to 91%. This suggests that training set size is not the only factor affecting segmentation quality; other factors, potentially including annotation depth, morpheme inventory, and dataset-specific segmentation principles, may also play an important role.

A second important observation is that XLMR+word shows signs of overfitting on several languages. This is visible in the decrease

in performance when moving from reduced to full training data, for example for Armenian, Udmurt, Marathi, Malayalam, and Kannada in the root-disjoint setting. This may partially explain why the XLMR+word approach is outperformed by the Qwen3-32B model in the main experiment for some language groups.

Finally, reducing the training set size has a much stronger effect in the random-split setting than in the root-disjoint setting. In the random split, the gain from using the full dataset instead of 4K examples often reaches 10–15 percentage points in Word Accuracy. In the root-disjoint split, however, the corresponding difference is typically much smaller, around 5–10 percentage points. This suggests that simply adding more annotated examples may be insufficient to fully address OOV-root generalization. Instead, it likely requires new, more linguistically informed modeling approaches.

Overall, the reduced-data experiments support two conclusions. First, more training data generally improves segmentation quality, but the rate of improvement differs substantially across languages. Second, cross-lingual variation is not explained by dataset size alone. Other factors, such as segmentation paradigm, annotation granularity, morpheme frequency, and the prevalence of unseen roots, are likely to contribute and should be controlled for in future comparisons.

F.3 LLM Models (LoRA approach)

Table 10 reports the comparison of three LoRA-tuned Qwen models in the random-split setting. The results clearly favor Qwen3-32B, which achieves the highest or tied-highest scores for nearly all languages and both metrics. The difference is particularly visible in languages where the task is difficult for decoder models, such as French, Italian, Nguni languages, and Swedish, as well as in Belarusian and Croatian. Qwen3-4B often matches or slightly improves over Qwen2.5-32B on Dravidian and Indo-Aryan languages, indicating that the Qwen3 generation is beneficial even at smaller scale; however, it is less stable overall. Since Qwen3-32B combines the advantages of the newer model generation with substantially larger capacity and yields the most consistent results across languages, we use it as the representative LoRA-tuned decoder-only model in the main experiments.

F.4 LLM Models (Prompting Approach)

For the preliminary experiments, we used three state-of-the-art LLMs released in the fall of 2025: Gemini 3 Pro Preview by Google, GPT 5.1 by OpenAI, and Claude Sonnet 4.5 by Anthropic. We conducted preliminary experiments with three prompting strategies: zero-shot, few-shot, and two-stage generation. In the zero-shot approach, the prompt (provided in Appendix G.2) explicitly specified the language of the task and the set of morpheme types found in the dataset. The few-shot approach used the same prompt, followed by 50 random segmentations from the dataset.

In the first step of the Corpus-to-Prompt Distillation (C2PD) approach, the LLM generated an enhanced prompt based on the dataset. To do this, the Gemini 3 Pro Preview model was given the task to construct efficient prompt based on the dataset, the original zero-shot prompt as input, along with a excerpt from the dataset containing 10,000 segmentations (or less due to the dataset overall size). The generated prompt without manual changes was then used in the second step for the generation of segmentations. We did not limit the model in the number of examples mentioned in the generated prompt. As a result, the number of examples for different languages ranged from 3 to 5. The exception without segmentation examples in the generated prompt was Xhosa. An example of a generated prompt (for Swedish) is provided in Appendix G.3; prompts for each of the languages are available in the repository[26].

We limited the test datasets for the preliminary experiments to one hundred random words for each language. The results of the preliminary experiments are presented in Table 11. The best results for each language are highlighted in bold.

Few-shot prompting of Gemini 3 Pro Preview outperformed the other combinations in most cases. Zero-shot prompting, as expected, showed very unstable results, which we attribute to the ambiguity of the concept of morpheme segmentation. The C2PD algorithm almost always underperformed the few-shot approach, but on average required four times fewer input tokens, making it potentially competitive in budget-conscious settings.

[26] https://anonymous.4open.science/r/morpheme-segmentation-050C

Metric	F1	WA	F1	WA	F1	WA	F1	WA	F1	WA	F1	WA	F1	WA
Language	**Armenian**		**Belarusian**		**Croatian**		**Czech**		**English**		**French**		**Hindi**	
Qwen2.5-32B	93.7	86.7	98.0	89.4	96.5	87.4	99.1	93.1	97.6	94.6	86.8	71.7	97.7	96.7
Qwen3-4B	93.8	87.0	97.6	87.7	96.0	85.8	99.1	93.1	96.8	92.9	85.4	68.3	97.6	96.5
Qwen3-32B	**93.9**	**87.2**	**98.3**	**90.8**	**97.1**	**89.1**	**99.3**	**94.3**	**97.8**	**94.9**	**88.1**	**73.6**	**97.9**	**97.1**
Language	**Italian**		**Kannada**		**Malayalam**		**Marathi**		**Ndebele**		**Persian**		**Russian**	
Qwen2.5-32B	87.1	69.8	90.1	79.3	88.2	79.5	93.9	82.8	90.0	58.0	95.0	84.1	99.0	93.3
Qwen3-4B	85.7	66.6	90.3	80.1	88.4	80.0	94.3	84.1	89.4	57.0	94.8	83.5	98.8	92.4
Qwen3-32B	**88.7**	**72.9**	**90.8**	**81.0**	**88.5**	**80.3**	**94.6**	**85.0**	**90.5**	**60.4**	**95.5**	**85.7**	**99.1**	**93.7**
Language	**Swati**		**Swedish**		**Tajik**		**Udmurt**		**Xhosa**		**Zulu**			
Qwen2.5-32B	90.5	73.3	94.8	78.1	**94.6**	**87.3**	**95.0**	**89.8**	91.6	68.0	92.6	70.0		
Qwen3-4B	89.9	71.8	93.3	74.5	**94.6**	87.1	94.9	89.7	91.0	66.1	92.0	68.8		
Qwen3-32B	**92.7**	**77.9**	**95.4**	**80.9**	**94.6**	87.1	94.9	89.6	**92.3**	**70.7**	**93.3**	**73.1**		

Table 10: Results of LoRA-tuned Qwen models. Qwen2.5-32B stands for Qwen2.5-32B-Instruct.

Metric	F1	WA	F1	WA	F1	WA	F1	WA	F1	WA	F1	WA	F1	WA
Language	**Armenian**		**Belarusian**		**Croatian**		**Czech**		**English**		**French**		**Hindi**	
Gemini-zero	60.7	8	91.9	57	88.0	44	68.8	11	83.4	55	78.0	34	86.0	66
GPT-zero	66.4	12	90.7	45	87.6	48	70.5	15	85.2	59	80.0	41	79.7	51
Claude-zero	61.4	12	88.9	37	82.4	38	66.8	11	80.1	50	77.8	40	80.5	53
Gemini3+few	**95.0**	**91**	**94.5**	68	95.1	**83**	83.9	43	**93.3**	**83**	**84.2**	**58**	97.2	94
GPT-few	92.0	82	94.2	**71**	**95.4**	79	**85.0**	**48**	89.7	78	81.6	50	94.6	88
Claude-few	87.4	67	92.5	64	93.0	77	78.5	31	90.1	74	80.6	51	92.0	81
Gemini-C2PD	93.1	83	93.6	61	93.2	78	79.9	25	90.3	77	80.8	50	**98.6**	**97**
GPT-C2PD	91.2	83	93.7	67	94.9	79	82.0	38	90.0	78	79.9	46	**98.6**	**97**
Claude-C2PD	87.7	75	92.6	57	92.5	70	75.5	26	90.0	76	80.3	47	98.1	96
Language	**Italian**		**Kannada**		**Malayalam**		**Marathi**		**Ndebele**		**Persian**		**Russian**	
Gemini-zero	70.2	24	62.2	37	70.0	35	77.5	44	84.6	36	92.5	73	93.9	58
GPT-zero	75.9	36	72.0	47	62.1	20	74.2	34	82.2	26	92.9	74	91.1	63
Claude-zero	71.9	29	62.9	28	59.0	23	72.4	29	62.7	12	88.2	58	89.9	56
Gemini3+few	79.8	51	**79.9**	**65**	82.0	72	**90.3**	**75**	**89.0**	**52**	92.5	**78**	**96.0**	**77**
GPT-few	**81.3**	**55**	78.0	60	80.3	70	84.4	66	86.0	37	**93.5**	77	92.8	68
Claude-few	75.5	40	75.7	55	79.6	66	80.0	52	86.2	41	91.2	68	92.9	70
Gemini-C2PD	75.2	36	72.0	47	**82.8**	**73**	87.3	69	87.3	43	92.7	72	95.6	75
GPT-C2PD	75.3	39	73.7	50	74.5	62	83.9	64	87.7	48	92.0	71	92.6	65
Claude-C2PD	71.6	32	66.3	43	73.6	59	81.4	58	83.1	36	89.9	64	91.7	61
Language	**Swati**		**Swedish**		**Tajik**		**Udmurt**		**Xhosa**		**Zulu**			
Gemini-zero	77.5	41	89.2	55	74.9	27	79.2	51	84.4	33	86.0	45		
GPT-zero	74.5	39	88.4	54	76.6	31	76.0	40	85.6	36	84.5	39		
Claude-zero	68.4	34	87.3	56	74.9	27	70.9	32	80.2	25	82.0	30		
Gemini3+few	93.7	76	**95.6**	**82**	**93.9**	**85**	**93.2**	**85**	**91.4**	**64**	**90.1**	55		
GPT-few	88.6	65	91.6	66	91.4	76	92.1	82	88.6	47	88.2	**56**		
Claude-few	87.5	57	89.0	63	88.3	69	90.1	79	85.4	39	86.2	40		
Gemini-C2PD	**94.1**	**79**	91.7	63	88.8	74	93.0	84	87.0	42	85.4	40		
GPT-C2PD	91.0	69	91.3	65	89.7	75	90.3	80	86.6	39	85.9	45		
Claude-C2PD	83.6	54	90.4	62	87.0	67	81.2	59	85.5	39	86.5	43		

Table 11: Results of LLM preliminary experiments

F.5 Errors Analysis

This appendix provides additional details for the error analysis summarized in Section 5. We focus on XLMR+word and Qwen3-32B, since they are the two strongest models and represent different modeling paradigms: discriminative character-level sequence labeling and autoregressive generation. The goal of this analysis is not only to count errors, but also to determine whether the remaining mistakes are model-specific, language-specific, or caused by dataset annotation conventions.

We classify each incorrect prediction using the taxonomy in Table 12. At the coarse level, we distinguish boundary-level operations: adding extra boundaries, omitting gold boundaries, shifting a boundary by a small distance, combining several boundary errors, or assigning wrong morpheme types. At the fine-grained level, we further identify the main linguistic configuration involved in the error, such as root over-splitting, missed root–suffix boundaries, suffix merging, prefix ambiguity, or Nguni-specific prefix-chain errors.

Table 13 shows that the two models often fail on the same words, but the degree of overlap differs substantially across groups. The highest over-

Coarse type	Fine-grained examples	Interpretation
Over-segmentation	root over-split; suffix over-split; false initial prefix	The model inserts one or more extra boundaries.
Under-segmentation	missed root–suffix boundary; suffixes merged; prefix absorbed by root	The model omits one or more gold boundaries.
Boundary shift	root–suffix shift; prefix–root shift; suffix–suffix shift	The model places a boundary close to the gold boundary.
Mixed error	mixed boundary errors	Extra, missing, and shifted boundaries occur in the same word.
Type-only	type-only	Boundaries are correct, but at least one morpheme type is wrong.
Nguni-specific	Nguni prefix-chain over/under-split; long segment over-split	Residual errors refined using language-specific heuristics.

Table 12: Error taxonomy used for classifying incorrect segmentations.

Group	Jaccard	Shared/XLMR	Shared/Qwen
Croatian	0.29	56.7	38.0
Czech	0.37	51.9	56.7
Dravidian	0.58	69.2	78.4
East Slavic	0.36	57.2	49.2
Indo-Aryan	0.45	58.5	66.8
Italian	0.36	59.9	47.4
MorphoLex	0.31	48.1	45.2
Nguni	0.47	70.1	59.1
Persian	0.42	59.6	58.1
Swedish	0.39	63.1	50.8
UniParser	0.52	61.3	76.4

Table 13: Overlap between incorrectly segmented words for XLMR+word and Qwen3-32B. Jaccard is computed over the sets of erroneous word forms. Shared/XLMR and Shared/Qwen indicate the percentage of each model's errors that are also errors of the other model.

lap is observed for Dravidian and UniParser-based datasets: the Jaccard similarity reaches 0.58 and 0.52, respectively, and more than three quarters of Qwen3-32B errors in these groups are also errors of XLMR+word. This suggests that many errors in these datasets may be caused by intrinsically difficult word forms or annotation decisions rather than by a single architecture. By contrast, the overlap is lower for MorphoLex, Croatian, Italian, and East Slavic datasets, indicating a larger architecture-specific component. For example, in Croatian, only 38.0% of Qwen3-32B errors are shared with XLMR+word, whereas 56.7% of XLMR+word errors are shared with Qwen3-32B.

A more detailed view of the dominant fine-grained clusters is given in Table 14. The largest common pattern is disagreement about segmentation granularity. In East Slavic, Indo-Aryan, MorphoLex, Czech, Italian, and Persian, many errors involve either splitting morphemes too finely or failing to separate a root–suffix boundary. These errors are particularly prominent in Czech, where root over-splitting accounts for 37.5% of XLMR+word errors, while missed root–suffix boundaries account for 32.0% of Qwen3-32B errors.

UniParser-based datasets have a different profile. Since these resources annotate inflectional morphology only, the dominant error is not deep derivational decomposition but the exact placement of the stem–suffix boundary. This is especially clear for Qwen3-32B, for which boundary shifts account for 58.1% of all errors in this group. The Dravidian datasets also contain many boundary shifts, but they are accompanied by complex errors combining shifts with over- or under-segmentation. This makes them less localized than the UniParser errors.

Nguni and Swedish show the clearest resource-specific patterns. The surface segmentation datasets used in our experiments do not include morpheme type labels for the Nguni languages. However, the original annotation files do contain this information, and we used it as auxiliary evidence when interpreting the error clusters. This shows that, in Nguni languages, both models mainly make prefix-chain errors: they either split the initial sequence of short pre-root morphemes too finely or merge several such elements. Swedish, in contrast, is dominated by type-only errors. These errors preserve morpheme boundaries but disagree with the gold morpheme labels, often for one-letter elements such as *-a-* and *-e-*, which can be annotated either as linking elements or as suffixes depending on the word.

To test whether boundary errors are arbitrary or linguistically plausible, we additionally checked whether morphemes produced by over-segmentation, under-segmentation, and boundary-shift errors are attested elsewhere in the corresponding gold data. Table 15 reports weighted averages over languages, using the number of relevant errors as weights. For over-segmentation and boundary shifts, at least one produced morpheme is attested in more than 99% of cases for both models. Under

Group	XLMR+word	Qwen3-32B
East Slavic	root over-split (19.5); suffix over-split (15.1)	root over-split (16.6); suffix over-split (13.4)
Indo-Aryan	root over-split (34.2); missed root–suffix (24.5)	root over-split (26.6); missed root–suffix (23.7)
MorphoLex	root over-split (30.1); root–suffix shift (23.6)	root–suffix shift (25.5); missed root–suffix (22.8)
Czech	root over-split (37.5); missed root–suffix (22.1)	missed root–suffix (32.0); root over-split (29.6)
Croatian	root–suffix shift (37.3); false initial prefix (20.7)	root–suffix shift (33.9); prefix–root shift (18.1)
Italian	root-prefix merge (20.3); false initial prefix (19.8)	false initial prefix (20.6); root–suffix shift (19.7)
Persian*	over-split (45.4); under-split (41.0)	under-split (44.6); over-split (39.6)
Dravidian	root–suffix shift (25.3); mixed errors (25.1)	mixed errors (28.3); root–suffix shift (27.1)
Nguni	prefix-chain over-split (31.8); prefix-chain under-split (20.8)	prefix-chain over-split (29.3); prefix-chain under-split (22.7)
UniParser	stem–suffix shift (42.5); stem–suffix over-split (19.2)	stem–suffix shift (58.1); mixed errors (19.8)
Swedish	type-only (39.4); root over-split (18.2)	type-only (27.0); root over-split (16.1)

Table 14: Dominant fine-grained error clusters by language group. Values in parentheses indicate the percentage of all errors for the corresponding model and group. For the Persian language the morpheme types are not presented in the dataset.

Model	Error type	Inst.	Any	All
Qwen3-32B	over-seg.	51,273	99.4	79.6
Qwen3-32B	under-seg.	46,354	52.5	52.5
Qwen3-32B	boundary shift	89,640	99.5	92.1
Qwen3-32B	all	187,267	87.9	78.9
XLMR+word	over-seg.	96,787	99.6	55.2
XLMR+word	under-seg.	54,088	48.4	48.4
XLMR+word	boundary shift	72,974	99.7	91.2
XLMR+word	all	223,849	87.3	65.3

Table 15: Attestation of morphemes produced by boundary errors. Inst. is the number of error instances. Any indicates the percentage of cases where at least one produced morpheme is attested elsewhere in the language-specific gold data; All requires all produced morphemes to be attested. Values are weighted averages over languages.

a stricter criterion, requiring all produced morphemes to be attested, the effect remains very strong for boundary shifts: 92.1% for Qwen3-32B and 91.2% for XLMR+word. Over-segmentation also often produces attested morphemes, especially for Qwen3-32B, where all resulting morphemes are attested in 79.6% of cases.

Under-segmentation behaves differently. Since the model merges several gold morphemes into a larger segment, the resulting segment is much less likely to be independently attested: all produced segments are attested in 52.5% of Qwen3-32B under-segmentation errors and 48.4% of XLMR+word errors. This contrast suggests that over-segmentation and boundary-shift errors often correspond to plausible alternative analyses supported by the dataset inventory, whereas under-segmentation more often creates larger units that do not function as independent morphemes elsewhere in the data.

Table 16 shows that this pattern is not uniform across datasets. In several languages, over-segmentation is highly inventory-supported. Russian, Czech, Tajik, Udmurt, Persian, Kannada, Marathi, and the Nguni languages all show very high Any values, usually above 98%, and high All values for over-segmentation. Such errors are unlikely to be arbitrary character-level noise. They more plausibly reflect disagreement about the granularity of segmentation.

The strongest evidence of this type comes from derivationally rich datasets. In Russian and Belarusian, over-segmentation often produces short but frequent elements such as *н*, *к*, *а*, *о*, and *ов*. These elements are plausible derivational or inflectional material, but their status may differ across words: in some cases they are annotated separately, while in others they may be absorbed into a larger root or suffix. Under-segmentation in the same languages also produces larger derivational complexes such as Russian *енн*, *ени*, *ник*, *тельн*, and Belarusian *нік*. Czech also provides a clear example of segmentation granularity. Over-segmentation produces frequent attested fragments such as *k*, *n*, *e*, *sk*, and *ov*, while under-segmentation often yields larger attested units such as *ek*, *ovsk*, *ick*, *ník*, *ák*, *ář*, and *ík*. In such cases, the error may reflect irregularities in gold annotation or a mismatch with the annotation convention rather than a linguistically impossible prediction.

The English and French MorphoLex-datasets behave differently. Over-segmentation and boundary-shift errors often produce real suffixes such as English *s*, *ed*, *es*, *er*, and *y*, or French *ement*, *ment*, *ance*, *ion*, *a*, *e*, and *s*. However, under-segmentation almost never produces attested morphemes: only 6.1% of English and 2.6% of French under-segmentation errors are attested under the All criterion. The merged strings in these errors are typ-

	Over-seg.			Boundary shift			Under-seg.	
Language	Inst.	Any	All	Inst.	Any	All	Inst.	Any/All
Russian	3,536	99.7	85.9	730	99.2	69.9	3,106	63.8
Belarusian	2,576	99.1	72.6	674	96.9	51.2	2,020	46.2
Czech	50,701	99.8	80.3	2,489	99.1	71.9	40,886	57.6
English	2,091	96.8	44.5	1,857	97.5	44.4	1,784	6.1
French	2,656	97.4	42.6	1,983	94.7	33.6	2,717	2.6
Italian	1,684	98.6	52.9	1,114	88.7	29.9	1,408	16.9
Croatian	575	98.6	43.1	1,451	97.9	36.2	371	15.9
Swedish	1,072	97.8	66.1	531	91.1	35.6	1,025	29.5
Tajik	11,751	99.9	87.2	12,917	99.7	92.5	8,296	71.7
Udmurt	6,691	99.5	85.4	36,367	99.8	95.7	5,360	62.1
Armenian	39,333	99.8	24.6	89,182	100.0	98.3	10,790	30.3
Ndebele	2,674	99.1	77.2	574	99.3	66.7	2,077	63.8
Swati	890	96.4	48.5	119	96.6	46.2	714	22.0
Xhosa	2,345	98.9	70.6	834	98.9	61.9	1,666	45.9
Zulu	2,002	99.3	72.6	787	99.4	61.9	1,747	63.3
Kannada	2,495	99.2	78.3	4,324	99.0	79.6	3,167	63.6
Malayalam	3,030	98.0	69.7	3,531	98.0	68.3	3,701	31.7
Hindi	154	97.4	69.5	111	97.3	64.9	132	44.7
Marathi	6,396	99.5	77.9	1,889	98.6	66.4	4,020	49.9
Persian	5,408	99.4	79.7	1,150	98.7	77.3	5,455	38.1

Table 16: Language-level attestation of morphemes produced by boundary errors. Counts are aggregated over XLMR+word and Qwen3-32B. Any and All are percentages of error instances.

ically whole-word-like or stem-like chunks rather than morphemes, for example English fragments such as *megalith*, *boilersuit*, and *pussycat*, or French fragments such as *atlantique*, *éperdu*, and *visse*. A similar picture is observed in the cases of Italian, Croatian, and Swedish. In all three datasets, over-segmentation often produces attested short elements, but under-segmentation is rarely supported by the inventory: only 16.9% of Italian, 15.9% of Croatian, and 29.5% of Swedish merged segments are attested. Frequent unattested under-segmentation fragments include Italian *quadra*, *mura*, *digest*, *solvere*, and Croatian *smuć*, *stumb*, *irava*, and *tra*. These cases are more likely to represent genuine missed boundaries rather than plausible alternatives. Thus, in these datasets, missed boundaries are much more likely to be genuine errors than plausible alternative segmentations.

The UniParser-based datasets show another pattern. For Tajik and Udmurt, both over-segmentation and boundary-shift errors are strongly supported by the inventory: the All value reaches 87.2% and 85.4% for over-segmentation, and 92.5% and 95.7% for boundary shifts, respectively. This is consistent with the annotation paradigm of these resources, which mostly mark inflectional morphology. The main ambiguity lies in the exact placement of the stem–suffix boundary, and both sides of a shifted boundary are often attested as suffixal or stem-final material elsewhere in the dataset. Armenian differs from Tajik and Udmurt: boundary shifts are almost always fully attested (98.3%), but over-segmentation has a much lower All value (24.6%). Frequent unattested pieces include վ, գն, գվ, եգն, եգդ, ացն, and ացդ. This suggests that models sometimes attempt a deeper internal decomposition of Armenian forms than the gold inventory represents.

The Nguni languages confirm the prefix-chain interpretation proposed in the main text. Frequent attested fragments produced by over-segmentation include *e*, *a*, *el*, *is*, *ku*, *m*, and *o*; under-segmentation often produces attested longer elements such as *aba*, *oku*, *ezi*, *kwe*, *eli*, and *eni*. These examples are consistent with errors involving prefixal chains. The pattern is not identical across the four Nguni datasets. Ndebele and Zulu have relatively high All values for under-segmentation (63.8% and 63.3%), whereas Swati is much lower (22.0%). This may reflect the smaller size of the Swati dataset, lower coverage of the morpheme inventory, or differences in how the surface segmentation adaptation represents prefixal material.

For Dravidian, Indo-Aryan, and Persian scripts that are difficult to render reliably with the conference LaTeX setup, we provide examples in Latin transliteration.

In the Dravidian and Indo-Aryan datasets, many error-induced morphemes are productive endings or post-root elements. Kannada over-segmentation frequently produces short fragments transliterated as *-i*, *-isa*, *-da*, *-a*, *-ra*, *-na*, and *-u*. Marathi shows

a similar pattern with fragments transliterated as *-a*, *-e*, *-am*, *-i*, *-ta*, *-cyā*, *-ce*, and *-madhye*. These fragments are common in the dataset inventory, so models often produce real morphemes even when the gold segmentation does not separate them at that point. Malayalam, however, requires special caution. Some frequent fragments are tied to the script and Unicode representation, such as the chandrakkala/virama sign and sequences containing zero-width joiners, alongside more morphologically interpretable fragments transliterated as *-il*, *-yil*, *-ukaḷil*, and *-ṅṅaḷ*. Therefore, high attestation in Malayalam does not always guarantee a linguistically meaningful morpheme; part of the effect may reflect orthographic or encoding-level segmentation.

Persian shows high attestation for over-segmentation and boundary shifts: 79.7% and 77.3% of such errors, respectively, are fully inventory-supported. The most frequent attested fragments include elements transliterated as *-i*, *-e/-h*, *-hā*, *-ān*, *-and*, *-šān*, *-gi*, and *-tarin*. These are frequent suffixal or word-final elements, and models often disagree with the gold standard about whether they should be separated. Since the Persian dataset lacks morpheme type labels, these cases cannot be further disambiguated into root, suffix, or linking-element errors, but they clearly point to segmentation granularity as the dominant source of disagreement.

Overall, observed high attestation of over-segmentation and boundary-shift errors provides strong evidence that many model errors are not random: they often consist of morphemes that the same dataset recognizes elsewhere. This is especially clear for Russian, Belarusian, Czech, Persian, Nguni, Tajik, Udmurt, Kannada, and Marathi. When the same short or productive elements repeatedly appear as error-induced morphemes, especially in derivationally rich datasets, this provides indirect evidence for competing annotation conventions or incomplete explicitness of the segmentation guidelines. In particular, inconsistent separation of historical suffixes from roots, ambiguous treatment of linking vowels, and variable granularity of suffix chains appear to be major sources of disagreement between models and gold annotations. Despite this, it should be noted that high attestation does not by itself prove annotation inconsistency. A morpheme may be valid in one word but not segmentable in another because of lexicalization, historical opacity, or language-specific morphophonology.

G Prompts

G.1 LoRA System Prompt

System prompt for LoRA

```
You are a linguistic expert. Segment the word
     into morphemes strictly following the
    format 'morpheme:TYPE/morpheme:TYPE'.
    Output ONLY the segmentation.
```

G.2 Zero-shot Prompt Example

Zero-shot prompt for Udmurt

```
You are an expert linguist specializing in
    morphological analysis for the Udmurt
    language. Your task is to perform surface-
    level morpheme segmentation of a given
    word.

Rules:
1. **Surface Segmentation**: The
    concatenation of the morpheme text parts
    MUST be exactly equal to the source word.
     Do not normalize, stem, or fix spelling.
2. **Tags**: Assign a tag to each morpheme
    from the following list: [PREF, ROOT,
    SUFF, CONNECTOR].
   - PREF: Prefix
   - STEM: Stem
   - SUFF: Suffix
   - CONNECTOR: Hyphen
3. **Output Format**: Return ONLY a valid
    JSON object containing a list of
    morphemes. Do not add explanations.
Expected JSON structure:
{
  "word": "source_word",
  "morphemes": [
    {"text": "part1", "type": "TAG"},
    {"text": "part2", "type": "TAG"}
  ]
}
```

G.3 Corpus-to-Prompt Distillation Prompt Example (for Swedish)

Corpus-to-Prompt Distillation Prompt for Swedish

```
You are an expert computational linguist
    specializing in the morphological
    analysis of the Swedish language. Your
    task is to perform a fine-grained,
    surface-level morpheme segmentation of a
    given source word based on etymological
    principles.

**Segmentation Principles & Granularity:**
1.  **Deep Etymological Decomposition**: You
    must break down words into their smallest
     historical constituents, particularly
    for words of Latin, Greek, or Romance
    origin. Do not stop at the stem; find the
```

```
     historical root.
    *   *Example*: `civilisation` -> `civ` (
    ROOT) + `il` (SUFF) + `is` (SUFF) + `at`
    (SUFF) + `ion` (SUFF).
    *   *Example*: `individualism` -> `in` (
    PREF) + `di` (PREF) + `vid` (ROOT) + `u`
    (LINK) + `al` (SUFF) + `ism` (SUFF).
    *   *Example*: `instrument` -> `in` (PREF
    ) + `stru` (ROOT) + `ment` (SUFF).
2.  **Germanic Derivation**: For native
    Swedish words, separate prefixes and
    suffixes from the root.
    *   *Example*: `förstånd` -> `för` (PREF)
     + `stå` (ROOT) + `nd` (SUFF).
    *   *Example*: `misstänka` -> `miss` (
    PREF) + `tänk` (ROOT) + `a` (LINK/SUFF).
3.  **Surface Constraint**: The concatenation
     of the segmented texts MUST match the
    input word exactly. Do not correct
    spelling or restore dropped characters (e.
    g., if a letter is doubled or dropped at
    a boundary, segment the surface form as
    it appears).

**Tagging Rules:**
Assign one of the following tags to each
    segment:
*   **PREF**: Prefixes (e.g., `be-`, `för-`,
    `o-`, `re-`, `kon-`, `in-`, `psyk-`, `
    eko-`).
*   **ROOT**: The core lexical morpheme. In
    loanwords, this is often a short bound
    root (e.g., `nom` in *ekonomisk*, `flikt`
     in *konflikt*).
*   **SUFF**: Derivational suffixes (e.g., `-
    ing`, `-het`, `-isk`, `-tion`) and verb
    endings (e.g., `-a`).
*   **LINK**: Use this tag for:
    *   Thematic vowels at the end of noun
    stems (e.g., the 'a' in `skara`, the 'e'
    in `bonde` or `hanne`).
    *   Linking elements in compounds or
    derivations (e.g., the 's' in `svensson`
    or `dygnsvis`, the 'o' in `psykolog`, the
     'u' in `individualism`).
    *   Note: While verb infinitive markers
    ('-a') are usually SUFF, noun endings ('-
    a', '-e') are strictly LINK in this
    schema.

**Few-Shot Examples:**
*   Input: "planering" -> Output: [{"text": "
    plan", "type": "ROOT"}, {"text": "er", "
    type": "SUFF"}, {"text": "ing", "type": "
    SUFF"}]
*   Input: "skara" -> Output: [{"text": "skar
    ", "type": "ROOT"}, {"text": "a", "type":
     "LINK"}]
*   Input: "psykolog" -> Output: [{"text": "
    psyk", "type": "PREF"}, {"text": "o", "
    type": "LINK"}, {"text": "log", "type": "
    ROOT"}]
*   Input: "konstatera" -> Output: [{"text":
    "kon", "type": "PREF"}, {"text": "stat",
    "type": "ROOT"}, {"text": "er", "type": "
    SUFF"}, {"text": "a", "type": "SUFF"}]
*   Input: "oekonomisk" -> Output: [{"text":
    "o", "type": "PREF"}, {"text": "eko", "
    type": "PREF"}, {"text": "nom", "type": "
    ROOT"}, {"text": "isk", "type": "SUFF"}]
**Output Format:**
Return ONLY a valid JSON object with the
    structure below. No commentary.
```json
{
  "word": "source_word",
  "morphemes": [
    {"text": "segment1", "type": "TAG"},
    {"text": "segment2", "type": "TAG"}
  ]
}
```
```