

Subword Segmental Language Modelling for Nguni Languages

Francois Meyer and Jan Buys
Department of Computer Science
University of Cape Town

MYRFRA008@myuct.ac.za, jbuys@cs.uct.ac.za

Abstract

Subwords have become the standard units of text in NLP, enabling efficient open-vocabulary models. With algorithms like byte-pair encoding (BPE), subword segmentation is viewed as a preprocessing step applied to the corpus before training. This can lead to sub-optimal segmentations for low-resource languages with complex morphologies. We propose a subword segmental language model (SSLM) that learns how to segment words while being trained for autoregressive language modelling. By unifying subword segmentation and language modelling, our model learns subwords that optimise LM performance. We train our model on the 4 Nguni languages of South Africa. These are low-resource agglutinative languages, so subword information is critical. As an LM, SSLM outperforms existing approaches such as BPE-based models on average across the 4 languages. Furthermore, it outperforms standard subword segmenters on unsupervised morphological segmentation. We also train our model as a word-level sequence model, resulting in an unsupervised morphological segmenter that outperforms existing methods by a large margin for all 4 languages. Our results show that learning subword segmentation is an effective alternative to existing subword segmenters, enabling the model to discover morpheme-like subwords that improve its LM capabilities.

1 Introduction

Subword segmentation has become a standard practice in Natural Language Processing (NLP). The dominant approach is to run an algorithm like BPE (Sennrich et al., 2016) as a preprocessing step, segmenting the corpus into subwords. This enables the model to learn features based on subwords, compose words, and handle rare and unknown words as an open-vocabulary model. Subword segmentation is an active area of research, since no single technique outperforms others across all tasks, languages, and dataset sizes (Zhu et al., 2019a,b). Be-

sesihambe	
Morphemes	se-si-hamb-e
BPE	sesi-ha-mbe
Unigram LM	se-si-hambe
Morfessor	se-s-ihambe
SSLM	se-si-hamb-e

Table 1: Segmentations of the isiXhosa word *sesihambe* produced by existing subword segmentation algorithms, compared to the actual morphemes and the output of our model (SSLM).

sides deterministic segmenters like BPE, stochastic algorithms like unigram LM (ULM) (Kudo, 2018) have also been proposed.

Subword segmentation is particularly important for the Nguni languages of South Africa (isiXhosa, isiZulu, isiNdebele, and Siswati) because they are agglutinative languages that are written conjunctively.¹ These are morphologically rich languages in which words are formed by stringing together morphemes (Tajard and Bosch, 2006). Morphemes are the primary linguistic units. For example, the isiXhosa word “sesihambe” means “we are gone”, where “se” means “we”, “si” means “are”, and “hamb-e” means “gone”, with the “-e” suffix indicating past tense. As shown in Table 1, existing segmenters do not reliably capture this.

The Nguni languages are under-resourced, which compounds the importance of subword segmentation. Available datasets are small, so any held-out dataset will contain rare or previously unseen words. Therefore it is critical for models to learn useful subword features and effectively model morphological composition. In a low-resource setting it may then be more effective to learn subword segmentation as part of model training rather than

¹The Sotho-Tswana languages of South Africa are also agglutinative, but are written disjunctively i.e. a single linguistic word may be written as several orthographic words.

as a distinct preprocessing step.

The probabilistic models underlying existing subword segmentation methods such as ULM and Morfessor (Creutz and Lagus, 2007) assume that subwords are context-independent, making them unsuitable for language modelling. In this paper we propose a subword segmental language model that simultaneously learns how to segment words while training as an autoregressive LM. This allows the model to learn subword segmentations that optimise a left-to-right language modelling objective, thereby being conditioned on the context. Our model learns the subwords that it can most effectively leverage for language modelling.

We train our model in the 4 Nguni languages of South Africa. We compile LM datasets for these languages from publicly available corpora and release our train/validation/test sets. On intrinsic language modelling performance averaged across the 4 languages our model outperforms neural LMs trained with characters, BPE, and ULM subwords. On the task of unsupervised morphological segmentation (which determines to what extent subwords correspond to actual morphemes) our model outperforms standard subword segmenters like BPE and ULM on average across the 4 languages.

In addition to these LMs, we train a second set of subword segmental models that train on single words in isolation (without having to model context for long-range language modelling). Our word-level models outperform all existing methods on unsupervised morphological segmentation (including segmenters like Morfessor) by a large margin across all 4 languages. Finally, we discuss the importance of a subword lexicon to our model, analysing how hyperparameters that control lexicon construction affect performance. In summary, this paper makes the following contributions:²

1. We propose a subword segmental language model (SSLM) that unifies subword segmentation and language modelling in a single end-to-end neural architecture.
2. We compile and release curated LM datasets for 4 Nguni languages.
3. We evaluate our model as an LM and an unsupervised morphological segmenter, and it outperforms existing methods on both tasks.

²Our code, trained models, and datasets are available at <https://github.com/francois-meyer/subword-segmental-lm>.

4. We present an analysis of how lexicon-related hyperparameters affect our model.

2 Subword Segmentation

In this section we review the paradigm that currently dominates subword segmentation, discuss its limitations, and introduce the family of models we draw inspiration from for our approach to subword segmentation — segmental sequence models.

2.1 Subword Segmentation Algorithms

Recently proposed subword segmentation algorithms start with some initial vocabulary (e.g. all characters) and iteratively amend it based on corpus subword statistics until a pre-specified vocabulary size has been reached. The goal of BPE (Sennrich et al., 2016) is to represent common characters sequences as distinct vocabulary items. ULM (Kudo, 2018) aims to maximise the likelihood of the training corpus under a unigram LM, in which subwords are generated independently.

These algorithms work well in certain contexts, but are not universally applicable. Klein and Tsarfaty (2020) show that they are sub-optimal for morphologically rich languages. Zhu et al. (2019b) show that the best method varies across languages and tasks, and existing segmenters require extensive tuning. They also find that subword segmentation is particularly beneficial for low-resource languages, but on average a simple character n-gram method outperforms BPE (Zhu et al., 2019a).

Recently it has become popular to construct shared multilingual vocabularies, but this leads to over-segmented words in low-resource languages (Wang et al., 2021b; Ács, 2019). Some have argued that these problems can be overcome by avoiding segmentation altogether (Clark et al., 2021) or by more sophisticated hyperparameter tuning (Salesky et al., 2020). But the limitations arise partly from the fact that the segmentation algorithms themselves are separated from model training. To overcome this we turn to a different paradigm, where we can cast subword segmentation as something for the model to learn.

2.2 Segmental Sequence Models

The main idea behind segmental sequence modelling is to let the model learn segmentation itself. This involves treating sequence segmentation as a latent variable to be marginalised over. The motivation behind this is that the model would be able

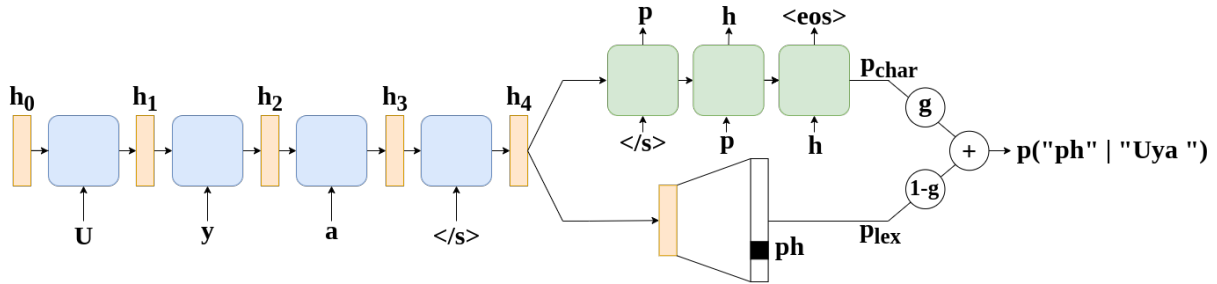


Figure 1: The SSLM computing the probability for the subword segment “ph” in the isiXhosa sentence “Uya phi?”. A character-level LSTM encodes the unsegmented text history “Uya ”, while a mixture model (equation 6) that interpolates between a separate character-level LSTM decoder and a lexicon model generates the segment “ph”. This is repeated for all possible subwords in a sequence to compute the forward scores of equation 3.

to “discover” the optimal segments for sequence prediction. These segments might correspond to natural underlying sequence units, such as words in text or phonemes in speech.

Variants of this idea have been used in a few neural sequence models. Kong et al. (2016) propose a bidirectional RNN that learns segment embeddings for handwriting recognition and POS tagging. Wang et al. (2017) propose SWAN (Sleep-Wake Networks), a segmental RNN for text segmentation and speech recognition. Both of these models use dynamic programming to efficiently compute marginal likelihood during training (by summing over all possible segmentations) and to find the most likely segmentation of a sequence.

Sun and Deng (2018) coined the term “segmental language model” (SLM) in applying this approach to Chinese language modelling for unsupervised word segmentation. Kawakami et al. (2019) extended their approach by equipping the model with a lexical memory and introducing segment length regularisation. Segmental models for word discovery have also been proposed as masked LMs (Downey et al., 2021) and bi-directional LMs (Wang et al., 2021a). Inspired by these works, we adapt segmental sequence modelling for the joint task of language modelling and subword discovery.

3 Subword Segmental Language Model

Our SSLM combines autoregressive language modelling and subword segmentation in a single model that can be trained end-to-end. The architecture is shown in Figure 1. It represents a radical divergence from segmenters like BPE and ULM, which view subword segmentation as context-independent. The SSLM views subword segmentation and language modelling jointly, so it can learn

subwords that optimise conditional LM generation.

3.1 Generative Model

The SSLM generates a sequence of space-separated words $\mathbf{w} = \mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n$, corresponding to an underlying sequence of characters \mathbf{x} , and generates each word \mathbf{w}_i as a sequence of subwords $\mathbf{s}_i = s_{i1}, s_{i2}, \dots, s_{i|\mathbf{s}_i|}$. The probability of a text sequence \mathbf{w} is computed through the marginal distribution over all possible word segmentations as

$$p(\mathbf{w}) = \sum_{\mathbf{s}: \pi(\mathbf{s})=\mathbf{w}} p(\mathbf{s}), \quad (1)$$

where $\pi(\mathbf{s})$ is the unsegmented text underlying the sequence of segmented words $\mathbf{s} = \mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n$. Using the chain rule, we define the probability of a sequence of segmented words as

$$p(\mathbf{s}) = \prod_{i=1}^{|\mathbf{w}|} \prod_{j=1}^{|\mathbf{s}_i|} p(s_{ij} | \mathbf{s}_{\leq i, < j}), \quad (2)$$

where $\mathbf{s}_{\leq i, < j}$ is the subword sequence preceding the j^{th} subword of the i^{th} word (this includes all subwords in the preceding words and the subwords preceding s_{ij} in the current word).

We treat white spaces and punctuation as assumed segments that are equivalent to 1-character words. In this way we implicitly model the end of a word. When the model predicts a non-alphabetical character (e.g. space) that is equivalent to a word boundary. Segments cannot cross word boundaries, so the only segmentation learned by the model is how to segment words into subwords.

3.2 Dynamic Programming Algorithm

Conditioning the probabilities of a segment $p(s_{ij} | \mathbf{s}_{\leq i, < j})$ on all possible segmentation histories

Model	Validation set BPC					Test set BPC				
	xh	zu	nr	ss	avg	xh	zu	nr	ss	avg
Char-LSTM	1.24	1.22	1.41	1.38	1.31	1.32	1.26	1.39	1.30	1.32
BPE-LSTM	1.23	1.19	1.39	1.38	1.30	1.30	1.22	1.39	1.28	1.30
ULM-LSTM	1.22	1.23	1.39	1.40	1.31	1.25	1.27	1.39	1.31	1.31
Char-Transformer	1.51	1.43	1.49	1.49	1.48	1.53	1.48	1.47	1.43	1.48
BPE-Transformer	1.30	1.22	1.38	1.38	1.32	1.33	1.27	1.36	1.30	1.31
ULM-Transformer	1.32	1.22	1.38	1.38	1.35	1.34	1.27	1.36	1.29	1.31
SSLM	1.24	1.19	1.35	1.38	1.29	1.27	1.21	1.35	1.28	1.28

Table 2: Intrinsic LM performance, as measured by BPC scores on the validation and test sets.

is computationally intractable, so we follow previous segmental sequence models by introducing a conditional semi-Markov assumption,

$$p(s_{ij} | \mathbf{s}_{\leq i, < j}) \approx p(s_{ij} | \pi(\mathbf{s}_{\leq i, < j})) \quad (3)$$

$$= p(s_{ij} | \mathbf{x}_{< k}), \quad (4)$$

where $\mathbf{x}_{< k}$ is the raw, unsegmented text preceding the current segment (assuming the current segment starts at the k^{th} character). Now the segment generation probability does not depend on the segmentations in the preceding sequence of words, or within the current word. Instead, the probability is conditioned on the unsegmented word and character history. This enables us to compute the marginal likelihood of equation 1 incrementally using a dynamic programming algorithm. Given $\alpha_0 = 1$, at each step the algorithm computes a forward score,

$$\alpha_t = \sum_{k=f(\mathbf{x}, t)}^t \alpha_k p(s = \mathbf{x}_{k:t} | \mathbf{x}_{< k}), \quad (5)$$

where k is the starting index of the current word (the longest possible subword segment is the entire word). Each of the expressions in the summation represents the probability of concluding the sequence at character t by generating a segment starting at character k . We can efficiently compute the marginal in equation 1 as $p(\mathbf{w}) = p(\mathbf{x}) = \alpha_{|\mathbf{x}|}$.

3.3 Neural Model

Each segment probability is computed as a mixture of 2 probability distributions computed as

$$p(s_{ij} | \mathbf{x}_{< k}) = g_k p_{\text{char}}(s_{ij} | \mathbf{h}_k) + (1 - g_k) p_{\text{lex}}(s_{ij} | \mathbf{h}_k), \quad (6)$$

where \mathbf{h}_k encodes the sequence history, g_k is a mixture coefficient, p_{lex} is a fully connected neural

Language	# Tokens	
	Train	Valid/Test
isiXhosa (xh)	3.4mil	190k
isiZulu (zu)	3.1mil	200k
isiNdebele (nr)	450k	25k
Siswati (ss)	500k	28k

Table 3: Language modelling dataset sizes.

layer that generates the entire segment from a lexicon as a single event, while p_{char} is an LSTM that generates the segment character by character. The probability assigned by p_{char} is computed using the chain rule over the character sequence followed by a special end-of-segment $\langle \text{eos} \rangle$ character.

The character and lexicon models are conditioned on the unsegmented text history through \mathbf{h}_k , a vector representation computed by a character-level LSTM encoder. The LSTM (Hochreiter and Schmidhuber, 1997) is better suited to the probabilistic conditioning required for our model than the Transformer (Vaswani et al., 2017), since it computes a single hidden state \mathbf{h}_k representing the entire sequence history. This can be passed to segment predictors within an efficient dynamic programming algorithm. The mixture coefficient g_k is also computed from \mathbf{h}_k with a fully connected neural layer, so the model can learn when to rely on the lexicon and when to revert to character-by-character generation. The model is trained by maximising the log likelihood over the training corpus.

4 Language Modelling

We evaluate our SSLM on intrinsic language modelling performance to determine whether learning subword segmentation during training can improve

Model	xh			zu			nr			ss		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
BPE	25.00	25.42	25.21	25.20	22.89	23.99	21.23	21.70	21.46	22.30	24.38	23.30
ULM	33.01	34.07	33.53	29.04	27.07	28.02	25.88	26.56	26.21	25.70	29.25	27.36
Morfessor	21.68	17.05	19.09	20.25	17.58	18.82	18.52	17.70	18.10	24.02	22.14	23.04
Entropy-based (Stddev)												
LSTM	40.99	30.43	34.93	39.26	28.23	32.85	38.61	29.01	33.13	33.95	30.40	32.07
Transformer	42.99	33.92	37.92	39.24	28.93	33.31	38.93	29.65	33.66	33.16	29.74	31.35
Subword segmental models												
SSLM	20.43	41.86	27.46	24.03	43.69	31.01	26.44	44.59	33.20	19.56	30.12	23.71
Word-level	44.55	38.07	41.06	49.44	39.75	44.07	41.39	38.01	39.63	38.34	38.26	38.30

Table 4: Morpheme identification (MI) metrics averaged over the annotated evaluation set.

the inherent predictive capabilities of an LM.

The lexicon is constructed before training. It contains all subwords shorter than a prespecified maximum length L that occur in the training corpus. The lexicon size V is also prespecified. The lexicon consists of the V most frequent subwords up to L characters long. The lexicon model p_{lex} outputs a probability for each subword in the lexicon.

4.1 Data

We train our models on LM datasets we compiled for isiXhosa (**xh**), isiZulu (**zu**), isiNdebele (**nr**), and Siswati (**ss**). For each language we collected publicly available datasets and combined them into a single corpus. To avoid some of the pitfalls of low-resource data collection (Kreutzer et al., 2021), we set specific criteria for including datasets. We collected datasets from reputable sources such as the South African Centre for Digital Language Resources (SADiLaR).³ We also inspected individual datasets (manually and using scripts) and discarded datasets of questionable quality (e.g. containing a significant amount of English text). The sources we used are listed in the appendix. We split our corpora 80%/10%/10% into train/validation/test sets. The dataset sizes are listed in Table 3.

4.2 Evaluation

We evaluate our models using bits-per-character (BPC) - an intrinsic evaluation metric that measures how well an LM predicts a corpus. It is cross-entropy-based and normalised by character length, so it allows for comparison across different subword segmentations. BPC is computed as

$$\text{BPC}(X) = -\frac{1}{N} \sum_{\mathbf{x} \in X} \log_2 p(\mathbf{x}), \quad (7)$$

³<https://repo.sadilar.org/discover>

where X is a corpus of sequences \mathbf{x} and N is the length of the corpus in characters.

4.3 Models and Training

For each language we train an SSLM and 6 baselines. Our baselines use 3 standard subword methods: character tokens, BPE, and ULM. For each we train LSTM and Transformer LMs. We tune the hyperparameters of all our models by optimising for validation set BPC. The hyperparameter settings for our SSLMs and baselines are provided in appendix A.

4.4 Results

BPC scores on the validation and test sets are shown in Table 2. The SSLM emerges as the best LM on average across the languages. It achieves the best BPC scores for all languages except isiXhosa, where it still comes very close to the best-performing model. Among the baselines the best neural architecture and subword segmenter depends on the language. For example, the Transformer subword models outperform the LSTM models for isiNdebele, but perform surprisingly poorly for isiXhosa. This inconsistency is one of the primary limitations we are trying to address with our approach. The results show that the SSLM succeeds in this regard - it is more consistent and generally applicable across these languages.

5 Unsupervised Morphological Segmentation

We evaluate our SSLM on unsupervised morphological segmentation (UMS), a challenging task for morphologically rich languages (Poon et al., 2009; Eskander et al., 2019; Üstün and Can, 2016).

Model	xh			zu			nr			ss		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
BPE	40.71	41.79	41.24	45.23	39.00	41.88	38.43	39.76	39.08	32.73	37.75	35.06
ULM	45.92	48.24	47.05	47.04	42.25	44.52	41.48	43.18	42.31	35.91	44.04	39.56
Morfessor	38.14	25.46	30.54	38.83	31.11	34.54	35.11	32.75	33.89	35.61	31.09	33.20
Entropy-based (Stddev)												
LSTM	67.56	40.30	50.49	68.47	39.28	49.92	66.52	41.10	50.81	51.62	42.30	46.50
Transformer	66.98	44.73	53.64	67.72	40.63	50.79	66.62	42.18	51.66	51.12	42.19	46.22
Subword segmental models												
SSLM	30.93	81.33	44.82	36.17	80.67	49.95	37.28	75.84	49.98	32.54	61.11	42.47
Word-level	65.54	50.74	57.20	71.93	50.72	59.49	61.50	53.99	57.50	52.99	52.90	52.95

Table 5: Morpheme *boundary* identification (MBI) metrics averaged over the annotated evaluation set.

This tests to what extent our model discovers morphemes as linguistic units.

5.1 Data

We evaluate our models on morphologically annotated data from the Annotated Text Corpora⁴ released by the National Center for Human Technology (NCHLT) in South Africa (Eiselen and Puttkammer, 2014). The dataset required some preprocessing for our use, which we detail in appendix B. We evaluate our models on the test sets for isiXhosa, isiZulu, isiNdebele, and Siswati. Each test set consists of around 3500 words of free text and morphological analyses of all the words.

5.2 Models

Baselines Morfessor (Creutz and Lagus, 2007) is a widely used family of UMS algorithms. We use the Morfessor Baseline model, which is trained on unsegmented words and based on minimum description length. Smit et al. (2014) notes that Morfessor tends to undersegment when trained on large corpora. To evaluate the true potential of Morfessor we view the dataset size as a hyperparameter. We train Morfessor on several subcorpora of our LM training sets, at different orders of dataset size. We report the results of Morfessor trained on a subset of 10k tokens, which gave the best performance.

Entropy-based segmenter We also implemented a character-level entropy-based segmenter, based on the work of Mzamo et al. (2019a). Their approach consists of training a character-level LM and using the entropy of its probability distribution to predict word segment boundaries. The condi-

tional entropy of x_i in a sequence \mathbf{x} is defined as

$$H(x_i|\mathbf{x}_{<i}) = - \sum_{x \in V} p(x|\mathbf{x}_{<i}) \log p(x|\mathbf{x}_{<i}), \quad (8)$$

where V is the character vocabulary. The entropy-based segmenter splits words at positions where conditional entropy is high. The motivation behind this is that model uncertainty (entropy) will decrease inside a morpheme and increase at morpheme boundaries, where the next character is harder to predict (Elman, 1990).

Mzamo et al. (2019a,b) train n-gram and bi-LSTM LMs for isiXhosa, while Moeng et al. (2021) train left-to-right and right-to-left LSTM LMs for isiXhosa, isiZulu, isiNdebele, and Siswati. Both these works experimented with different entropy-based criteria for segmentation, including segmenting on entropy increases, comparing character entropy to mean word entropy, and using thresholds. We use the character-level LSTM and Transformer LMs trained as baselines in §4. These LMs were tuned for validation BPC, not segmentation accuracy, since we are applying them as fully *unsupervised* segmenters. We experimented with 3 entropy-based criteria:

- **Spike:** Predict subword boundary where entropy increases and then decreases.
- **Increase:** Predict subword boundary where entropy increases.
- **Stddev:** Predict subword boundary where entropy exceeds one standard deviation greater than the mean sequence entropy.

SSLM To apply our SSLMs as segmenters we compute the segmentations that maximise the likelihood of a sentence, using the Viterbi algorithm.

⁴Datasets are available at <https://repo.sadilar.org/handle/20.500.12185/7>

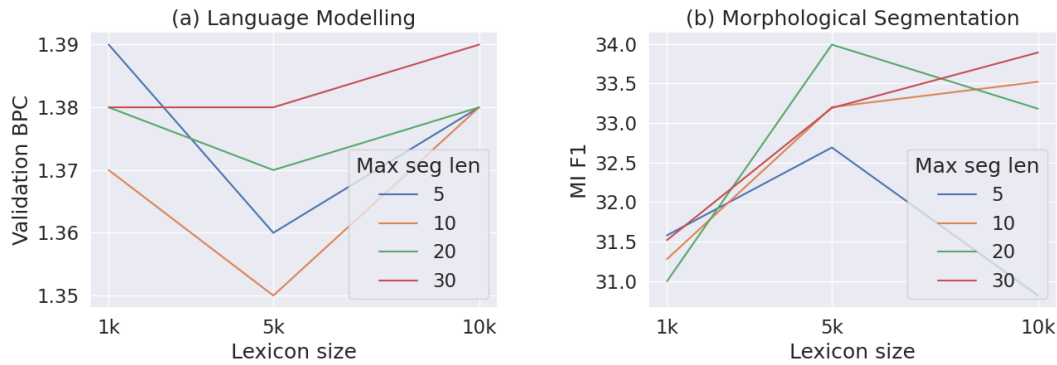


Figure 2: Comparing isiNdebele SSLM performance across varied lexicon sizes and maximum segment lengths.

For each language we evaluate 2 subword segmental models. First we consider the models in §4, trained as long-range SSLMs and carrying hidden states between batches.

Second, we introduce a new set of models trained on single words in isolation. These models (which we refer to as our word-level models) are not LMs. They are trained on the same datasets as our long-range SSLMs, but they process one word at a time without any surrounding context. This removes the need to model long-range linguistic dependencies, allowing the subword segmental model to focus on the short-range task of word prediction and segmentation. We added these models because it is a common approach for UMS algorithms (e.g. Morfessor) to operate on the word-level. We wanted to investigate the morphological segmentation abilities of our subword segmental approach when uncoupled from the task of long-range language modelling. We tune the word-level models on validation BPC and not on segmentation accuracy, so all our models are fully unsupervised with respect to morphological segmentation.

5.3 Results

We report precision, recall, and F1 scores on morpheme identification (MI) and morpheme boundary identification (MBI). MBI is the standard measure of morphological segmentation accuracy, but we include MI because it reflects to what extent our models “discover” morphemes. In MI a morpheme is correctly identified if it is among the subwords a word is segmented into. Table 4 shows a consistent pattern in the results. The subword segmental models outperform the baselines on all the languages. Our word-level models obtain the best MI precision and F1 scores, while the SSLMs generally

obtain better recall. In MBI the goal is to correctly classify whether two consecutive characters are separated by a morpheme boundary. The results in Table 5 reveal the same pattern as for MI, although for some languages the entropy-based segmenters obtain greater precision than any of the SSLMs.

Our word-level subword segmental models do particularly well, emerging as strong unsupervised morphological segmenters. The SSLMs might not be state-of-the-art segmenters, but still outperform segmenters like BPE and ULM on F1 scores. Surprisingly, Morfessor is the weakest among all the models. On the other hand, the entropy-based segmenters achieve consistently impressive results, confirming it as an effective approach to UMS. We only include results for the **Stddev** model here, since it is the strongest entropy-based segmenter overall. The results for all the entropy-based segmenters are included in the appendix.

MBI reveals the type of errors made by our models. The SSLM obtains high recall and low precision, indicating that the model is identifying a large proportion of morpheme boundaries, but often predicting boundaries where there aren’t. This reflects a tendency to over-segment, which might be explained by the low-resource setting. For an LM to utilise long subwords it would have to be exposed to sufficient examples of its use. This might not be possible with smaller training sets, so the model relies on shorter segments instead.

Nevertheless, the SSLM outperforms BPE and ULM on UMS. The greater linguistic plausibility of its segmentations might contribute to its strength as a LM in §4. Letting the model learn which segmentations for language modelling results in the model discovering, to some extent, morphemes as linguistic units.

Sentence	Sibuye sithokoze khulu kwamanikelela emphakathini weentjhabatjhaba ngesekelo labo elinganakuzaza emzabalazweni wethu.
Morphemes	Si-buy-e si-thokoz-e khulu k-w-amanik-elel-a e-m-phakath-ini weentjhabatjhaba nge-sekelo labo eli-nga-nakuzaza e-mzabalazw-eni w-ethu.
SSLM	Si-buy-e s-i-t-h-oko-z-e k-hulu kwam-a-nikele-l-a e-m-phakath-i-n-i ween-tjhaba-tjhab-a n-g-e-sekelo l-a-b-o e-l-i-ngana-kuz-az-a e-mz-abal-az-w-e-n-i w-e-thu.
BPE	Si-bu-ye si-tho-ko-ze khulu kwa-m-ani-k-elela em-phaka-thini ween-tjhaba-tjhaba nge-se-ke-lo la-bo eli-ng-ana-ku-za-za em-za-bala-z-weni we-thu.
ULM	Si-bu-ye si-tho-ko-ze khulu kwama-nikele-la emphakathin-i w-eentjhabatjhab-a nge-se-ke-lo la-bo e-lingana-ku-za-za em-za-ba-la-zwe-ni we-thu.

Table 6: The output of subword segmenters compared to the annotated morphological segmentation of an isiNdebele sentence. Correctly identified morphemes are indicated in green.

6 Analysis

We analyse the effect of hyperparameter and architectural choices on the performance of our SSLMs. This section does not report findings for our word-level models, since we are primarily interested in investigating which model components contribute to the success of our long-range SSLMs (in terms of language modelling and morphological segmentation). Kawakami et al. (2019) found two components to be crucial to the success of their segmental LM for Chinese word discovery: the lexicon and expected length regularisation. The former stores frequent subwords and the latter introduces a regularisation term to the training objective that encourages shorter segments.

Contrary to Kawakami et al. (2019), we did not find length regularisation to be useful. This is because our datasets are much larger than those used in their ablation studies (they use the Brent corpus of 27k words). When a segmental model is trained on a small dataset, it overfits by copying long segments from the lexicon. Length regularisation prevents overfitting by biasing the model towards shorter segments. This is not a problem on larger datasets like ours, because the lexicon cannot cover all possible long segments in the corpus. In fact, our model tends to over-segment rather than under-segment. Table 7 confirms this, showing that SSLM subwords are on average much shorter than morphemes. It is also evident in Table 6, where we show a segmented isiNdebele sentence. The SSLM often over-segments, but sometimes its segmentations are more accurate because of its tendency towards shorter segments. In the examples BPE and ULM fail to identify any of the 1-character morphemes, while the SSLM identifies several.

The lexicon proved to be essential for the SSLM.

Language	xh	zu	nr	ss
Morphemes	2.93	2.86	3.03	3.45
SSLM segments	1.43	1.48	1.64	2.24

Table 7: Average subword length on UMS test sets.

During tuning it consistently improved validation BPC. The average lexicon coefficient ($1 - g_t$ in equation 6) of the isiNdebele SSLM on the LM test set was 0.27, indicating a reliance on the lexicon for subword generation. We analyse two lexicon-related hyperparameters: lexicon size and maximum segment length. Figure 2 compares the performance of isiNdebele SSLMs across lexicon sizes and maximum segment lengths. Figure 2 (a) plots intrinsic LM performance. Smaller lexicon sizes improve LM performance up to a point, with 5k subwords being optimal. A maximum segment length of 10 characters achieves optimal performance across all lexicon sizes, striking a balance between memorising long segments where possible, and otherwise relying on short subwords.

Figure 2 (b) plots UMS performance, where the picture is less clear-cut. Since our model is an *unsupervised* morphological segmenter, we only considered LM performance (validation BPC) when tuning and selecting our final models. Figure 2 shows that optimal LM performance does not necessarily imply optimal UMS performance. Longer maximum segment lengths tend to improve UMS performance. Biasing the model towards longer segments reduces the over-segmentation problem, but relinquishes some LM performance. However, there is at least some correlation between LM and UMS performance, and the model selected on LM performance is not far off optimal UMS accuracy.

7 Related Work

A few African languages have been included in large multilingual LMs, such as mBERT (Devlin et al., 2019) and XLM-R (CONNEAU and Lample, 2019). Ogueji et al. (2021) trained AfriBERTa, a multilingual LM trained on 11 African languages. There has been less work on monolingual LMs for African languages. Ralethe (2020) trained AfriBERT, a masked LM for Afrikaans. Mesham et al. (2021) trained BPE-based autoregressive LMs for isiZulu and Sepedi. Nzeyimana and Rubungo (2022) proposed KinyaBERT, a masked LM for Kinyarwanda with a two-tier neural architecture that incorporates a morphological analyzer.

8 Conclusion

In this paper we proposed subword segmental language modelling (SSLM), an approach that unifies language modelling and subword segmentation. We showed that SSLM improves intrinsic LM performance for low-resource agglutinative languages, while yielding subwords that approximate morphemes better than previous approaches. As opposed to most neural model architectures in NLP research, which are either language-agnostic or overfit to high-resource languages, our model is designed to suit agglutinative languages like the Nguni languages of South Africa. Our results show that learning subword segmentation in training overcomes some of the limitations of existing subword segmenters. For future work, the SSLM could be applied to downstream NLP tasks suited to its autoregressive architecture, such as text generation. More generally, the idea of learning subword segmentation during training could be adapted to other NLP models and tasks.

Limitations

We evaluate our model on languages from a single language group - the Nguni languages. Our findings might not hold for languages with different types of morphological complexity (e.g. fusional languages, where segmentation is difficult because morphemes are fused together). The SSLM achieved consistently good LM performance across all four Nguni languages, but we had to tune the lexicon size and maximum segment length separately for each language. These optimal hyperparameter values varied across languages and would have to be tuned from scratch for new languages.

Our subword segmental approach is able to improve over all baselines as a morphological segmenter, but only if we train it as a word-level sequence model. The SSLM outperforms standard segmenters like BPE and ULM, but performs worse than our entropy-based baselines on F1 scores. This shows that there is a deterioration in segmentation performance because the SSLM is required to model long-range linguistic dependencies - the model tends to over-segment words. We only evaluate our segmentations with automatic evaluation metrics, which provides a rigid, morpheme-based perspective on the segmentation quality. It would be ideal to also include human evaluations of the linguistic plausibility of segmentations.

Ethical Considerations

We release LM datasets for 4 Nguni languages, consisting of free text split into train/validation/test sets. Our datasets are compilations of existing, publicly available datasets. The datasets we sourced are listed in Table 10 in the appendix. As outlined in section 4.1, we took certain steps to ensure that the datasets we sourced were of reasonable quality.

Nevertheless, since the source datasets were originally scraped from the web, we acknowledge that we do not avoid all the pitfalls of large scale data collection for low-resource languages (most notably, the presence of text in other languages). Furthermore, most of the data is sourced from South African government publications, so they are domain-specific to some extent. The texts cover diverse topics, but generally fall within the categories and style expected of government publications.

Acknowledgements

This work is based on research supported in part by the National Research Foundation of South Africa (Grant Number: 129850) and the South African Centre for High Performance Computing. Computations were performed using facilities provided by the University of Cape Town’s ICTS High Performance Computing team: hpc.uct.ac.za. Francois Meyer is supported by the Hasso Plattner Institute for Digital Engineering, through the HPI Research School at the University of Cape Town.

References

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2021. [CANINE: pre-training an efficient](#)

- tokenization-free encoder for language representation. *CoRR*, abs/2103.06874.
- Alexis CONNEAU and Guillaume Lample. 2019. **Cross-lingual language model pretraining**. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- 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.
- Mathias Creutz and Krista Lagus. 2007. **Unsupervised models for morpheme segmentation and morphology learning**. *ACM Trans. Speech Lang. Process.*, 4(1).
- 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.
- C. M. Downey, Fei Xia, Gina-Anne Levow, and Shane Steinert-Threlkeld. 2021. **A masked segmental language model for unsupervised natural language segmentation**. *arXiv:2104.07829*.
- Roald Eiselen and Martin Puttkammer. 2014. **Developing text resources for ten South African languages**. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3698–3703, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Jeffrey L. Elman. 1990. **Finding structure in time**. *Cognitive Science*, 14(2):179–211.
- Ramy Eskander, Judith Klavans, and Smaranda Muresan. 2019. **Unsupervised morphological segmentation for low-resource polysynthetic languages**. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 189–195, Florence, Italy. Association for Computational Linguistics.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Kazuya Kawakami, Chris Dyer, and Phil Blunsom. 2019. **Learning to discover, ground and use words with segmental neural language models**. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6429–6441, Florence, Italy. Association for Computational Linguistics.
- Diederik P. Kingma and Jimmy Ba. 2015. **Adam: A method for stochastic optimization**. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Stav Klein and Reut Tsarfaty. 2020. **Getting the ##life out of living: How adequate are word-pieces for modelling complex morphology?** In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 204–209, Online. Association for Computational Linguistics.
- Lingpeng Kong, Chris Dyer, and Noah A. Smith. 2016. **Segmental recurrent neural networks**. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.
- Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wajah, Daan van Esch, Nasanbayar Ulzii-Orshikh, Alahsera Auguste Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmongkol Sarin, Sokhar Samb, Benoît Sagot, Clara E. Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Javier Ortiz Suárez, Iroro Fred Onome Orife, Kelechi Ogueji, Rubungo Andre Niyongabo, Toan Nguyen, Mathias Müller, André Müller, Shamsuddeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhlov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa de Silva, Sakine Çabuk Ballı, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. 2021. **Quality at a glance: An audit of web-crawled multilingual datasets**. *Transactions of the Association for Computational Linguistics (TACL)*.
- 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.
- Stuart Mesham, Luc Hayward, Jared Shapiro, and Jan Buys. 2021. **Low-resource language modelling of south african languages**. In *Proceedings of the Second Southern African Conference for Artificial Intelligence Research (SACAIR)*, Online. Springer.
- Tumi Moeng, Sheldon Reay, Aaron Daniels, and Jan Buys. 2021. **Canonical and surface morphological segmentation for nguni languages**. In *Proceedings of the Second Southern African Conference for Artificial Intelligence Research (SACAIR)*, pages 125–139, Online. Springer.

- Lulamile Mzamo, Albert Helberg, and Sonja Bosch. 2019a. [Towards an unsupervised morphological segmenter for isiXhosa](#). In *Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA)*, pages 166–170.
- Lulamile Mzamo, A.S. Helberg, and Sonja E. Bosch. 2019b. Evaluation of combined bi-directional branching entropy language models for morphological segmentation of isiXhosa. In *South African Forum of Artificial Intelligence Research*.
- 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.
- Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. 2021. [Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages](#). In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 116–126, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Hoifung Poon, Colin Cherry, and Kristina Toutanova. 2009. [Unsupervised morphological segmentation with log-linear models](#). In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 209–217, Boulder, Colorado. Association for Computational Linguistics.
- Sello Ralethe. 2020. [Adaptation of deep bidirectional transformers for Afrikaans language](#). In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2475–2478, Marseille, France. European Language Resources Association.
- Elizabeth Salesky, Andrew Runge, Alex Coda, Jan Niehues, and Graham Neubig. 2020. [Optimizing segmentation granularity for neural machine translation](#). *Machine Translation*, 34(1):41–59.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Neural machine translation of rare words with subword units](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.
- 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.
- Zhiqing Sun and Zhi-Hong Deng. 2018. [Unsupervised neural word segmentation for Chinese via segmental language modeling](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4915–4920, Brussels, Belgium. Association for Computational Linguistics.
- Elsabé Taljard and Sonja E. Bosch. 2006. [A comparison of approaches to word class tagging: Disjunctively vs. conjunctively written bantu languages](#). *Nordic Journal of African Studies*, 15:428–442.
- Ahmet Üstün and Burcu Can. 2016. [Unsupervised morphological segmentation using neural word embeddings](#). In *Statistical Language and Speech Processing*, pages 43–53, Cham. Springer International Publishing.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.
- Chong Wang, Yining Wang, Po-Sen Huang, Abdelrahman Mohamed, Dengyong Zhou, and Li Deng. 2017. [Sequence modeling via segmentations](#). In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, page 3674–3683. JMLR.org.
- Lihao Wang, Zongyi Li, and Xiaoqing Zheng. 2021a. [Unsupervised word segmentation with bi-directional neural language model](#). *arXiv:2103.01421*.
- Xinyi Wang, Sebastian Ruder, and Graham Neubig. 2021b. [Multi-view subword regularization](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 473–482, Online. Association for Computational Linguistics.
- Yi Zhu, Benjamin Heinzerling, Ivan Vulić, Michael Strube, Roi Reichart, and Anna Korhonen. 2019a. [On the importance of subword information for morphological tasks in truly low-resource languages](#). In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 216–226, Hong Kong, China. Association for Computational Linguistics.
- Yi Zhu, Ivan Vulić, and Anna Korhonen. 2019b. [A systematic study of leveraging subword information for learning word representations](#). 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 912–932, Minneapolis, Minnesota. Association for Computational Linguistics.
- Judit Ács. 2019. [Exploring BERT’s vocabulary](#).

A SSLM Training

We tuned our SSLMs to optimise BPC on our LM validation sets. We used the Adam optimizer (Kingma and Ba, 2015), halving the learning rate if validation loss failed to improve for 3 epochs and stopping when no validation loss improvement occurred for 6 epochs. We used several standard regularisation techniques, including dropout (on all except recurrent layers), weight decay, and gradient clipping. Table 9 shows the hyperparameter settings we used for our subword segmental models and baseline LMs. The optimal lexicon size and maximum segment length varied across the languages, as shown in Table 8.

We trained our models on virtually partitioned instances of NVIDIA A100 GPUs with 3 compute units and 20GB memory. The isiXhosa and isiZulu long-range SSLMs converged after about 40 epochs of the training corpus, taking 3 days to train. The isiNdebele and Siswati long-range SSLMs converged after 30 epochs, taking 10 hours to train. The word-level SSLMs trained quite slowly, since each word is processed as an individual sequence. Therefore we trained our isiXhosa and isiZulu models on 500k-word subcorpora of the LM training sets (matching the sizes of the isiNdebele and Siswati datasets). These converged before 20 epochs, taking 10 hours to train. Segmenting the morphological evaluation data took less than a few minutes on a laptop computer, since the Viterbi algorithm is computationally efficient.

B UMS Data

The data consists of a train/test set of free text in which words have been morphologically analysed. Words are segmented into their *canonical segmentations* i.e. standardised morphemes that do not necessarily correspond to word substrings (Cotterell et al., 2016). For example, the canonical segmentation of the isiXhosa word “yedwa” is “ya-i-dwa”. Since our SSLM segments words into substrings, we require *surface segmentations* (segments correspond to substrings) for evaluation. Most of the segmentations can be used as is, because the canonical and surface segmentations are identical. Where the segmentations differ, we use the scripts made available by Moeng et al. (2021) to map canonical segmentations to surface segmentations. They employ a heuristic approach based on the Levenshtein distance minimal edit operations to map from the de-segmented canonical form to

the surface form. They also filter out tokens that are unsuitable for morphological segmentation.

Model	Lexicon size	Max seg len
Long-range SSLM		
isiXhosa	10k	5
isiZulu	10k	5
isiNdebele	5k	10
Siswati	10k	20
Word-level SSLM		
isiXhosa	10k	10
isiZulu	5k	20
isiNdebele	10k	10
Siswati	5k	20

Table 8: Lexicon hyperparameters for our SSLMs.

Hyperparameter	Subword segmental models		Baseline models	
	Long-range	Word-level	LSTM	Transformer
Attention heads				4/8*
LSTM layers	3	1	3	3
Embedding size	512	512	128/512*	512
Hidden size	1024	1024	1024	1024
Learning rate	0.001	0.005	0.001	0.001
Dropout	0.5	0.2	0.2	0.1
Batch size	64	16	64	64
Sequence	120 chars	1 word	120 chars	120 chars
Weight decay	1e-5	1e-5	1e-5	1e-5
Gradient clip	1.0	1.0	1.0	1.0

Table 9: Hyperparameter settings for all our models, with * indicating where the optimal hyperparameter value (based on validation BPC) depended on the language. For embedding size, 128 was optimal for isiXhosa and Siswati, while 512 was optimal for isiZulu and isiNdebele. The Transformer models had 8 attention heads for isiXhosa and isiZulu, and 4 for isiNdebele and Siswati.

Data set	Type	Source
isiXhosa		
NCHLT Text	monolingual	South African government websites
SADiLaR Monolingual	monolingual	South African government websites
Navy Corpus	parallel	South African government websites
isiZulu		
NCHLT Text	monolingual	South African government websites
Autshumato	parallel	South African government websites
Isolezwe News Corpus	monolingual	news articles
isiNdebele		
NCHLT Text	monolingual	South African government websites
Siswati		
NCHLT Text	monolingual	South African government websites

Table 10: Our language modelling data sets were compiled from these publicly available data sets. We split the individual corpora into train/validation/test sets before combining them respectively into one train/validation/test data set. This ensured that the individual corpora are equally distributed in the training and evaluation sets.

	xh			zu			nr			ss		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Char-LSTM Entropy												
Spike	34.71	36.97	35.81	33.56	34.36	33.95	29.12	32.62	30.77	28.94	35.38	31.83
Increase	27.64	38.14	32.06	29.21	37.52	32.85	25.90	36.36	30.25	24.32	37.01	29.35
Stddev	40.99	30.43	34.93	39.26	28.23	32.85	38.61	29.01	33.13	33.95	30.40	32.07
Char-Transformer Entropy												
Spike	34.80	37.84	36.26	33.22	34.83	34.00	27.03	31.10	28.92	25.82	33.12	29.02
Increase	29.65	40.00	34.06	29.83	38.47	33.60	24.89	35.03	29.10	22.86	35.91	27.93
Stddev	42.99	33.92	37.92	39.24	28.93	33.31	38.93	29.65	33.66	33.16	29.74	31.35

Table 11: Morpheme identification performance metrics for all entropy-based models.

	xh			zu			nr			ss		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Char-LSTM Entropy												
Spike	51.08	56.25	53.54	53.08	54.98	54.01	46.36	54.77	50.22	41.57	56.61	47.93
Increase	40.15	63.61	49.23	44.62	63.41	52.38	39.31	63.01	48.42	35.48	65.27	45.97
Stddev	67.56	40.30	50.49	68.47	39.28	49.92	66.52	41.10	50.81	51.62	42.30	46.50
Char-Transformer Entropy												
Spike	50.40	57.24	53.60	52.11	55.91	53.94	44.35	54.40	48.86	38.03	55.52	45.14
Increase	42.12	64.68	51.02	45.19	64.59	53.17	38.36	61.62	47.29	33.16	63.73	43.62
Stddev	66.98	44.73	53.64	67.72	40.63	50.79	66.62	42.18	51.66	51.12	42.19	46.22

Table 12: Morpheme boundary identification performance metrics for all entropy-based models.

(a) isiXhosa sentence segmentation	
Sentence	Siphinda kwakhona umbulelo wethu osuka emazantsi entliziyi kwabezizwe ngezizwe ngenkxaso yabo engagungqiyo ekuxhaseni umzabalazo wethu.
Morphemes	Si-phind-a kwa-khona u-m-bulelo w-ethu o-suk-a emazantsi e-n-tliziyi kwa-bezizwe nge-zi-zwe ngenkxaso y-abo engagungqiyo e-ku-xhas-eni u-m-zabalazo w-ethu.
SSLM	S-i-phin-d-a kwak-hon-a u-m-bule-l-o w-e-thu osuk-a e-m-a-zant-s-i e-n-tl-iziy-o k-w-a-b-e-z-i-zw-e ngezi-zw-e n-g-e-nkxa-s-o y-a-b-o e-ngag-ungqi-yo e-k-u-xhas-e-n-i u-m-zab-alaz-o w-e-thu.
BPE	Si-phin-da kwa-khona um-bu-lelo we-thu o-su-ka ema-zantsi ent-li-zi-yo kwa-b-ezi-zwe ngezi-zwe ngen-kxaso ya-bo enga-gu-ng-q-i-yo eku-xha-s-eni um-za-bala-zo we-thu.
ULM	Si-phi-nda kwa-khona u-mbu-lelo we-thu o-suka e-ma-za-nts-i e-nt-li-zi-yo kwa-be-zi-z-we nge-zi-z-we nge-nkxaso ya-bo e-n-ga-gu-ng-q-i-yo e-ku-xh-a-se-ni um-za-ba-la-zo we-thu
(b) isiZulu sentence segmentation	
Sentence	Siyaphinda sibonga siyanconcoza emphakathini womhlaba ngokuseseka kwawo emzabalazweni wethu.
Morphemes	Si-ya-phind-a si-bong-a si-ya-nconcoz-a e-m-phakath-ini wo-m-hlaba n-gokusesek-a kwa-wo e-mzabalazw-eni w-ethu.
SSLM	S-i-yaph-inda s-i-bong-a siya-nco-nco-z-a e-mphak-a-t-h-i-n-i w-o-m-hlaba n-g-o-k-u-s-eseka k-w-a-w-o emz-abala-z-w-e-n-i wethu.
BPE	Si-ya-phi-nda si-bo-nga si-ya-n-co-n-co-za em-phakathi-ni wo-m-hlaba ngoku-se-se-ka kwa-wo em-za-bala-zweni we-thu.
ULM	Si-ya-phi-nda si-bo-nga si-ya-n-co-n-co-za emphakathini wo-mhlaba ngoku-se-se-ka kwa-wo em-za-ba-la-zwe-ni we-thu.
(c) Siswati sentence segmentation	
Sentence	Sendlulisa kubonga kwetfu kummango wemave emhlaba ngekwesekela umzabalazo wetfu ngendlela lengenakunyakatiswa.
Morphemes	S-endlulisa ku-bong-a kwetfu ku-mmango wemave emhlaba ngekwesekela u-mzabalazo wetfu nge-n-dlela le-n-genakunyakatiswa.
SSLM	S-e-n-dlulis-a k-u-bong-a kw-e-tfu k-u-m-mango w-e-mave emhlaba n-g-e-k-w-e-sekel-a u-m-zaba-laz-o w-e-tfu ngendle-l-a l-e-ngena-ku-n-yaka-t-isw-a
BPE	S-en-dlu-lisa kubo-nga kwetfu kum-ma-ngo wema-ve em-hlaba ngekwese-kela um-za-bala-zo we-tfu ngendlela le-n-gen-aku-nya-kati-swa.
ULM	Se-ndlu-lisa kubo-nga kwe-tfu ku-m-ma-ngo we-mave e-mhlaba ngekwese-kela um-za-ba-la-z-o we-tfu ngendlela le-n-gena-ku-nya-kati-swa.

Table 13: The output of subword segmenters compared to the annotated morphological segmentation of Nguni language sentences. Correctly identified morphemes are indicated in green.