The Learning Dynamics of Subword Segmentation
for Morphologically Diverse Languages

Francois Meyer and Jan Buys
Department of Computer Science
University of Cape Town
`francois.meyer@uct.ac.za, jbuys@cs.uct.ac.za`

Abstract

Subword segmentation is typically applied in preprocessing and stays fixed during training. Alternatively, it can be learned *during* training to optimise the training objective. In this paper we study the learning dynamics of subword segmentation: if a language model can dynamically optimise tokenisation, how do its subwords evolve during training? To explore this, we extend the subword segmental language model (SSLM), a framework for learning subwords during training, to support pretraining and finetuning. We train models for three typologically diverse languages to study learning dynamics across the morphological spectrum: IsiXhosa is conjunctive (long word forms composed of many morphemes), Setswana is disjunctive (morphemes written as separate words), and English represents a typological middle ground. We analyse subword dynamics from a linguistic perspective, tracking morphology, productivity, and fertility. We identify four stages of subword learning, with the morphologically complex isiXhosa exhibiting greater instability. During finetuning, subword boundaries shift to become finer-grained. Lastly, we show that learnable subwords offers a promising approach to improve text generation and cross-lingual transfer for low-resource, morphologically complex languages.

1 Introduction

Subword tokenisers like BPE (Sennrich et al., 2016) and ULM (Kudo, 2018) are applied during preprocessing, after which they remain unchanged during language model (LM) pretraining and finetuning. If tokenisers produce subwords that are ineffective units for language modelling, this cannot be fixed during training. This is especially problematic for settings in which models are sensitive to tokenisation decisions, such as low-resource tasks (Zhu et al., 2019a; Wang et al., 2021) and morphologically rich languages (Zhu et al., 2019b; Klein and Tsarfaty, 2020).
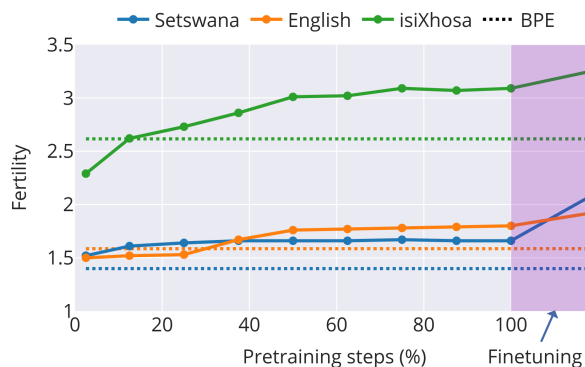

Figure 1: Subword fertility (average subwords per word) gradually plateaus for isiXhosa, while converging early for English and Setswana.

Ideally, subword segmentation should be learned during end-to-end training. Previous works achieve this by jointly optimising segmentation and model parameters (Kreutzer and Sokolov, 2018; He et al., 2020; Meyer and Buys, 2022). In this paper, we use this paradigm to study the learning dynamics of subword segmentation. When an LM is able to learn subword segmentation, how do its subword boundaries evolve over the course of pretraining and finetuning? The question has practical implications, as it reveals how tokenisation requirements vary across different stages of training. There is no guarantee that subword units that are optimal for the early epochs of pretraining would also be suitable for later stages of pretraining, or meet the demands of subsequent task-specific finetuning.

Subword learning dynamics may also depend on linguistic properties, particularly morphology. We conduct experiments on three languages spanning a range of morphological complexity: Setswana, English, and isiXhosa. Setswana and isiXhosa are South African languages that represent opposite ends of the morpho-orthographic spectrum. Both are classified as agglutinative, but they differ in how orthographic (written) words are constructed (Taljard and Bosch, 2006). IsiXhosa is conjunctive: morphemes are concatenated into long word forms, whose meaning depends on morphological decom-

Setswana			English				isiXhosa		
ga	ke	itse	I	do	n't	know	a	nd	azi
ba	a	bala	they	are	read	ing	ba	funda	
ke	a	leboga	thank	you			ndi	ya	bulela

Table 1: Morphological decompositions of translations, illustrating a spectrum from disjunctive (Setswana) to conjunctive (isiXhosa), with English in between.

position. Setswana is disjunctive: morphemes are written as space-separated words and many words consist of single morphemes. English, with its limited inflectional morphology, serves as a typological middle ground. The examples in Table 1 illustrate the contrasting subword structures.

Our study requires a model capable of dynamically learning subword segmentation during pretraining and finetuning. For this purpose we extend the subword segmental language model (SSLM) (Meyer and Buys, 2022), which jointly optimises subword segmentation and LM training. We develop a Transformer-based variant, **T-SSLM**, incorporating learnable subword segmentation into modern LM practices like pretraining, task-specific finetuning, and text generation.

We pretrain T-SSLMs for Setswana, English, and isiXhosa, tracking learned subwords across checkpoints from three linguistic perspectives. **Morphological alignment** measures how closely subwords align with morphological boundaries. **Productivity** (Gutierrez-Vasques et al., 2023) measures the generative capacity of subwords, while **Fertility** (Ács, 2019) estimates lexical coverage. Across all three criteria, isiXhosa exhibits the greatest instability, while Setswana converges earliest. We identify distinctive phases of subword learning, characterised by clear changes and trends in our linguistic metrics. Our results show that different stages of training and different languages necessitate characteristically different subword units.

Finally, to evaluate the impact of learnable subwords on downstream performance, we evaluate T-SSLM on isiXhosa text generation. It comfortably outperforms tokenisation-based LMs, with isiXhosa data-to-text BLEU gains of 6.25 and a notable reduction in text degeneration. This suggests finetuning subword segmentation as a promising avenue for improving low-resource text generation, while confirming – from a model performance perspective – that subword requirements evolve across different stages of training.

2 Related Work

2.1 Learning dynamics of language models

Previous research tracked the evolution of syntactic and semantic knowledge during training. Some studies track internal representation change (Saphra and Lopez, 2019; Chiang et al., 2020; Liu et al., 2021; Müller-Eberstein et al., 2023), while others track performance on grammatical tasks (Choshen et al., 2022; Evanson et al., 2023). In all these studies (as in most LM research) tokenisation is fixed, so there is no subword learning trajectory to study. Gutierrez-Vasques et al. (2023) analyse how BPE token properties change over successive merges; however, BPE is still fixed after preprocessing, so the emergence of subwords in LM training is not studied. That requires considering approaches that incorporate subword learning into training.

2.2 Unifying training and tokenisation

Previous works unify training and subword learning by marginalising over different candidate tokenisations during training. This has been used to learn subwords for translation (Kreutzer and Sokolov, 2018; He et al., 2020; Meyer and Buys, 2023) and unsupervised Chinese word segmentation (Sun and Deng, 2018; Kawakami et al., 2019; Downey et al., 2022). Meyer and Buys (2022) adapt this technique to propose subword segmental language modelling (SSLM), an LSTM-based LM architecture that dynamically optimises subword segmentation. They train small-scale SSLMs for low-resource, agglutinative languages. They do not conduct finetuning or downstream evaluation, but perplexity-based evaluation demonstrates the potential of SSLM over tokenisation-based LMs.

3 Transformer SSLM

We leverage the technique underlying the SSLM of Meyer and Buys (2022) – learning subword units during LM training by marginalising over potential tokenisations – as an opportunity to study subword learning dynamics. The original SSLM architecture is based on a shallow LSTM. To align SSLM with current best practices for pretraining, task-finetuning, and open-ended text generation, we develop a Transformer version of the model. We propose **T-SSLM**, a Transformer-based adaptation of SSLM that brings the model up to date with modern LM architectures.

From a modelling perspective, our T-SSLM is a conceptually straightforward but technically non-

trivial extension of the original SSLM. We do not position our architectural changes as the main contribution of this paper. Our primary research goal is empirical analysis: to study subword learning dynamics across different stages of training. We develop T-SSLM as a new tool for studying subword segmentation as a learnable component of language modelling. We will release our implementation of T-SSLM upon publication.[1]

In this section, we present a technical overview of how T-SSLM dynamically learns subword segmentation during pretraining and finetuning. For a comprehensive introduction to the underlying SSLM framework, we refer readers to Meyer and Buys (2022). Here we summarise the method behind the original SSLM, outline how we adapt it to Transformer-based modelling, and explain how we use T-SSLM to track subword learning.

3.1 SSLM background

For a standard tokenisation-based LM with parameters θ, the probability assigned to a document D is based on a particular fixed subword tokenisation T (e.g. obtained with BPE), computed using the chain rule as

$$p_\theta(D, T) = \prod_{i=1}^{n} p_\theta(t_i | t_{<i}), \qquad (1)$$

where $t_1, t_2, ..., t_n$ is the sequence of subwords that D is tokenised into by T.

Instead of using a single pre-determined tokenisation T, SSLM marginalises over all possible subword segmentations during training. The probability assigned to a document D is computed as

$$p_\theta(D) = \sum_{T \in \pi(D)} p_\theta(D, T), \qquad (2)$$

where $\pi(D)$ denotes the set of all potential subword segmentations of document D.

The number of possible subword segmentations of a document D grows exponentially with its length, so the marginalisation in Eq. 2 becomes intractable. Meyer and Buys (2022) propose a dynamic programming algorithm to compute Eq. 2 efficiently. Suppose document D consists of characters $\mathbf{c} = c_1, c_2, \ldots, c_{|D|}$. The algorithm computes the marginalised probability up to the kth character as $p(D_{1:k}) = \alpha_k$, where $\alpha_0 = 1$ and α_k

are forward scores computed as

$$\alpha_k = \sum_{j=f(c,k)}^{k} \alpha_{j-1} p(t = c_{j:k} | c_{<j}), \qquad (3)$$

where f returns the index of the first character in the word containing the kth character, thereby prohibiting subwords that cross word boundaries. During pretraining, the marginal (Eq. 2) of the full document is optimised by computing $p(D) = \alpha_{|D|}$.

3.2 T-SSLM: adapting SSLM for pretraining and finetuning

Meyer and Buys (2022) parameterise the SSLM described above with an LSTM (Hochreiter and Schmidhuber, 1997), so the subword probabilities in Eq. 3 are conditioned on LSTM encodings. We propose T-SSLM, a new variant of SSLM parameterised by a Transformer (Vaswani et al., 2017). This enables larger-scale pretraining and aligns SSLM with modern LM architectures.

To construct a neural LM that can compute $p_\theta(D, T)$ for any candidate segmentation T, Meyer and Buys (2022) introduce several architectural innovations, which we replicate in T-SSLM. One of these innovations is to encode the preceding text $t_{<i}$ as an untokenised character sequence $c_{<j}$ to achieve computationally feasible conditioning on the tokenisation history. In T-SSLM, we use a character-level Transformer for this encoding instead of an LSTM. This Transformer forms the backbone of our architecture: the subword probabilities computed in Eq. 3 are conditioned on its learned output representations.

Like the original SSLM, T-SSLM computes the next-subword probability $p_\theta(t_i|\cdot)$ as a weighted mixture of two distributions: one over character sequences (p_{char}) and another over a subword lexicon (p_{lex}). The lexicon contains a fixed set of high-frequency subwords, while p_{char} allows the model to compose arbitrary subwords from characters. This design, with implementation details in Appendix A.1, enables T-SSLM to adapt both its segmentation algorithm *and* its subword vocabulary over the course of training.

The Transformer backbone enables scalable pretraining. In addition, we introduce two extensions to the original SSLM to enable downstream application of T-SSLM. First, we modify the dynamic programming algorithm. The original formulation of Eq. 3 computes the marginal likelihood $p(D)$

of a full sequence D, but prompt-based finetuning requires maximising the *conditional* likelihood $p(O|C)$, where C is an input prompt and O is the expected completion. We derive a modified version of the original dynamic program to handle conditional likelihoods (see Appendix A.2 for derivation). Second, we develop a custom decoding algorithm (outlined in Appendix A.3) to enable open-ended text generation with T-SSLM, since standard beam search implementations are incompatible with the mixture model described above.

3.3 Extracting learned subwords for analysis

During pretraining, T-SSLM learns to assign higher probability to segmentations T that maximise Eq. 2. After pretraining, or for any intermediate checkpoint, we can extract the model's preferred segmentation T^* of document D by applying Viterbi decoding to the dynamic program of Eq. 3:

$$T^* = \underset{T \in \pi(D)}{\operatorname{argmax}} \, p_\theta(D, T). \qquad (4)$$

This is the probability-maximising segmentation according to the model's current parameters θ, reflecting the subword segmentation that the model has learned as optimal at that stage of training.

During finetuning, T-SSLM adapts its subword segmentation to suit the requirements of the downstream task. Rather than optimising full-document likelihood, as in pretraining, the model now maximises the conditional likelihood of output completions given prompts. This leads to new subword preferences that better support task-specific generation. We again use the Viterbi algorithm to extract the model's preferred segmentation of a document after finetuning, revealing how subword segmentation changes from pretraining to finetuning.

4 Experimental Setup

We pretrain monolingual T-SSLMs for Setswana, English, and isiXhosa. Pretraining and finetuning hyperparameters are provided in Appendix B.

4.1 Pretraining

T-SSLM's time complexity (detailed in §8.1) makes it computationally challenging to scale the pretraining data size to match state-of-the-art LMs. However, we view the technique as an opportunity for analysis: a unique lens through which to analyse the role of subword segmentation in LM training. For pretraining, we prioritise high-quality corpora over scale to ensure that models

Dataset	Train	Valid	Test
Pretraining (# words)			
Setswana: PuoData	4.06m	247k	212k
English: BabyLM	9.95m	998k	994k
isiXhosa: WURA	13.08m	767k	724k
Downstream isiXhosa finetuning (# examples)			
Triples-to-isiXhosa (T2X)	3,859	600	378
MasakhaNEWS (MNH)	1,032	147	297

Table 2: Pretraining and finetuning dataset statistics.

are trained on clean, representative language input. Our corpus sizes are listed in Table 2. For isiXhosa we use the 13m-word WURA (Oladipo et al., 2023) corpus, created by filtering mC4 (Xue et al., 2021) to discard low-quality data. For the lower-resourced Setswana we use the 4m-word PuoData corpus (Marivate et al., 2023). For English we use the 10m-word BabyLM corpus (Warstadt et al., 2023). To track learning dynamics, we store model checkpoints at regular pretraining intervals. After pretraining, we analyse the subwords of nine checkpoints spread evenly across pretraining: one checkpoint after the first epoch, one after the final epoch, and seven additional checkpoints spaced at regular intervals (1/8ths of total pretraining).

4.2 Finetuning

We study how subwords change during finetuning from two perspectives: how they adapt to a specific downstream task, and how they shift when transferring across languages with different morphological structures (i.e. cross-lingual finetuning). We finetune all our T-SSLMs (Setswana, English, isiXhosa) on two isiXhosa text generation tasks, data-to-text and headline generation. We focus on isiXhosa for finetuning because its complex morphological structure makes it a challenging language to adapt subword modelling cross-lingually.

For data-to-text we use the T2X dataset (Meyer and Buys, 2024). It contains triples (subject, relation, object) paired with descriptive isiXhosa sentences e.g. (South Africa, leader, Cyril Ramaphosa) → "uCyril Ramaphosa yinkokheli yoMzantsi Afrika" ("Cyril Ramaphosa is the leader of South Africa"). For headline generation from the body of a news article we use MasakhaNEWS (Adelani et al., 2023), which contains news articles paired with headlines. This is a much harder task than data-to-text, especially considering the smaller finetuning dataset (see Table 2).
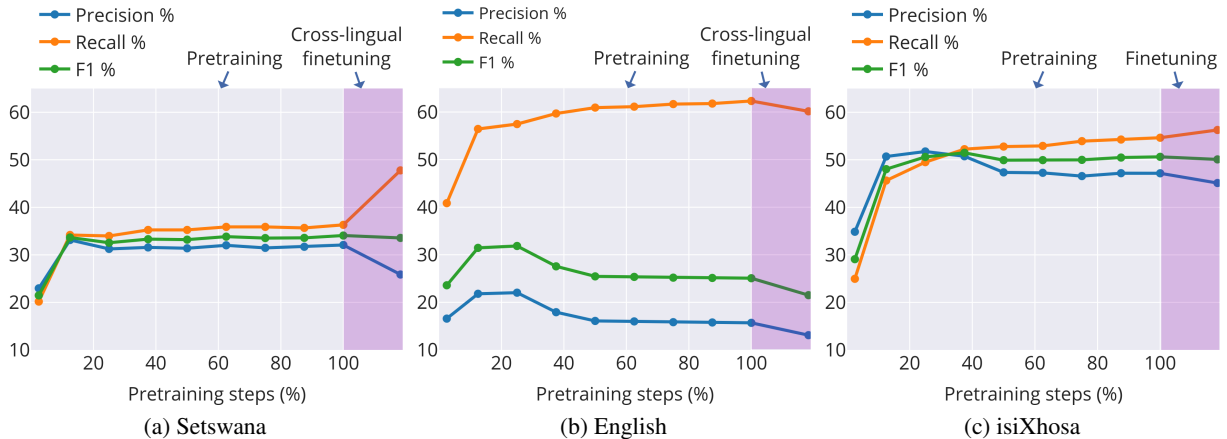
Figure 2: Morphological boundary overlap between learned subwords and morphological segmentations. Pretraining is performed on Setswana/English/isiXhosa, while finetuning is performed on isiXhosa data-to-text.

4.3 Baselines

For each language, we pretrain and finetune five LMs with fixed subword tokenisation (hyperparameters are reported in Appendix C). We use the same datasets as our T-SSLMs to ensure a fair comparison. Three baselines use well-established tokenisers: (1) BPE (Sennrich et al., 2016), (2) ULM (Kudo, 2018), and (3) BPE-dropout (Provilkov et al., 2020). Additionally, we use (4) character-level segmentation and (5) byte-based modelling, both of which have been shown to outperform subword-based LMs on low-resource languages (Edman et al., 2022, 2024; Adelani et al., 2022).

These LMs are baselines for downstream evaluation (Section 6), allowing us to compare learnable subword segmentation and fixed tokenisation in terms of text generation performance. Besides plotting BPE as a reference in Figure 1, we do not include these baselines in our analysis of subword learning dynamics (Section 5), since their tokenisation is fixed and does not evolve during training.

4.4 Subword analysis

For any given T-SSLM checkpoint, we can encode a document and extract its probability-maximising subword segmentation using the Viterbi algorithm. This reveals the model's preferred tokenisation of the text at that point in pretraining/finetuning. We analyse and track these learned segmentations from three linguistic perspectives.

4.4.1 Morphological alignment

Our models do not have access to morphological annotations, but we investigate to what extent they "discover" morphemes as subword units by tracking the amount of overlap between learned subwords

and morphemes. For Setswana and isiXhosa we use the NCHLT test corpus (Eiselen and Puttkammer, 2014), which contains 5,000 morphologically annotated words for each language. For English we use the 2022 SIGMORPHON dataset (Batsuren et al., 2022), which contains 18,000 morphologically segmented words. We compare the subwords learned by our models to the gold-standard morphological segmentations. To measure morphological alignment, we compute precision, recall, and F1 for morphological boundary identification.

4.4.2 Productivity and Idiosyncrasy

Morphological productivity is the capacity of a morpheme to combine with other morphemes to form new words (Bybee, 2001). Productive morphemes occur in many word types, often forming regular morphological patterns. For example, in isiXhosa prefixes can be attached to stems to indicate plurality e.g. "abahlobo" (friends) and "abazali" (parents). In contrast, *idiosyncratic* morphemes are contained in few word types, but these types occur frequently. For example, the isiXhosa stem "–ntu" (human) will not occur in as many unique words as prefixal morphemes, but the words in which "–ntu" occurs are commonly used e.g. "umntu" (person) and "abantu" (people).

We quantify productivity and idiosyncrasy as proposed by Gutierrez-Vasques et al. (2023),

$$\text{productivity}(s) = |W_s|, \quad (5)$$

$$\text{idiosyncrasy}(s) = \frac{\sum_{w \in W_s} \text{freq}(w)}{|W_s|}, \quad (6)$$

where W_s is the set of unique words in which s occurs and $\text{freq}(w)$ is the corpus frequency of a word
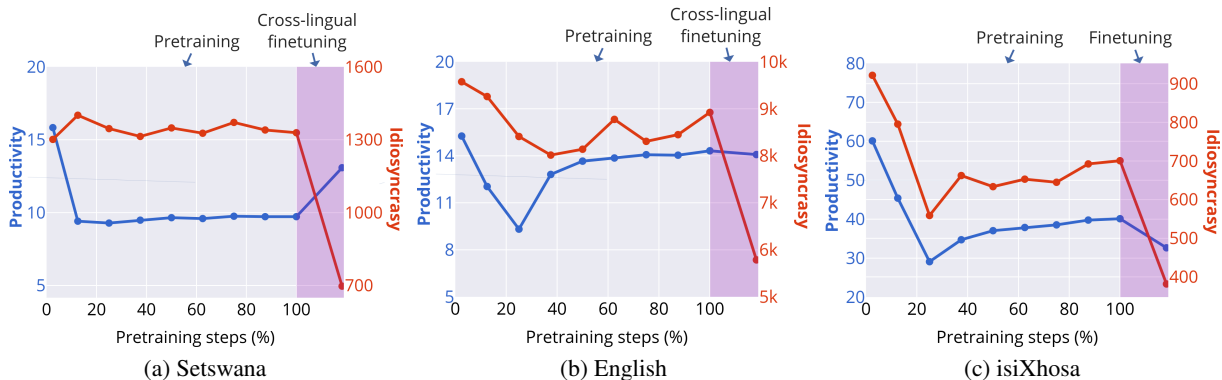
Figure 3: The average productivity and idiosyncrasy of learned subwords. Pretraining is performed on isiXhosa and Setswana, respectively, while finetuning is conducted on isiXhosa data-to-text generation.

w. Productivity quantifies how many *types* contain a subword, while idiosyncrasy is the average *token* frequency of types containing a subword. We compute these metrics from the subword segmentation and frequencies of our LM validation corpora.

4.4.3 Fertility

The fertility of a tokeniser is the average number of subwords per segmented word (Ács, 2019). It is a measure of lexical coverage – the extent to which the words of a language are included in the tokeniser vocabulary. A fertility of 1 indicates that every word in a corpus is included in the vocabulary (no segmented words), while a higher fertility corresponds to finer-grained segmentation and lower lexical coverage. We also compute fertility based on the segmentation of our LM validation corpora.

5 Subword Learning Dynamics

We analyse how T-SSLM subword units evolve during pretraining and finetuning.[2] The learning dynamics are visualised in Figures 1, 2 and 3. We identify four distinctive stages of subword learning. We include representative examples of segmentations at each stage in Tables 3 and 4.

Stage 1: Rapid evolution of subwords

The initial stage of subword learning (first 20% of pretraining) is characterised by dramatic changes in subword boundaries. The models start with randomly initialised subword boundaries and find subword units that optimise the training objective in the initial period of a high-loss training. As shown in Figure 2, initial subwords have little morphological alignment, but models quickly adjust their subword

---

[2] We present the analysis for finetuning on isiXhosa data-to-text. Results are similar for our other task, headline generation.

boundaries to more closely align with morpheme boundaries. This shows that an important part of the early subword learning is the emergence of morphemes, to an extent, as subword units. We include examples of morphological subwords emerging after stage 1 in Table 3.

Figure 3 shows similarly rapid changes in subword productivity. For English and isiXhosa, productivity and idiosyncrasy decrease at a constant rate. Gutierrez-Vasques et al. (2023) found a similar pattern in early BPE merges. Highly productive, high-frequency subwords are identified at the outset, after which the model incorporates progressively less common subwords. This leads to decreased productivity and idiosyncrasy. For Setswana, the learning trajectory is different. Productivity also decreases, but idiosyncrasy remains stable. This also agrees with Gutierrez-Vasques et al. (2023), who showed that BPE tokens for morphological simple languages have low productivity but high idiosyncrasy. Many Setswana subwords are near-whole words, so they occur in few types, but the types are not rare (i.e. high idiosyncrasy).

Stage 2: Inflection point for English & isiXhosa

After an initial period of adjustment, Setswana subwords stabilise. In contrast, at this stage (between 20% and 40% of pretraining) isiXhosa and English undergo a sudden change in learning trajectories. After the initial period of decreasing productivity and idiosyncrasy, both metrics increase (Figure 3c). This also coincides with a large increase in fertility (Figure 1), indicating finer grained segmentation. We interpret this as the initially expanded vocabulary being trimmed down. In the initial phase of exploration, the model identified a broad set of candidates for effective subwords. Subsequently, learn-

Language	Setswana			English			isiXhosa		
Word	malapa	dingwe	rona	assuming	lately	donuts	umntu	ukuxhasa	zoncedo
Stage 1	mal apa	dingw e	rona	assu ming	l ately	don uts	umntu	uku xhasa	zon cedo
Stage 2	ma lapa	di ngwe	rona	a ssum ing	late ly	donut s	um ntu	uku xhas a	zoncedo
Stage 3	ma lapa	di ngwe	rona	a ssum ing	latel y	donut s	u m ntu	uku xhas a	zo ncedo
Stage 4	ma lapa	dingw e	rona	assu m ing	latel y	donut s	um ntu	uku xhasa	zo ncedo

Table 3: Examples of changes in learned subword segmentations throughout training, highlighting subwords corresponding to morphemes (morphemes obtained from linguistically annotated datasets introduced in §4.4.1).

Pretrained	A keem D ent	zi se New Y ork	M e x ico	sase L o s A ngele s
Finetuned	A k e e m D e n t	zi se N e w Y o r k	M e x i c o	sase L o s A n g e l e s

Table 4: Segmentations of entities in the data-to-text dataset, produced by isiXhosa T-SSLM before/after finetuning. Fertility increases as named entities, which lack meaningful subword units, are segmented into characters.

ing shifts to finding subwords that improve LM generalisation, by preferring subwords that occur in many word types (e.g. productive morphemes) and/or in high-frequency words (idiosyncratic).

This inflection point represents a shift in the morphological alignment of isiXhosa subwords, as shown by Figure 2c. Initially a large proportion of subwords have morphological boundaries (high precision). After the inflection point, the model increases its coverage of morphological boundaries (high recall), but at the cost of more morphologically unsound boundaries. The model is erring on the side of overly-aggressive segmentation, as shown in the finer-grained segmentation of "umntu" and "zoncedo" in Table 3 (from stage 2 to 3). This increases morphological boundary coverage but over-segments some morphemes.

Stage 3: Stabilised learning dynamics

The mid-to-late phases of pretraining are more stable than early subword learning. Throughout pretraining, isiXhosa and English dynamics are more variable than Setswana. For example, Setswana fertility quickly converges to a low value of 1.66. Because of its disjunctive orthography, Setswana leaves little opportunity for aggressive subword segmentation, so low fertility is unsurprising. English, which exhibits some morphological inflection, settles on a slightly higher fertility of 1.8. In contrast, isiXhosa fertility gradually increases to 3.05, almost double that of Setswana.

The unstable dynamics of isiXhosa demonstrates the challenges of modelling the subword structure of a morphologically complex language. Even late in pretraining, optimising the loss requires changes in subword boundaries, as better subword units continually emerge. In contrast, the Setswana subwords discovered early on are sufficient for the remainder of pretraining.

Figure 1 plots fertility of BPE tokenisers for each language (from the baselines of §4.3) as fixed references. For each language, the fertility of learned subwords is slightly higher than BPE. Equipped with the ability to optimise subword boundaries, T-SSLM learns finer-grained segmentation than BPE. One possibility is that T-SSLM converges on tokenisation that balances the benefits of BPE and character-level modelling, suggesting a potentially more optimal middle ground between them.

Stage 4: Task-oriented realignment

During finetuning, fertility increases for all three languages (Figure 1). Previous work showed finergrained units, such as characters or bytes, are beneficial for low-resource languages with small finetuning datasets (Edman et al., 2022, 2024; Adelani et al., 2022). More aggressive segmentation may also suit the data-to-text dataset, since it contains many named entities (e.g. "Akeen Dent" in Table 4). Names cannot be segmented into semantically coherent subwords, since they do not consist of morphemes. It is reasonable to model names as character sequences, which is what we observe after finetuning. We present examples of this change from pretraining to finetuning in Table 4.

For isiXhosa, productivity and idiosyncrasy decrease during finetuning (Figure 3c). As the model narrows in on subword patterns that occur frequently in the data-to-text dataset, its subword units lose the more general morphological expressivity

Pretrained →	isiXhosa			English			Setswana			isiXhosa			English		
Finetuned →	isiXhosa data-to-text									isiXhosa headline generation					
Model	chrF	BLEU	deg%	chrF	BLEU	deg%	chrF	BLEU	deg%	chrF	BLEU	deg%	chrF	BLEU	deg%
BPE	44.9	12.0	6.4	41.1	10.1	**1.6**	41.8	10.3	3.7	14.9	0.0	10.8	12.3	0.2	1.0
ULM	46.1	13.2	6.1	36.5	8.2	2.1	42.5	11.5	**1.1**	15.6	0.0	8.1	12.3	0.2	2.0
BPE-dropout	44.6	11.0	4.5	35.6	7.4	2.7	39.9	9.8	1.9	14.4	0.0	8.4	13.9	0.2	2.0
Characters	43.5	7.6	5.3	49.0	10.0	2.4	44.5	7.3	4.8	17.1	0.8	2.4	17.3	0.2	**0.7**
Bytes	48.2	13.2	4.5	48.7	14.3	3.7	45.1	10.1	7.9	18.0	0.0	5.1	18.5	0.2	1.0
T-SSLM	**49.2**	**19.5**	**1.1**	**49.1**	**17.0**	1.6	**46.3**	**14.4**	2.4	**21.7**	**1.9**	**0.3**	**21.6**	**0.9**	3.4

Table 5: Test set performance of finetuned LMs on isiXhosa tasks, measured by BLEU and chrF (normalised between 0 and 100) and text degeneration (deg% is the proportion of examples that produce incoherent, repetitive text). We omit headline generation results for Setswana-pretrained models, as all models produced zero scores.

required for isiXhosa pretraining. Our Setswana and English T-SSLMs are finetuned cross-lingually on isiXhosa data-to-text. To model the complex morphological structure of isiXhosa, they lose some morphological alignment in the language of pretraining (Figure 2a) and improve isiXhosa morphological alignment (Figure 5 in the appendix).

6 Downstream Text Generation

Next we study subword learning dynamics via its effect on downstream performance. Our preceding analysis found that finetuning subword segmentation leads to the emergence of task-specific subwords (examples in Table 4). This raises the question of whether the ability to adapt subword boundaries during finetuning provides a performance advantage over models with fixed tokenisation (e.g. BPE or ULM), which lack this flexibility.

We finetune T-SSLM and five tokenisation-based baseline LMs (see §4.3) on two isiXhosa text generation tasks (data-to-text and headline generation, see §4.2). We evaluate performance with BLEU (Papineni et al., 2002) and chrF (Popović, 2015), which quantifies subword-level overlap. Additionally we measure the proportion of test examples that produce degenerative text (incoherent or repetitive language). A generation is considered degenerative if a word is repeated at least three times (word-level repetition) or if a word is longer than 30 characters (subword-level repetition).

The results in Table 5 shows that T-SSLM outperforms tokenisation-based models across reference-based metrics, with substantial BLEU gains. It overcomes text degeneration to a considerable degree for in-language (isiXhosa) pretraining and finetuning, producing much less repetition (Table 7 in the appendix shows examples of generated text). Text degeneration is a challenging problem in low-resource tasks, as evidenced by the high propor-

tions of degenerative examples produced by baselines. T-SSLM does not match the performance of massively multilingual LMs (Meyer et al., 2024), but our aim is to isolate the effect of task-adaptable subwords over fixed tokenisation. We therefore focus on comparing SSLM to tokenisation-based LMs of the same size trained on the same datasets. In this comparison our results demonstrate the benefit of learnable subwords in data-scarce settings.

We observe similar cross-lingual results, where T-SSLMs pretrained on Setswana/English are finetuned on isiXhosa text generation. T-SSLM consistently outperforms baselines across BLEU and chrF, with competitive text degeneration scores (especially considering reference-based performance differences). This demonstrates the model's ability to improve performance by adapting its subword boundaries to the complex morphology of isiXhosa. These results highlight the potential of learnable subwords, not only for in-language finetuning, but also for cross-lingual transfer in low-resource tasks.

7 Conclusion

We investigated the learning dynamics of LMs where the subword segmentation is learnt jointly with knowledge of other aspects of language, in contrast to previous studies that only considered a setup where the subwords segmentation is fixed. We identified contrasting patterns in the subword dynamics of Setswana, English, and isiXhosa. While learning subword boundaries dynamically may be computationally prohibitive for larger-scale pretraining, it enables a new type of LM analysis, providing insight into the role of subwords in different stages of training and demonstrating the potential of task-adaptable segmentation. Our findings also underscore the distinct subword needs of typologically diverse languages, particularly in low-resource and morphologically complex settings.

8 Limitations

8.1 Computational complexity

Like other models that marginalise over multiple tokenisations (Kreutzer and Sokolov, 2018; He et al., 2020; Meyer and Buys, 2022), T-SSLM is computationally more expensive than standard LM training. Despite the dynamic program of Eq. 3, marginalisation adds unavoidable cost by computing probabilities for multiple candidate subword sequences (standard LMs compute the likelihood for a single subword sequence). In practice, T-SSLM training times increase by an order of magnitude ($\times 10$) over tokenisation-based LMs.

The added computational complexity of learning subwords prevents us from pretraining on larger datasets. It is an open question whether our findings would scale – whether similar subword learning trajectories would be observed for larger-scale pretraining and higher-resourced languages. This would require pretraining a large-scale SSLM from scratch on a massive corpus, which is not possible given the compute resources at our disposal. The claims of this paper are limited to smaller-scale pretraining of low-resource languages. This is a relevant and important use case, since data scarcity and computational constraints often co-occur in practice (Ahia et al., 2021).

8.2 Linguistic scope

Our findings are limited to three languages, Setswana, English, and isiXhosa, so we cannot guarantee that they will generalise to other language families. We chose these languages based on their morpho-orthopraphic properties, since it allows us to compare the subword learning dynamics of languages with contrasting linguistic subword structures. This contrast simplifies the identification and analysis of differences in subword learning trajectories, making these languages ideal for an initial exploration of subword dynamics.

Our downstream finetuning experiments are limited to isiXhosa data-to-text and headline generation. Given the under-resourcedness of isiXhosa, these are the only text generation evaluation dataset available with sufficient training instances for finetuning. We cannot guarantee similar performance gains for other text generation tasks. Throughout our paper, we are careful not to overstate the impact of our findings in terms of downstream task performance. Instead of positioning T-SSLM as the solution to low-resource text generation, we place more emphasis on the learning dynamics analysis and our finding that pretraining and downstream tasks have different requirements with regards to subword modelling.

References

David Adelani, Jesujoba Alabi, Angela Fan, Julia Kreutzer, Xiaoyu Shen, Machel Reid, Dana Ruiter, Dietrich Klakow, Peter Nabende, Ernie Chang, Tajuddeen Gwadabe, Freshia Sackey, Bonaventure F. P. Dossou, Chris Emezue, Colin Leong, Michael Beukman, Shamsuddeen Muhammad, Guyo Jarso, Oreen Yousuf, Andre Niyongabo Rubungo, Gilles Hacheme, Eric Peter Wairagala, Muhammad Umair Nasir, Benjamin Ajibade, Tunde Ajayi, Yvonne Gitau, Jade Abbott, Mohamed Ahmed, Millicent Ochieng, Anuoluwapo Aremu, Perez Ogayo, Jonathan Mukiibi, Fatoumata Ouoba Kabore, Godson Kalipe, Derguene Mbaye, Allahsera Auguste Tapo, Victoire Memdjokam Koagne, Edwin Munkoh-Buabeng, Valencia Wagner, Idris Abdulmumin, Ayodele Awokoya, Happy Buzaaba, Blessing Sibanda, Andiswa Bukula, and Sam Manthalu. 2022. A few thousand translations go a long way! leveraging pre-trained models for African news translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3053–3070, Seattle, United States. Association for Computational Linguistics.

David Ifeoluwa Adelani, Marek Masiak, Israel Abebe Azime, Jesujoba Alabi, Atnafu Lambebo Tonja, Christine Mwase, Odunayo Ogundepo, Bonaventure F. P. Dossou, Akintunde Oladipo, Doreen Nixdorf, Chris Chinenye Emezue, Sana Al-azzawi, Blessing Sibanda, Davis David, Lolwethu Ndolela, Jonathan Mukiibi, Tunde Ajayi, Tatiana Moteu, Brian Odhiambo, Abraham Owodunni, Nnaemeka Obiefuna, Muhidin Mohamed, Shamsuddeen Hassan Muhammad, Teshome Mulugeta Ababu, Saheed Abdullahi Salahudeen, Mesay Gemeda Yigezu, Tajuddeen Gwadabe, Idris Abdulmumin, Mahlet Taye, Oluwabusayo Awoyomi, Iyanuoluwa Shode, Tolulope Adelani, Habiba Abdulganiyu, Abdul-Hakeem Omotayo, Adetola Adeeko, Abeeb Afolabi, Anuoluwapo Aremu, Olanrewaju Samuel, Clemencia Siro, Wangari Kimotho, Onyekachi Ogbu, Chinedu Mbonu, Chiamaka Chukwuneke, Samuel Fanijo, Jessica Ojo, Oyinkansola Awosan, Tadesse Kebede, Toadoum Sari Sakayo, Pamela Nyatsine, Freedmore Sidume, Oreen Yousuf, Mardiyyah Oduwole, Kanda Tshinu, Ussen Kimanuka, Thina Diko, Siyanda Nxakama, Sinodos Nigusse, Abdulmejid Johar, Shafie Mohamed, Fuad Mire Hassan, Moges Ahmed Mehamed, Evrard Ngabire, Jules Jules, Ivan Ssenkungu, and Pontus Stenetorp. 2023. MasakhaNEWS: News topic classification for African languages. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific*

Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), pages 144–159, Nusa Dua, Bali. Association for Computational Linguistics.

Orevaoghene Ahia, Julia Kreutzer, and Sara Hooker. 2021. The low-resource double bind: An empirical study of pruning for low-resource machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3316–3333, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Joan Bybee. 2001. *Phonology and Language Use*. Cambridge Studies in Linguistics. Cambridge University Press.

Cheng-Han Chiang, Sung-Feng Huang, and Hung-yi Lee. 2020. Pretrained language model embryology: The birth of ALBERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6813–6828, Online. Association for Computational Linguistics.

Leshem Choshen, Guy Hacohen, Daphna Weinshall, and Omri Abend. 2022. The grammar-learning trajectories of neural language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8281–8297, Dublin, Ireland. Association for Computational Linguistics.

C.m. Downey, Fei Xia, Gina-Anne Levow, and Shane Steinert-Threlkeld. 2022. A masked segmental language model for unsupervised natural language segmentation. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 39–50, Seattle, Washington. Association for Computational Linguistics.

Lukas Edman, Gabriele Sarti, Antonio Toral, Gertjan van Noord, and Arianna Bisazza. 2024. Are character-level translations worth the wait? comparing ByT5 and mT5 for machine translation. *Transactions of the Association for Computational Linguistics*, 12:392–410.

Lukas Edman, Antonio Toral, and Gertjan van Noord. 2022. Subword-delimited downsampling for better character-level translation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 981–992, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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).

Linnea Evanson, Yair Lakretz, and Jean Rémi King. 2023. Language acquisition: do children and language models follow similar learning stages? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12205–12218, Toronto, Canada. Association for Computational Linguistics.

Ximena Gutierrez-Vasques, Christian Bentz, and Tanja Samardžić. 2023. Languages through the looking glass of BPE compression. *Computational Linguistics*, 49(4):943–1001.

Xuanli He, Gholamreza Haffari, and Mohammad Norouzi. 2020. Dynamic programming encoding for subword segmentation in neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3042–3051, Online. 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.

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.

Julia Kreutzer and Artem Sokolov. 2018. Learning to segment inputs for NMT favors character-level processing. In *Proceedings of the 15th International Conference on Spoken Language Translation*, pages 166–172, Brussels. International Conference on Spoken Language Translation.

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.

Zeyu Liu, Yizhong Wang, Jungo Kasai, Hannaneh Hajishirzi, and Noah A. Smith. 2021. Probing across time: What does RoBERTa know and when? In

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 820–842, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Vukosi Marivate, Moseli Mots'Oehli, Valencia Wagner, Richard Lastrucci, and Isheanesu Dzingirai. 2023. Puoberta: Training and evaluation of a curated language model for setswana. In *Artificial Intelligence Research. SACAIR 2023. Communications in Computer and Information Science*.

Francois Meyer and Jan Buys. 2022. Subword segmental language modelling for nguni languages. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6636–6649, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Francois Meyer and Jan Buys. 2023. Subword segmental machine translation: Unifying segmentation and target sentence generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2795–2809, Toronto, Canada. Association for Computational Linguistics.

Francois Meyer and Jan Buys. 2024. Triples-to-isiXhosa (T2X): Addressing the challenges of low-resource agglutinative data-to-text generation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16841–16854, Torino, Italia. ELRA and ICCL.

Francois Meyer, Haiyue Song, Abhisek Chakrabarty, Jan Buys, Raj Dabre, and Hideki Tanaka. 2024. NGLUEni: Benchmarking and adapting pretrained language models for nguni languages. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 12247–12258, Torino, Italia. ELRA and ICCL.

Max Müller-Eberstein, Rob van der Goot, Barbara Plank, and Ivan Titov. 2023. Subspace chronicles: How linguistic information emerges, shifts and interacts during language model training. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13190–13208, Singapore. Association for Computational Linguistics.

Akintunde Oladipo, Mofetoluwa Adeyemi, Orevaoghene Ahia, Abraham Owodunni, Odunayo Ogundepo, David Adelani, and Jimmy Lin. 2023. Better quality pre-training data and t5 models for African languages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 158–168, Singapore. Association for Computational Linguistics.

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

Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online. Association for Computational Linguistics.

Naomi Saphra and Adam Lopez. 2019. Understanding learning dynamics of language models with SVCCA. 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 3257–3267, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

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.

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.

Xinyi Wang, Sebastian Ruder, and Graham Neubig. 2021. 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.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on

developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

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

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, 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 T-SSLM Adaptation

A.1 Mixture model

T-SSLM encodes the preceding text as an untokenised character sequence, using a character-level Transformer. This discards information about segmentation history, but achieves computationally feasible conditioning. For next-subword prediction, we adapt the approach of Meyer and Buys (2022) to use a Transformer encoding. To condition next-subword prediction on preceding text, we pass the final output embedding to two decoding subnetworks, a character-level model and a softmax layer over a subword lexicon.

We model next subword probability as a mixture of the two distributions,

$$p_\theta(t_i|t_{<i}) = \phi_i p_{\text{char}}(t_i|t_{<i}) + (1 - \phi_i)p_{\text{lex}}(t_i|t_{<i}), \quad (7)$$

where ϕ_i is a mixture coefficient computed by a fully connected layer $\phi_i = \text{MLP}(t_{<i})$. The subword lexicon contains the V most frequent character n-grams in the training corpus. The lexicon p_{lex} models common subwords (e.g. morphemes), while the character-level decoder p_{char} allows probability assignments to arbitrary character sequences, even those unseen in training (e.g. rare words, names). The lexicon and character components are both softmax layers, jointly trained with the mixture coefficient MLP, which allows T-SSLM to learn in which contexts to rely on the lexicon or character-level predictions.

A.2 Finetuning algorithm

While we can use the existing dynamic programming algorithm of Meyer and Buys (2022) for pretraining, we have to adapt it to finetune our T-SSLM for prompt-based text generation. Their dynamic program is represented by Eq. 3, which computes (and thereby maximises) the likelihood for a full text sequence. This is not compatible with open-ended text generation, which are cast as text completion tasks: given an input prompt C as context, generate the expected output O. The model is finetuned to maximise $p(O|C)$ directly, instead of finetuning on generating both C and O. Such completion-only finetuning is standard practice in prompt-based generation.

Suppose D is a completed prompt consisting of (C, O). Instead of maximising $p(D)$, we maximise

$$p(O|C) = \frac{p(C,O)}{p(C)} = \frac{p(D)}{p(C)} = \frac{\alpha_{|C|+|O|}}{\alpha_{|C|}}, \quad (8)$$

where $\alpha_{|C|}$ is the forward score up to the end of the input context. By maximising this finetuning objective, T-SSLM adapts its subword segmentation to optimise the generation of completion O given input context C. This allows the model to adjust the subword boundaries learned during pretraining to optimise performance on the downstream task.

A.3 Decoding

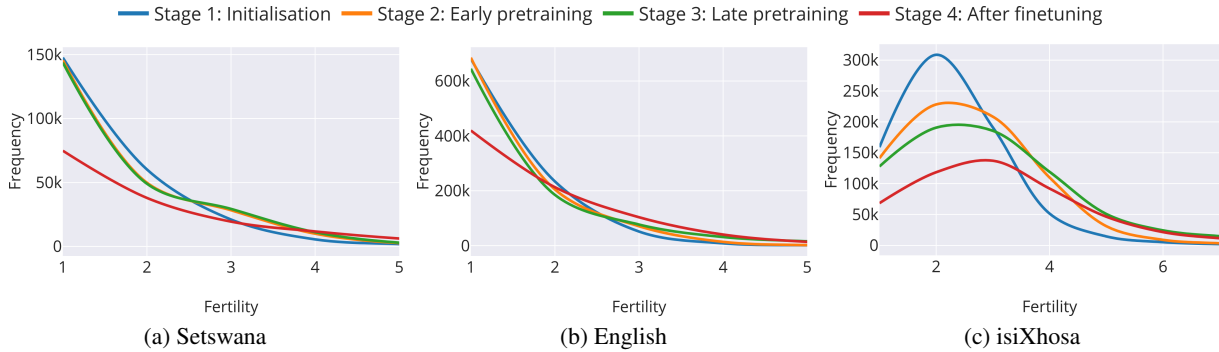Standard beam search operates on one vocabulary, but the mixture model of T-SSLM combines two

Figure 4: Fertility distributions (subwords per segmented word) across the four training stages identified in this study. All languages exhibit increasing fertility as pretraining progresses, with a more pronounced shift during finetuning. Changes are especially dramatic for isiXhosa, whose complex morphology leads to larger distributional shifts than Setswana or English.
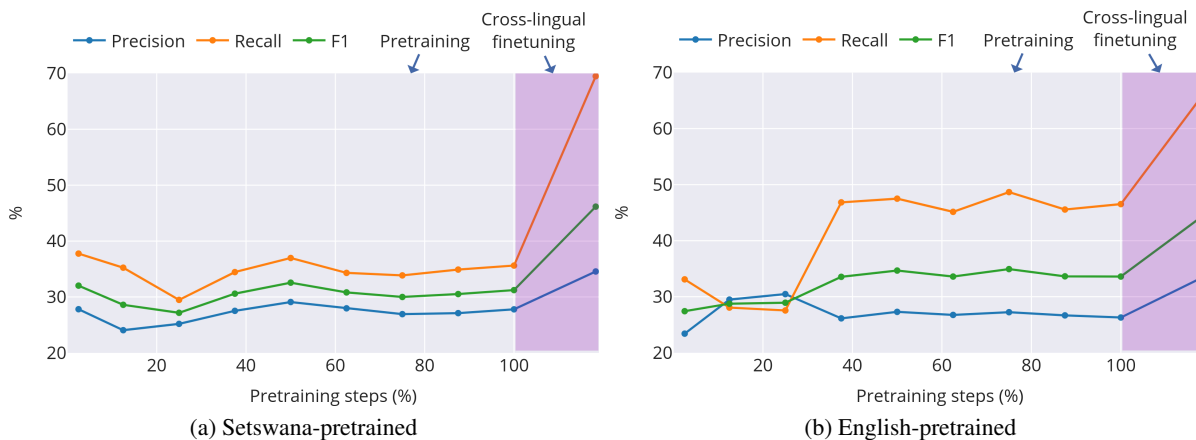

Figure 5: Boundary overlap between isiXhosa morphemes and subwords learned by our T-SSLMs pretrained on Setswana (left) and English (right). During isiXhosa finetuning, the models adjusts their subword segmentation to align with the morphological boundaries of the new language.

vocabularies, subwords and characters, complicating text generation. We adapt dynamic decoding, previously proposed for encoder-decoder subword segmental models, for decoder-only generation. Dynamic decoding (Meyer and Buys, 2023) generates the target translation one character at a time and predicts subword boundaries based on mixture model probabilities. We adapt this algorithm for our model by conditioning next-character probabilities on preceding prompt context, as opposed to a source-language sentence. During generation, we use a beam size of 5 in our dynamic decoding algorithm.

B T-SSLM Hyperparameters

B.1 Pretraining

Our SSLM architecture settings are based on the fairseq `base_lm_architecture`: a 6-layer decoder-only model with 8 attention heads per block, and 512-dimensional embeddings. We pretrain our SSLMs until validation loss stops improving, which occurs by 40 epochs for English and isiXhosa and by 100 epochs for Setswana.

For isiXhosa pretraining, we use a learning rate of 5e-4 with an inverse square-root scheduler, 4000 warmup steps, and a dropout rate of 0.1. We use an effective batch size of 256 sequences with a maximum sequence length of 512 characters. Our subword lexicon consists of the 10k most frequent character n-grams in the pretraining corpus and we set the maximum subword segment length, a hyperparamter of the SSLM training algorithm, to 5 characters. For English and Setswana pretraining, we use the same architecture configuration and training hyperparameters as isiXhosa pretraining, except that we use a subword vocabulary of 5k subwords (since the English and Setswana pretraining corpus are both smaller than the isiXhosa corpus).

(a) Full prompt template examples for finetuning
South Africa + leaderName + Cyril Ramaphosa # # =uCyril Ramaphosa yinkokheli yomzantsi Afrika.
Ethiopia + currency + Ethiopian birr # # =Imali yase-Ethiopia yi-Ethiopian Birr.
Denmark + capital + Copenhagen # # =ICopenhagen likomkhulu laseDenmark.

(b) Incomplete prompt template examples for testing
Romania + capital + Bucharest # # =
India + currency + Indian rupee # # =
Netherlands + leaderName + Mark Rutte # # =

Table 6: To finetune decoder-only PLMS for T2X isiXhosa data-to-text generation, we cast the task as prompt completion. Here we demonstrate the prompt template, "{input text} # # ={output text}", with examples from the T2X dataset. During finetuning (a), we only maximise the generation of text after the equals sign "=". During testing (b), the special tokens "# # =" prompt output generation.

B.2 Finetuning

To finetune our LMs, we transform examples into the following template:

```
{input text} # # = {output text}
```

During completion-only finetuning (§A.2), we maximise the likelihood of the text following "=". Table 6 contains examples of prompts created for the T2X dataset. For finetuning, we perform a grid search over varying hyperparameter settings and select the final settings based on downstream validation performance. We use a batch size of 16 and a learning rate of 1e-4. The learning rate undergoes 500 warmup steps and an inverse square-root scheduler. We perform finetuning for 20 epochs and selected the final model checkpoint based on validation performance.

C Baseline Hyperparameters

For BPE, ULM, and BPE-dropout, we use a subword vocabulary size match the subword lexicon of our SSLM (5k for Setswana and English, 10k for isiXhosa). We implement our baselines with the Huggingface Transformers library (Wolf et al., 2020), training LMs of the same size as our isiXhosa SSLMs (6 layers, 8 attention heads, 512-dimensional embeddings, maximum sequence length of 512). We pretrain the baseline LMs using a batch size of 64 and the default Huggingface pretraining hyperparameters for causal language modelling: a learning rate of 5e-5, no warmup, a linear scheduler, and a dropout rate of 0.1.

To finetune our baselines, we also performed a validation-based grid search over hyperparameter values, settling on a batch size of 4, a learning rate of 5e-5 with no warmup and a linear scheduler. As for our SSLM, we finetuned our baselines for 20 epochs and evaluated the checkpoints with the best validation performance. For decoding, we use standard beam search with beam size of 5.

Example (a)	
Reference output	UMassimo Drago udlalela uA.S.D. S.S. Nola 1925.
BPE	UMassimo Drago udlalele i-AS.D.S.
ULM	Umdlali we-A.S. Nolandelilungu le-A.S. Nolandelilungu le-A.S. Nolandelilungu le-A.S. Nolandelilungu le-A.
BPE-dropout	UMassimo Drago wadlalela i-A.S. Nola 1925. S. Nola 1925. S. Nola 1925. ... ...
Char	UMassimo Drago wayengumthetheli kwi - A. S. D. S. S. D. S. S. D. S. D. S. ...
Byte	UMassimo Drago wayengumthetheli kwi-A.S.D. S.D.S. Nola 1925.
T-SSLM	UMassimo Drago udlalela i-A.S.D. S.D.S. Nola 1925.
Example (b)	
Reference output	I-St. Vincent-St. Mary school samabanga aphakamileyo siseMelika.
BPE	ISikhumbuzo i-St. Vincentlanticellanticellanticellanticellanticellanticella...
ULM	Isikolo seMary High School sise-United States.
BPE-dropout	ISithili i-St Vincent St. Vincent St. St. Vincent St. Vincent St. ...
Char	I - St. Vincentaccccccccccccccccccccccccccccccccccccccccccccccccccccccccc...
Byte	ISt. Vincentury High School iseUnited States.
T-SSLM	St. Vincent-St. Mary High School ifumaneka e United States.

Table 7: Examples of model generations on the T2X test set, compared to reference output texts. The tokenisation-based baselines often generate incoherent, repetitive text, while SSLM avoids text degeneration in all instances.