


UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER SCIENCE

CS Honours Project Final Paper 2024

Title: Deep Learning for Morphological Parsing of Nguni Languages

Author: Simbarashe Mawere

Project Abbreviation: MORPH_PARSE

Supervisor(s): Francois Meyer

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	0
Theoretical Analysis	*0*	*25*	0
Experiment Design and Execution	*0*	*20*	20
System Development and Implementation	*0*	*20*	5
Results, Findings and Conclusions	*10*	*20*	20
Aim Formulation and Background Work	*10*	*15*	15
Quality of Paper Writing and Presentation	*10*		10
Quality of Deliverables	*10*		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	*0*	*10*	
Total marks		**80**	

Deep Learning for Morphological Parsing of Nguni Languages

Supervisor: Francois Meyer

Simbarashe Mawere
mwrsim003@myuct.ac.za
University of Cape Town
Cape Town, Western Cape, South Africa

ABSTRACT

Morphological parsing is an important task in the fields of natural language processing (NLP) and computational linguistics since it extracts important grammatical information need for natural language understanding (NLU) by language models. It has not, however, been explored much for the four Nguni languages—isiNdebele, siSwati, isiXhosa, and isiZulu—as a collective. Currently, no such tool for morphological parsing exists for the entire set of languages. Despite the advances of NLP in languages like English and Chinese, there exists a considerable gap in linguistic resources and tools for low-resource languages like those in the Nguni group. To bridge this gap, we applied deep neural models to the task of morphological parsing for Nguni languages. The application was in two distinct parts: neural sequence tagging models trained from scratch and pre-trained language models (PLMs) fine-tuned for the task. The sequence tagging models trained from scratch were Conditional Random Fields (CRF) and Bidirectional Long Short-Term Memory (bi-LSTM) while the PLMs chosen were XLM-RoBERTa (XLM-R), Afro-XLMR and Nguni-XLMR.

Through various experiments to assess the models' performances, the neural language models were found to be viable for morphological parsing. Both the models trained from scratch and the PLMs managed to significantly outperform the baseline finite state model, ZulMorph for both micro and macro F_1 with margins of over 30%. Conclusively, of the models developed in this project, the CRF models were the best performing of the selection by clear margins from the PLMs.

KEYWORDS

machine learning, deep learning, morphological parsing, sequence tagging, part-of-speech tagging, computational linguistics, natural language

1 INTRODUCTION

1.1 Background

Morphology is the branch of linguistics that studies the formation of words, specifically how words are constructed from subword units [35]. This brings forth the concept of the morpheme—the smallest unit of linguistic meaning that a word can be split into. Each morpheme conveys grammatical meaning in a word and, more widely, a sentence; an example being that the word "largest" can be split into the morphemes "large-" (the stem) and "-est" (a suffix indicative of the superlative form/degree). Adding such morphological information into the training of tasks like dependency parsing, machine translation, and text filtering would improve the performance since they add information about the roles of words in sentences as conveyed by morphemes. [28].

This requires annotating large-scale training sets with morphological information, which poses a significant challenge. To acquire the data, there is need for language experts to morphologically annotate the data [20] in a process that is slow and expensive. A shortage of this data limits a wide range of tasks that rely on substantial data, such as deep learning machine translation, question answering, and other NLP applications. Consequently, there is a growing need for the development and implementation of automatic parsers to compensate for this shortage and enable more effective morphological analysis. These desired tools need to morphologically process plain text via tasks like segmentation [38], parsing [1] and paradigm learning [19]. Morphological parsing is a process in which individual morphemes are labelled with their grammatical role in a word (or sentence). For example, the isiNdebele word "itjho" is split into morphemes and parsed with the respective tags into "i[SC9]-tjh[VRoot]-o[VerbTerm]". From this, we gain the information that "itjho" is a verb that can be used in agreement with a noun from the grammatical class 9. Class 9 is one of the many grammatical noun classes in the Nguni languages which conveys information about a particular set of nouns [50].

In NLP, there has been a shift to use deep neural models [24] for the modelling of NLU tasks including morphological tasks. This is a shift that has occurred fairly recently from the original non-probabilistic models used in the NLP field in its early days. In language sets like the Nguni languages, deep learning models represent a unique area for exploration because, due to their recentness, they have not been thoroughly examined in a variety of morphological tasks.

1.2 Motivation

The Nguni languages are subset of African languages consisting of isiNdebele, siSwati, isiXhosa and isiZulu [33] and they are some of the most widely spoken in South Africa. Machine learning is immensely data-driven so it works notably well for languages like English with a lot of available data. Languages like the Nguni languages are termed low-resource as they do not have abundant text resources for machine learning [34]. With this lack of data, research shows that approaches that work well for high-resource languages like English cannot be directly applied to them. Owing to data-driven approaches, there has been a significant surge in collecting data for NLP, primarily focusing on English and other major European languages. This effort has been fruitful, leading to the development of popular large language models like OpenAI's GPT [42]. Low-resource languages, on the other hand, have received very little attention. According to Tsvetkov [55], they are "languages

lacking large monolingual or parallel corpora and/or manually crafted linguistic resources sufficient for building statistical NLP applications." African languages, including Nguni languages, have seen disproportionately fewer resources devoted to them despite their large number of speakers [34]. Exploring the languages has been shown to have great social benefits in language preservation and restoration.

Another major motivator for undertaking this project was the lack of application of deep learning approaches to the specific task of morphological parsing for the Nguni languages. As mentioned earlier, the grammatical information acquired from morphological parsing is crucial for better NLU which effectively necessitates exploration of areas like deep learning for the languages. Deep learning has been applied to a sub-task of morphological parsing: morphological segmentation [38] with commendable reported results. Therefore with the advent of deep learning approaches like fine-tuning pre-trained models [15], it seemed like a novel promising path to explore on morphological parsing. In this project, we will apply it to another sub-task of morphological parsing: morphological tag prediction.

1.3 Contributions

The main aim of the project is to apply deep learning models for the task of morphological parsing for the four Nguni languages. There are few publicly available baselines so our main goal in the project is to assess and compare our models to the baseline using experimental methods. The main available baseline for this is a finite state approach, ZulMorph[1] which is publicly available through an online parser. There are two main classes of neural models to be explored in this project: using models trained from scratch [23, 32] or fine-tuning pre-trained language models (PLMs) [7, 22, 40]. The main models from these categories to be used will be discussed in detail in the following sections. They will, furthermore, be compared to each other to determine the best approach at modelling this tasks. In addition to the development, training and evaluation of the models, we will further do analysis to reveal insights into the trade-offs presented by different models, and the several underlying reasons for the success or failure of different models across the Nguni languages.

2 RELATED WORKS

The aim of this work is to bridge two prominent lines of research for the Nguni languages: neural morphological parsing and the fine-tuning of PLMs. In doing so, we hope to improve morphological parsing for the Nguni languages.

2.1 Neural Morphological Parsing

Morphological analysis, which often involves the identification of morphemes and the extraction of meaning from them by models, has been studied extensively in NLP [4]. The general trend in this line of research has followed that of similar tasks NER [39] by moving from rule-based approaches to more neural approaches. Initial approaches like "MORPH" by Chapin and Norton [6], possessed explicit hand-crafted rules for analysis and morpheme combinations which required great effort to collect. Progress was made and more morphological analysis systems became based on finite-state systems [27, 29]. Rules and language grammar could be mapped into finite state traducers which could reproduce the grammar from the languages on unseen data [4]. Finite-state models have been, until recent times, used for tackling morphological analysis even in African languages [5, 13, 44]. However, as in other applications they proved to be resource intensive and not generalising well to dissimilar languages [9, 12].

With the emergence of data-driven methods to the field of artificial intelligence, probabilistic models came into prominence with tools for analysis like Morfessor [9, 10] for morphological segmentation. However, those methods, both supervised and unsupervised, often required heavy manual feature engineering to learn the language structure well [2, 21]. The advent of neural models like recurrent neural networks and feed forward networks removed the need for manual feature extraction as these models could extract features implicitly through their architecture [4, 23].

Moeng et al. (2021) [38] tackle the parsing subtask of morphological segmentation using neural models, namely Bidirectional Long Short-Term Memory networks (bi-LSTM) [23], Conditional Random Fields (CRF) [31, 45] and Transformers using several levels of attention [56] and achieve considerable success on the task across all four Nguni languages. This gives way to approaching the task through an architecture devised by Tsarfaty el al. [54] for parsing morphologically rich languages (MRL) by segmenting them into their morphemes and tagging the separate morphemes. These architectures, however, were tried only for non-Nguni MRLs like Hungarian, Czech and Arabic.

Morphological parsing can be tackled in one of ways: one system which segments and tags simultaneously or two systems in a pipeline - one for segmentation and one for tagging the segmentations. Additionally, the segmentation part of the task can be framed as two possible types of morphological segmentation: surface segmentation and canonical segmentation [26]. Surface segmentation is when a word is split into morphemes based on character boundaries, for example, in the word "zobomi" we can split into "zo-bo-mi". This differs to canonical segmentation in which some morphemes which would have been hidden or altered for phonology and lexical reasons are uncovered, where in the same example of "zobomi" some hidden morphemes can be extracted into the segmentation "za-u-bu-omi".

2.2 Fine-tuning PLMs for Nguni NLU

As mentioned earlier, there are two classes of models that are going to be used for the task: models trained from scratch and PLMs. This section of the project focuses on the fine tuning of pre-trained language models.

2.2.1 Pre-Trained Language Models . Transfer learning [53] is a concept in machine learning whereby a model trained for one task can be fine-tuned on another dataset for another task. The model is fine-tuned to work on the second task to reduce the costs incurred when training a model from scratch for the same task. Recently, the emergence of Transformer models [56] like the Generative Pre-trained Transformer (GPT) [42] and the Bidirectional Encoder Representations from Transformers (BERT) [15], has made PLMs a

[1] https://portal.sadilar.org/FiniteState/demo/zulmorph/

staple approach to modelling language processing tasks due to the simplicity of fine-tuning.

Transformers [15, 56] are a recent (2017) sequence to sequence architecture that produces output based on contextual attention. BERT is based on transformers by leveraging context from both left and right directions to gain a better understanding of a token's context unlike unidirectional models like the GPT which acquire context solely from left to right. The basis of this project is the PLMs' pre-training task masked language modelling (MLM). In MLM [24], random words in a training text are obscured with a [MASK] token which the model has to unmask correctly [48, 51]. This essentially makes PLMs model linguistic dependencies between words occurring in the same context such as syntactic roles and related meanings [47]. After pre-training on MLM, the model can then be fine-tuned for tasks like question answering on the Stanford Question Answering Dataset (SQuAD) [43], and NER [37, 52] with empirical evidence showing it outperforming contemporaries like GPT [42], ELMo [41] and others. The operation of BERT is shown in Figure 1.

2.2.2 PLMs for Nguni Languages. With the difficulty of building language models for the Nguni languages on such specific tasks such as morphological parsing due to the languages' low resource nature [34], transfer learning [53] can be leveraged to fine-tune PLMs using limited resources. Masked language models like BERT can be improved by further pre-training to learn different languages by performing MLM on more corpora. Models like RoBERTa [7] were made through adapting larger corpora and optimising pre-training through improvements like dynamic masking. Through this derivation process various models came into existence to make PLMs more viable for Nguni languages. Cross-lingual RoBERTa (XLM-R) [8] was produced by further pre-training RoBERTa on cross-lingual modelling between corpora from 100 different languages including isiXhosa. This was built upon through Afro-XLMR [3], an adaption of XLM-R fine-tuned using a technique called multilingual adaptive fine-tuning (MAFT) on corpora from 20 African languages including isiXhosa and isiZulu. Nguni-XLMR [36] is another adaptation of XLM-R which also incorporated MAFT but using corpora made up of the four Nguni languages.

Studies from projects by the Masakhane group[2], MasakhaNER [1] and MasakhaPOS [16], used fine-tuning of PLMs as one of their experiments in their research. The masked language models they use include multilingual BERT (mBERT) [15], XLM-R [8], AfriBERTa [40], African cross-lingual RoBERTa (Afro-XLMR) [3] and AfroLM [18]. In both NER and POS, it was conclusive that fine-tuning produced better results than their other approaches like convoluted neural networks with BiLSTMs and CRFs. Furthermore, there was a discovery that AfriBERTa performed poorly on languages that were not present in the pre-training corpora. This is of particular relevance because due to their low-resource nature [34], there is minimal chance of finding PLMs pre-trained well on them. Furthermore, the PLMs XLM-R and AfroXLMR, performed significantly better than the PLMs pre-trained on NER, showing that MLM alone is enough to achieve effective results.

[2]https://www.masakhane.io/

3 METHODOLOGY

3.1 Morphological Tag Prediction

In this project we focus on the task of morphological analysis as mentioned in earlier sections. However, the project will not be involved in the full task of analysis but rather the last sub-task in the process. The sub-task to be developed upon will be the prediction of the morphological tags corresponding to the grammatical roles of the morphemes. The full process of morphological analysis is shown in Figure 2. The section of the morphological analysis being developed is the tagger which is the tag predictor assuming the words have already been segmented by the morphological segmenter. Using the earlier example of "itjho" for morphological analysis, it would first pass through the segmenter and be segmented into "i-tjh-o". Following that, our models would come in to complete the job by predicting the tags corresponding. After the tags are predicted the correctness will be evaluated as discussed in subsection 3.3. In the example since the predicted and target sequences were "SC7-VRoot-VTerm" and "SC9-VRoot-VTerm" respectively, the individual F_1 score would be calculate and added to the running averages. Our task of predicting the tags is a sequence to sequence task where it takes in a sequence of morphemes and produces a sequence of the same length of tags. While the assessment of quality of the segmentations is out of scope for the project, the segmentations for the morphological segmenter part of the pipeline are from three sources:

(1) Gold canonical segmentations from the dataset [20] which were annotated and confirmed with consultation of human experts.
(2) Predicted canonical segmentations produced from the dataset by Moeng et al.[38]'s morphological segmenters.
(3) Predicted canonical segmentations produced from the dataset by Moeng et al.[38]'s morphological segmenters.

The latter two sources of segmentations are not perfect as none of their accuracy scores were close to 100% when evaluated on dataset. This was a source of a problem called misalignment in the project in which the segmenters produced segmented morpheme sequences of differing lengths to the expected sequence length. With the same example, if the word "zobomi" was surface segmented into "zo-bomi", our models would only predict three tags whereas the expected output has four tags. This will again be further discussed in the Evaluation section.

3.2 Models

Out of the possible PLMs, only three were chosen due to their languages available in their pre-training language sets. The models are as follows:

(1) XLM-R by Conneau et al. [8]: a large scale cross-lingual PLM trained on more than 100 languages including isiXhosa.[3].
(2) Afro-XLMR by Alabi et al. [3]: XLM-R further pre-trained on 20 African languages including isiXhosa and isiZulu[4].
(3) Nguni-XLMR by Meyer et al. [36]: XLM-R adapted for the 4 Nguni languages.[5].

[3]https://huggingface.co/FacebookAI/xlm-roberta-large
[4]https://huggingface.co/Davlan/afro-xlmr-large-76L
[5]https://huggingface.co/francois-meyer/nguni-xlmr-large

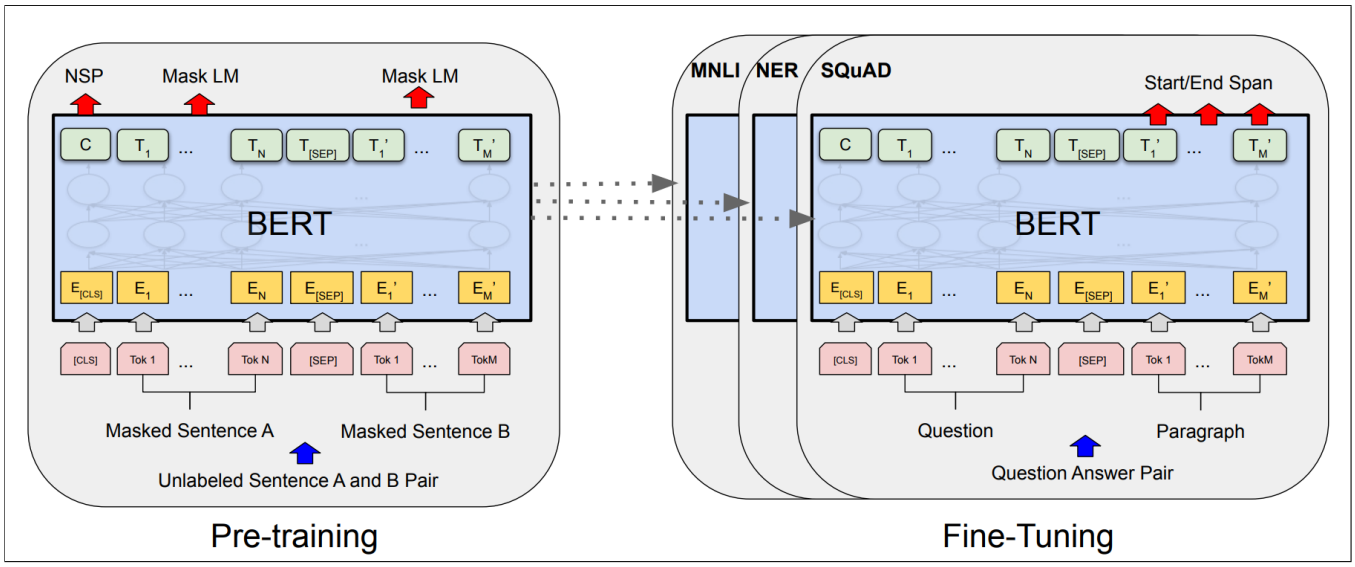

Figure 1: The operation of BERT, illustrating how the model is first pre-trained as an MLM and for next sentence prediction, then later fine-tuning it for downstream tasks like SQuAD and NER. The pre-training gets input as masked sentence pairs and in subsequent fine-tuning the input can be changed to question-answer pairs, e.t.c., depending on the downstream task [15].

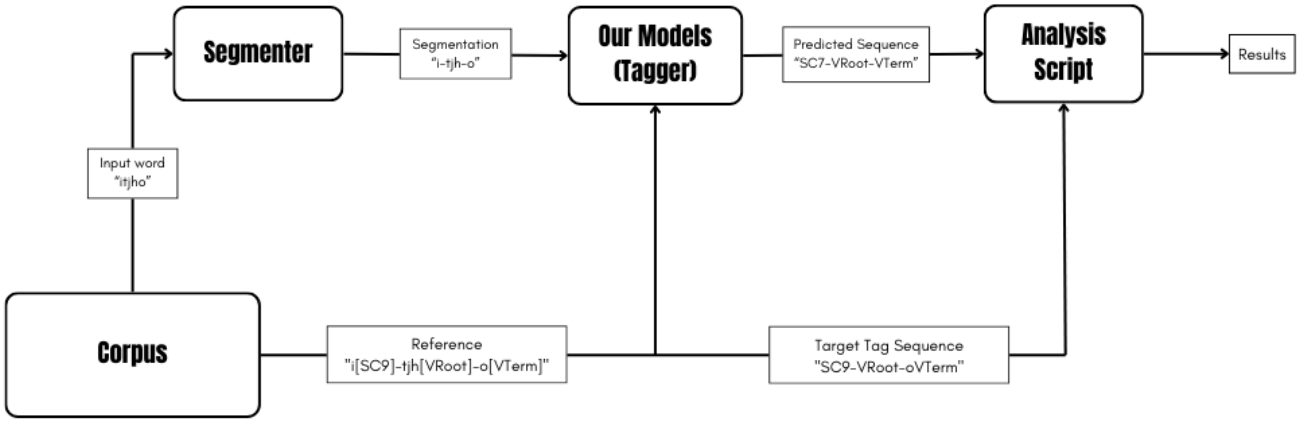

Figure 2: Full morphological analysis pipeline showing morphological segmentation and morphological tagging as the two composite and sequential sub-task. The tagger is what is being developed in the project as a tag predictor/parser for morphemes produced by the segmenter. At each step there is the example of "itjho" provided to illustrate what would happen at each step in the process.

The models chosen for this project were selected for their inclusion of African languages and more specifically Nguni languages in their pre-trained corpora. The project is interested in the varying degrees of Nguni language adaption as an influence on the transfer learning performance. This is especially the case for Nguni-XLMR which was further pre-trained for Nguni natural language understanding and generation tasks like POS tagging, NER and machine translation [36].

3.3 Evaluation

3.3.1 Metrics. Each of the models was evaluated to assess performance on three metrics: precision, recall and F_1. Precision is the

proportion of positive identifications that were correctly identified [14]. In the case of morphological parsing, this is the number of correctly identified tags in proportion to all the tags in the predicted output. Recall is the proportion of the correctly identified tags to all the tags in the expected output [14]. Both precision and recall can mathematically be described in terms of true and false positives and negatives (TP, FP, TN, and FN), as shown in Equation 1 and Equation 2.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

F_1 is the harmonic mean of precision and recall given by Equation 3 [11, 49]. F_1 balances the selectivity of the predictions through precision and the coverage of the predictions through recall. This essentially helps in finding insight for imbalanced data since it focuses on both false positives and false negatives.

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

A higher score for each metric would signify improved performance according to that metric, allowing comparison of each one's relative parsing skill. F_1 was calculated on a word basis using the *scikit-learn* metrics library and then the results were aggregated for each of the unique tags in the tag set to get an average. For the project, we report F_1 for the whole test set average in two ways. Micro F_1 is weighted by tag class frequency while macro F_1 weighs the performance for each tag individually. For example, in the isiNdebele test set, the AuxVStem class has a support of 6 examples while the NStem class has a support of 2943. Macro F_1 would be more punishing if the model predicted on three AuxVStem morphemes since it would be counted as a 50% despite it having few examples, while micro F_1 would be more lenient and just count the three in a total of all correct predictions. The rare tags like AuxVStem would be harder for the models to learn due to their sparsity in data therefore optimising macro F_1 would be harder for the model.

3.3.2 Sequence Alignment. The metrics discussed only work well and are explainable for target and prediction sequences of equal lengths. As mentioned in subsection 3.1, this is only guaranteed when the models are tested on the expert confirmed segmentations from the dataset. For the predicted segmentations, however, the issue of misalignment arises. The two types of misalignment are under-segmentation, in which the segmenter predicts fewer morphemes than are there and over-segmentation which is the opposite phenomenon. An alignment algorithm was created to equalise the lengths of the predicted and target tag sequences by padding the shorter with a pad token <?>. The padding was inserted according to an algorithm we developed which maximises the F_1 score on the single example. For example, if there were two sequences "A-D" and "A-B-C-D-E" the alignment would pad the shorter sequence to "A-?-?-D-?" to maximise matching on the tags "A" and "D". The algorithm is shown in the Appendix B.

Word	Morphological analysis
acebisayo	a[RelConc6]-cebis[VRoot]-a[VerbTerm]-yo[RelSuf]
kwibhunga	ku[LocPre]-i[NPrePre5]-(li)[BPre5]-bhunga[NStem]
izincomo	i[NPrePre10]-zin[BPre10]-como[NStem]

Table 1: Three examples from the isiXhosa part of the dataset as collected by Gaustad and Puttkammer [20].

4 EXPERIMENTS

4.1 Dataset

The dataset used in this project was developed by Gaustad and Puttkammer [20]. It consists of sentences from government records on a word level where each row is a word with its morphologically parsed canonical segmentation, its lemma and its part of speech. The first two columns are the relevant ones for this project as seen with the examples from the isiXhosa train set in Table 1. There were also sentence separation tags in the dataset to demarcate different sentences which were discarded as all the PLMs were trained as word-level models and not sentence-level.

The data was pre-processed to split the canonical segments from their tags as separate input and output, for example, i[NPrePre10]-zin[BPre10]-como[NStem] was separated into i-zin-como and NPrePre10-BPre10-NStem as a pair. A custom dataset class was created to accept the train and test dataset and tokenize them using XLM RoBERTa tokenizers from the Hugging Face library with a custom tokenization script. Since there was no outlined validation set, one was sampled as 10% of the training set using a seed of 42.

4.2 Hyperparameter Tuning

The first objective of the project was to find the best hyperparameter settings for fine-tuning models on each of the languages and therefore there was need to perform grid search for the hyperparameters. For this training to be possible and feasible in the runtime of the project, there was need of GPUs to perform the training cycles of the model. This processing power was requested and acquired from two separate high-performance computing (HPC) clusters: the National Integrated Cyberinfrastructure System's Centre for High-Performance Computing (NICIS CHPC)[6] and the University of Cape Town's HPC cluster[7]. The primary cluster was the NICIS CHPC using a single Nvidia V100 GPU at a time and the training was carried off successfully without need of the backup.

4.2.1 Testing. For each of the four Nguni languages, each of the three PLMs' key hyperparameters were acquired through a grid search methodology explained in the following section. This led to twelve fine-tuned model hyperparameter sets being produced, for which the PLMs were run with evaluation on the test set to acquire the conclusive results for this project's objectives. Due to the variability that can occur in the initialisation of neural models, it was possible to get results with large variance on the macro F_1 score. To offset this issue, five random seeds were selected for training evaluation and then for each evaluation metric an average of five was reported for statistical completeness. The chosen seeds were integers 1 through 5 for reproducibility.

[6] https://wiki.chpc.ac.za/chpc:gpu
[7] https://ucthpc.uct.ac.za/index.php/hpc-cluster/

param	value 1	value 2	value 3
lr	$1e^{-5}$	$3e^{-5}$	$5e^{-5}$
epochs	5	10	15
batch_size	8	16	32

Table 2: Values experimented on in the grid search for the models

As mentioned earlier, there are three qualities of segmentations that are going to be considered and three separate result sets will be produced from these:

(1) Training on the gold[8] canonical segmentations and evaluating on the gold canonical segmentations.
(2) Training on the gold canonical segmentations and evaluating on the predicted canonical segmentations.
(3) Training on the imperfect surface segmentations and evaluating on the predicted surface segmentations.

The first option only achieves half of the pipeline as it assumes we always have human annotated segmentations which is a fallible assumption. The last two sets are end-to-end models as shown in the pipeline, Figure 2, as they involved morphological segmentation and tags produced by neural models.

4.2.2 Grid Search. The key hyperparameters being being searched on were fine-tuning batch size, number of training epochs and the initial model learning rate. The values are shown in Table 2

Of the nine hyperparameter options, only the batch size of 8 was shown perform significantly less in early training runs and was dropped from the search space. With the four languages, three models and eight hyperparameters, there were 216 configurations to run and evaluate to pick the best models.

Out of the 216 configurations only twelve were chosen for the final selection for the experiments on the test set. The metrics were evaluated on the validation set (10% of the training set) and the selection was made on the models' macro F_1 score.

4.2.3 Grid Search Results. As mentioned in the experimentation description, a grid search was applied to the model on the selected hyperparameters in Table 2. The results from the grid search are shown in Table 3.

The grid search results revealed that the best performance was achieved within 10 epochs of training for all the PLMs. Afro-XLMR and Nguni-XLMR consistently outperforming XLM-R across all four languages. This performance can be attributed to the pre-training of these models on African languages with similar morphologies, especially for Nguni-XLMR [1, 53, 54]. The best results for isiXhosa and isiZulu came from Afro-XLMR while Nguni-XLMR performed best for isiNdebele and siSwati. The search for overall best batch size was inconclusive with both 16 and 32 yielding competitive results. The grid search, although limited by computational constraints, showed that the optimal hyperparameters for fine-tuning - 10 epochs, a 5E-5 learning rate and a batch size of 16 - were generally effective across languages for all models. The minor limitations of the grid are listed briefly in Appendix A.5.

[8]The "gold" standard for task is human annotation of morphological segmentations.

Language	Model	epochs	lr	batch	F1
isiNdebele	XLMR	10	$5e^{-5}$	32	0.7412
	Afro-XLMR	5	$3e^{-5}$	16	0.7405
	Nguni-XLMR*	5	$5e^{-5}$	32	**0.7455**
siSwati	XLMR	15	$5e^{-5}$	16	0.7248
	Afro-XLMR	15	$5e^{-5}$	16	0.7087
	Nguni-XLMR*	10	$5e^{-5}$	16	**0.7309**
isiXhosa	XLMR*	5	$5e^{-5}$	16	0.7391
	Afro-XLMR*	10	$3e^{-5}$	32	**0.7516**
	Nguni-XLMR*	10	$1e^{-5}$	16	0.7432
isiZulu	XLMR	10	$3e^{-5}$	32	0.6912
	Afro-XLMR*	10	$3e^{-5}$	16	**0.6981**
	Nguni-XLMR*	10	$5e^{-5}$	32	0.6803

Table 3: Results of the grid search showing the best version of each model in each of the Nguni languages. The F_1 score reported is the macro-F_1 average. Asterisk, *, inserted when the language was present in the pre-training of the model. The bold results are the best model per language indicating the bet hyperparameters. The underlined result is the best model performance in the grid search irrespective of language or model

5 RESULTS AND DISCUSSION

After acquiring the best models from the results of the grid search the averages of five training runs with different random seeds were acquired and are displayed in Table 4 and Table 5 for the gold canonical segmentations and prediction-based model segmentations respectively. Table 4 shows the data for half pipeline (tag prediction after segmentation) which works with the expert annotated canonical segmentations and tags acquired from the dataset [20]. Table 5 shows the data for the full morphological parsing pipeline (word to segmentations to parsing) for both canonical and surface segmentations. For the surface segmentations on the models trained from scratch, initial experiments showed the sentence-level models comfortably outperformed word-level models so we focused only on sentence-level models going forward.

5.1 Comparing Neural Parsing to Rule-Based Parsing

Due to limited data, the only available baseline was the ZulMorph segmenter-parser end-to-end model. It was demonstrably clear that, for the Zulu language, our neural models performed significantly better than the ZulMorph finite state approach. The model trained from scratch outperformed ZulMorph by more than double its macro-F_1 score. This shows how much more generalised the performance of neural models is compared to the finite state model. It is important to note that the test set that was used to evaluate the baseline and the models was from a government document domain [20] where the models were also trained from. However, with such a large gap in performance, the domain-specificity is not enough as justification meaning our models are better than the rule-based approaches for parsing. Furthermore, while the performance of the baseline, ZulMorph, is commendable it cannot compare to the

Model	isiNdebele		siSwati		isiXhosa		isiZulu	
	Micro F_1	Macro F_1	Micro F_1	Macro F_1	Micro F_1	Macro F_1	Micro F_1	Macro F_1
Baselines								
ZulMorph [5]	-	-	-	-	-	-	0.6471	0.3378
Word-level								
Bi-LSTM, morpheme	0.9222	0.6950	0.9160	0.6761	0.9529	0.7359	0.9264	**0.6860**
Bi-LSTM, char-sum	**0.9226**	0.6907	0.9177	0.6835	0.9553	0.7418	0.9271	0.6817
Sentence-level								
Bi-LSTM, morpheme	0.9188	0.7009	0.9190	0.6872	0.9609	0.7694	0.9263	0.6812
Bi-LSTM, char-sum	0.9142	0.6901	0.9132	0.6748	0.9585	0.7604	0.9210	0.6661
CRF, morpheme	0.9189	**0.7047**	**0.9196**	**0.6945**	0.9619	**0.7777**	**0.9272**	0.6825
CRF, char-sum	0.9167	0.7007	0.9179	0.6855	**0.9623**	0.7633	0.9255	0.6730
XLM-RoBERTa	**0.9152**	**0.6425**	0.9095	0.6420	0.9467	0.6773	0.9132	0.6157
Afro-XLMR	0.9133	0.6273	**0.9100**	**0.6460**	**0.9583**	**0.7363**	**0.9282**	**0.6610**
Nguni-XLMR	0.9104	0.6190	0.9042	0.6176	0.9488	0.6738	0.9187	0.6302

Table 4: Results for Morphological Parsing on the four Nguni languages making use of different deep learning models over the gold data evaluation. Results are the averages of five for over different seeds for each model language variant. Bolded represents the best in the model group (pre-trained or from scratch) and underlined represents the best overall

Model	isiNdebele		siSwati		IsiXhosa		isiZulu	
	Micro F_1	Macro F_1	Micro F_1	Macro F_1	Micro F_1	Macro F_1	Micro F_1	Macro F_1
Baselines								
ZulMorph [5]	-	-	-	-	-	-	0.6471	0.3378
Canonical segmentations on data from Moeng et al. [38]								
Word-level								
Bi-LSTM, morpheme	0.8084	0.5769	0.8230	0.5717	0.9110	0.6807	0.8250	0.5936
Bi-LSTM, char-sum	**0.8094**	**0.6816****	0.8289	0.5760	0.9107	0.6816	0.8248	**0.6033****
Sentence-level								
Bi-LSTM, morpheme	0.8059	0.5834	0.8274	0.5792	0.9172	0.7171	0.8265	0.5989
Bi-LSTM, char-sum	0.8010	0.5814	0.8238	0.5749	0.9135	0.7020	0.8184	0.5903
CRF, morpheme	0.8049	0.5961	**0.8301****	**0.5860****	**0.9190****	**0.7224****	**0.8280****	0.5991
CRF, char-sum	0.8081	0.5893	0.8271	0.5782	0.9181	0.7084	0.8246	0.5973
XLM-RoBERTa	**0.8151****	**0.5509**	**0.8280**	0.5278	0.9137	0.6346	0.8251	0.5438
Afro-XLMR	0.8137	0.5413	0.8273	**0.5296**	**0.9140**	**0.6423**	0.8269	0.5469
Nguni-XLMR	0.8144	0.5468	0.8264	0.5285	0.9155	0.6390	**0.8272**	**0.5495**
Surface segmentations on data from Moeng et al. [38]								
Sentence-level								
Bi-LSTM, morpheme	**0.7830**	**0.5409**	**0.8129**	**0.5274**	**0.8661**	**0.6724**	**0.8023**	**0.5529**
Bi-LSTM, char-sum	0.7742	0.5238	0.8053	0.5212	0.7994	0.6054	0.7976	0.5511
XLM-RoBERTa	**0.7282**	**0.4868**	0.5107	0.2231	0.7244	0.5208	0.6759	0.4349
Afro-XLMR	0.7275	0.4832	0.5216	0.2412	0.7273	**0.5300**	0.6794	0.4495
Nguni-XLMR	0.7258	0.4746	**0.5328**	**0.2505**	**0.7309**	0.5270	**0.6818**	**0.4512**

Table 5: End-to-End Results for Morphological Parsing on the four Nguni languages making use of data produced from segmenters by Moeng et al. [38] for both canonical and surface segmentations. Bolded indicates the best models in each category (pre-trained and from) for each language column within the segmentation type. Underlined indicates the best model in the segmentation type for each language. ** indicates the best model for end-to-end performance irrespective of segmentation type for each language.

learning capacity of neural models which can learn morphological rules without need for prescribed state transitions. The live demonstration from which the ZulMorph tag predictions also produced multiple sets of tags per word as opposed to our models which predicted only one as the most probable based on trained probabilities.

5.2 Comparing fine-tuning to training from scratch

From the results shown in Table 4 for the models on the given dataset, it is evident that the models trained from scratch were superior to the PLMs. The only metric in which the pre-trained models are superior to training from scratch is the micro-F_1 score for the isiZulu set. This is contrary to what was hypothesised at the beginning of the experiment in which it was thought that due to their size, robustness and pre-learned contextual embeddings [1, 8, 22], the pre-trained models would model the task significantly better than models trained from scratch. Combined with the results from the grid search, Table 3, an observation may be made that the models slightly over-fitted to the training data during fine-tuning leading to less generalised performance on testing data. Aside from the models in general, some of the model hyperparameters could have possibly overfit the validation set and optimised for rewards on that set instead of the languages in general. Nguni-XLMR is a good example of this as it performed the best for isiNdebele and siSwati on the evaluation dataset but eventually became the worst model for the task on the generalised test set.

5.2.1 Gold Canonical Segmentations. As discussed above, the models trained from scratch outperformed PLMs for all levels of segmentation quality. The best performance overall was the sentence-level CRF with morpheme tokenization showing best performance for both micro and macro F_1. This performance could be attributed to CRFs' excelling ability to explicitly model the dependencies between the output labels in a sequence [25]. The extended context in the canonical sentences also helped the model to achieve this performance. Although the sequence length (word or sentence) could explain the performance gap it is not enough to show why models trained from scratch are better than PLMs at the task. When both approaches are examined at the word level, the models trained from scratch were usually better by at least 2%. Since the models were provided with the best annotations for training and evaluation, the results in Table 5 are the point of reference for how well neural models can perform for morphological parsing.

5.2.2 Predicted Canonical Segmentations. The results for the section are in the middle third of Table 5 and were from training the models of the gold annotated data and evaluating on predicted canonical segmentations. As was expected there is a decline in performance and everything is relatively the same from the results of the gold annotation evaluation in Table 4. The models trained from scratch follow closer with an average macro F_1 deviation from the gold annotation performance of -6.72% while for the PLMs it was -11.58%. This shows that while they both struggled to model the tags for the predicted canonical morphemes, the models trained from scratch generalised significantly better. This is especially true for the rare tag classes as seen by the great drop in the PLMs' performance on macro F_1.

5.2.3 Predicted Surface Segmentations. For the surface segmentations, the models were both trained and evaluated on the predicted surface segmented morpheme and tag sets. The trend in decreasing performance continued from the previous two subsections as this section saw the worst performance of the models on the task of parsing. It is worth noting, though, that despite this bad performance by the standard of the neural language models, the isiZulu models performed comfortably above the ZulMorph baseline of 33.78% macro F_1. The outlier in this result set is for the PLMs on the siSwati dataset. While the other languages were around -9% from the canonical end-to-end PLMs, siSwati dropped with a difference of 27.69%. A possible reason for this is the quality of the morphological segmentation. Moeng et al. [38] report siSwati as having the worst surface morphological segmentation predictions. However, that performance is only slightly worse than for the other languages in their project and would not be enough to justify such a significant drop in performance. More investigation would have to be performed to analyse the specific reason for the PLMs poor performance on parsing surface segmentations. Overall as a class of models, the poor performance can be attributed to the words being split over superficial surface level boundaries which are not representative of the underlying canonical morphemes. The majority of this problem arises from under-segmentations which lose more than one word in the process. For example, in the isiXhosa dataset for "kubomi", the correct canonical segmentation is "ku-u-bu-omi" but the surface segmenter produced "ku-bomi" which in turn nullifies two meaningful morphemes. This performance automatically reduces the F_1 score to have a max of 50% on that word.

5.2.4 End-to-End Morphological Parsing. The results in both Tables 4 and 5 illustrate that the neural models are quite feasible with considerable performance on the task of morphological segmentation. Even with the introduction of prediction-based morphological segmentations, the performance tracked very well for the models trained from scratch with little deviation between the gold annotated and predicted segmentations. This is especially true for the CRFs while more work has to be done for fine-tuning the PLMs to the task. The deviation of about 7% confirms feasibility with existing canonical morphological segmenters even if there are no experts to segment the data for training.

5.3 Subword tokenisation

Another possible source of performance loss for the PLMs is due to tokenization of the words for the task. XLM-RoBERTa, the base model for all the PLMs in this project uses a SentencePiece subword tokenizer [8, 30] meaning that the morphemes which are subword by definition [35] are further split into tokens. For example, a morpheme "-bandela" can be further split into "-ba, #ndel #a" when the morpheme itself is a complete subword unit. By the nature of the Nguni languages being agglutinating [1, 33], the sub-morpheme tokenization was probably not a good strategy to tokenize the input morphemes as transfer learning is known to be more difficult for languages with agglutinating morphologies [36, 57]. This is usually due to the misalignment between what the tokenizers view as

good learnable subwords and the actually morphemes that exist in the word which would convey actual meaning. This is particularly detrimental to morphologically complex languages as the model will essentially be limited in understanding the specific combinations of morphemes if the words are split on non-existent subword boundaries. This view is counter-intuitive to the view that SentencePiece would provide better cross-lingual transfer [46] but further emphasises the fact tokenizer choice for such fine-grained tasks is important [17]. This sentiment is echoed greatly in the results of the models trained from scratch where at sentence-level, the morpheme tokenized models are the best performers for the 3 of the 4 languages. Üstün et al. [58] also show that while both character-level representations and morpheme-level representations are better than word-level for morphological languages, morpheme-level representations are significantly better since they incorporate crucial semantic and syntactic information that character-level representations would otherwise miss.

5.4 Lexical Analysis

Judging internally in the model classes, pre-trained and from scratch, it is noticed that there is very little variance in the macro-F_1 scores with the performance varying within 3% of each other except for the Afro-XLMR for isiXhosa for which it performs significantly above the other PLMs at 75.63%. This great yet anomalous performance extends to all the other models on the isiXhosa dataset as it proved to be the best performing language throughout the irrespective of the model type. This performance suggests that there is a property of isiXhosa that makes it more straightforward for the models to learn as compared to the other three languages. A simple lexical analysis of the dataset (both train and test sets) was conducted to discover which properties of the language differentiated it from the other languages and is shown in Figure 3 and Table 6. The most glaring difference between isiXhosa and the other languages it the number of unique morphemes it possesses. It only has has 2453 unique morphemes compare to languages which have upwards an additional 800 morphemes; isiNdebele even has more than double the number of morphemes. This would obviously make it easier to learn than the other languages since there are less morphemes for the models to learn hence less contexts as shown by its performance in Table 4. Aside from the number of unique morphemes, isiXhosa also possesses the smallest tag set of the four languages which gives it a smaller prediction space and makes its evaluation on F_1 averages more forgiving.

There is a clear relationship between the number of morphemes per words with the best model performance when it comes to the PLMs. The higher the value, the better the performance but without more analysis this could just can just be viewed as a correlation instead of a causality. It is further confirmed that this relationship is not causal since the models from scratch do not have this correlation between average morphemes per word and macro F_1. Despite the relationship between morphemes per word and macro F_1 not being fully confirmed, we noticed that the ratio decreases between the gold annotated data and the surface segmented data which reinforces the suggested inverse proportionality between the two variables.

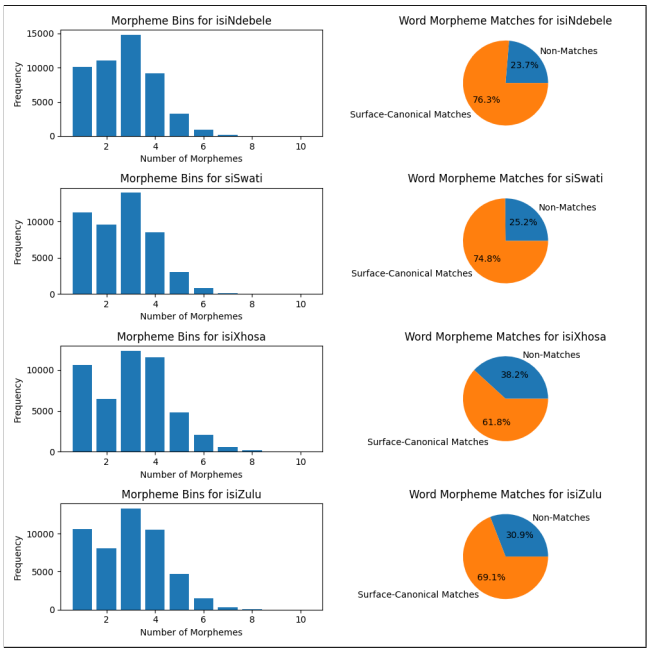

Figure 3: Number of morphemes per word and the proportions of words whose surface segmentation is the same as the canonical segmentation for all the four languages.

A property that is, however, worthy of note is the number of words in the vocabulary whose surface segmentation matches the canonical segmentation. An observation is that isiXhosa has less words whose canonical segmentations match their surface segmentations than the rest of the languages as shown in the pie charts in Figure 3. This can be indicative of the transformer-based PLMs' ability to learn the hidden morphemes better from context than the plain surface morphemes. As already shown in the results of the project, the models do better on canonical segmentation than surface segmentation and this matches the analysis performed on the words on the dataset. isiXhosa, which has more hidden morphemes performs better on average than the other languages and that attribute is possibly key in improving the performance of the PLMs in the long run.

6 CONCLUSION

The project explored the applicability of deep neural models on the task of morphological parsing for Nguni languages. While the performance is not perfect and ranging between 50% and 78% macro F_1, the models created and fine-tuned in this project have shown that deep neural models are indeed applicable to this task in comparison to existing rule-based tools like ZulMorph. The main takeaways are than the Conditional Random Field models are the best performing models trained from scratch while Afro-XLMR was best among the PLMs. All the results show promise of improvement with bigger datasets and more focused fine-tuning for the PLMs. Both approaches to applying neural models to morphological parsing have proven applicable within close margins of each other on expertly annotated data. We have additionally discovered that end-to-end

	isiNdebele	siSwati	isiXhosa	isiZulu
Word count	49689	47385	48735	49097
Gold Annotated Dataset				
Morpheme count	137400	127698	149294	144047
Morphemes/word	2.77	2.69	3.06	2.93
Unique morphemes	5100	3389	2453	3284
Unique tags	240	246	236	256
Surface Segmented Dataset				
Morpheme count	131204	125282	133244	133476
Morphemes/word	2.64	2.64	2.73	2.72
Unique morphemes	5843	4932	3762	4240
Unique tags	252	342	340	350

Table 6: Various lexical properties of the languages present in the canonical segmentation annotated dataset by Gaustad and Puttkammer [20] and the surface segmentation equivalent produced by Moeng et al. [38]'s segmenter aggregated for both the test set and the train set.

morphological parsing models are feasible with good performance through the use of existing canonical segmenters. Another key takeaway was the linguistic compactness of the isiXhosa data leading to its great performance as compared to all the other Nguni languages. The experiments from this project have also shown that canonical segmentations are more suited to the task of morphological parsing than surface segmentations despite the initial hypothesised performance of the two segmentation types.

It is the hope that future research will improve on the results of the project through various avenues. These would include using better morpheme-level tokenizers for the PLMs instead of subword tokenizers and doing a more thorough grid search to improve the results. There are also other PLMs pre-trained on African languages whose performance could be assessed to make more informed analyses on the applicability of PLMs to morphological parsing. Other improvements to the project would be to make models better suited to the morphological differences between the languages instead of generalising for the entire Nguni language set.

In conclusion, this project has produced multiple neural morphological parsers for the Nguni languages which outperform the existing finite-state models on relevant metric, F_1 score.

7 ACKNOWLEDGEMENTS

I would like to thank Francois Meyer for continuous guidance and instruction throughout the entirety of the project. I would also like to thank my project partner, Cael, for sharing in this passion for computational linguistics and bringing insightful contributions to weekly project discussions. Much thanks extended to family and friends, especially my mother for the constant check-ins and prayers.

REFERENCES

[1] David Ifeoluwa Adelani, Jade Abbott, Graham Neubig, Daniel D'souza, Julia Kreutzer, Constantine Lignos, Chester Palen-Michel, Happy Buzaaba, Shruti Rijhwani, Sebastian Ruder, Stephen Mayhew, Israel Abebe Azime, Shamsuddeen Muhammad, Chris Chinenye Emezue, Joyce Nakatumba-Nabende, Perez Ogayo, Anuoluwapo Aremu, Catherine Gitau, Derguene Mbaye, Jesujoba Alabi, Seid Muhie Yimam, Tajuddeen Gwadabe, Ignatius Ezeani, Rubungo Andre Niyongabo, Jonathan Mukiibi, Verrah Otiende, Iroro Orife, Davis David, Samba Ngom, Tosin Adewumi, Paul Rayson, Mofetoluwa Adeyemi, Gerald Muriuki, Emmanuel Anebi, Chiamaka Chukwuneke, Nkiruka Odu, Eric Peter Wairagala, Samuel Oyerinde, Clemencia Siro, Tobius Saul Bateesa, Temilola Oloyede, Yvonne Wambui, Victor Akinode, Deborah Nabagereka, Maurice Katusiime, Ayodele Awokoya, Mouhamadane MBOUP, Dibora Gebreyohannes, Henok Tilaye, Kelechi Nwaike, Degaga Wolde, Abdoulaye Faye, Blessing Sibanda, Orevaoghene Ahia, Bonaventure F. P. Dossou, Kelechi Ogueji, Thierno Ibrahima DIOP, Abdoulaye Diallo, Adewale Akinfaderin, Tendai Marengereke, and Salomey Osei. 2021. MasakhaNER: Named Entity Recognition for African Languages. arXiv:2103.11811 [cs.CL]

[2] Koray Ak and Olcay Taner Yildiz. 2011. Unsupervised morphological analysis using tries. In *Computer and Information Sciences II: 26th International Symposium on Computer and Information Sciences*. Springer, 69–75.

[3] Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. 2022. Adapting Pre-trained Language Models to African Languages via Multilingual Adaptive Fine-Tuning. In *Proceedings of the 29th International Conference on Computational Linguistics*, Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na (Eds.). International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 4336–4349. https://aclanthology.org/2022.coling-1.382

[4] Jatayu Baxi and Brijesh Bhatt. 2024. Recent advancements in computational morphology : A comprehensive survey. arXiv:2406.05424 [cs.CL] https://arxiv.org/abs/2406.05424

[5] Sonja Bosch, Laurette Pretorius, Kholisa Podile, and Axel Fleisch. 2008. Experimental Fast-Tracking of Morphological Analysers for Nguni Languages. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, and Daniel Tapias (Eds.). European Language Resources Association (ELRA), Marrakech, Morocco. http://www.lrec-conf.org/proceedings/lrec2008/pdf/643_paper.pdf

[6] Paul G Chapin and Lewis M Norton. 1968. A Procedure for Morphological Analysis. (1968).

[7] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.). Association for Computational Linguistics, Online, 8440–8451. https://doi.org/10.18653/v1/2020.acl-main.747

[8] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.). Association for Computational Linguistics, Online, 8440–8451. https://doi.org/10.18653/v1/2020.acl-main.747

[9] Mathias Creutz and Krista Lagus. 2002. Unsupervised Discovery of Morphemes. In *Proceedings of the Workshop on Morphological and Phonological Learning of ACL-02*. The Association for Computational Linguistics, United States, 21–30. 10 pages, Proceedings of Morphological and Phonological Learning Workshop of ACL'02 ; Workshop on Morphological and Phonological Learning of ACL-02 ; Conference date: 07-07-2002 Through 10-07-2002.

[10] Mathias Creutz and Krista Lagus. 2005. *Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0.* Helsinki University of Technology, Finland.

[11] Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Trans. Speech Lang. Process.* 4, 1, Article 3 (feb 2007), 34 pages. https://doi.org/10.1145/1187415.1187418

[12] Sajib Dasgupta and Vincent Ng. 2006. Unsupervised morphological parsing of Bengali. *Language Resources and Evaluation* 40, 3 (01 Dec 2006), 311–330. https://doi.org/10.1007/s10579-007-9031-y

[13] Turhan Daybelge and Ilyas Çiçekli. 2007. A Rule-Based Morphological Disambiguator for Turkish. https://api.semanticscholar.org/CorpusID:13220962

[14] Google Developers. 2022. Precision and Recall. https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall. Accessed: [13/04/2024].

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL]

[16] Cheikh M. Bamba Dione, David Ifeoluwa Adelani, Peter Nabende, Jesujoba Alabi, Thapelo Sindane, Happy Buzaaba, Shamsuddeen Hassan Muhammad, Chris Chinenye Emezue, Perez Ogayo, Anuoluwapo Aremu, Catherine Gitau,

Derguene Mbaye, Jonathan Mukiibi, Blessing Sibanda, Bonaventure F. P. Dossou, Andiswa Bukula, Rooweither Mabuya, Allahsera Auguste Tapo, Edwin Munkoh-Buabeng, Victoire Memdjokam Koagne, Fatoumata Ouoba Kabore, Amelia Taylor, Godson Kalipe, Tebogo Macucwa, Vukosi Marivate, Tajuddeen Gwadabe, Mboning Tchiaze Elvis, Ikechukwu Onyenwe, Gratien Atindogbe, Tolulope Adelani, Idris Akinade, Olanrewaju Samuel, Marien Nahimana, Théogène Musabeyezu, Emile Niyomutabazi, Ester Chimhenga, Kudzai Gotosa, Patrick Mizha, Apelete Agbolo, Seydou Traore, Chinedu Uchechukwu, Aliyu Yusuf, Muhammad Abdullahi, and Dietrich Klakow. 2023. MasakhaPOS: Part-of-Speech Tagging for Typologically Diverse African languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 10883–10900. https://doi.org/10.18653/v1/2023.acl-long.609

[17] Miguel Domingo, Mercedes García-Martínez, Alexandre Helle, Francisco Casacuberta, and Manuel Herranz. 2018. How Much Does Tokenization Affect Neural Machine Translation? *CoRR* abs/1812.08621 (2018). arXiv:1812.08621 http://arxiv.org/abs/1812.08621

[18] Bonaventure F. P. Dossou, Atnafu Lambebo Tonja, Oreen Yousuf, Salomey Osei, Abigail Oppong, Iyanuoluwa Shode, Oluwabusayo Olufunke Awoyomi, and Chris Chinenye Emezue. 2022. AfroLM: A Self-Active Learning-based Multilingual Pretrained Language Model for 23 African Languages. arXiv:2211.03263 [cs.CL]

[19] Greg Durrett and John DeNero. 2013. Supervised Learning of Complete Morphological Paradigms. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff (Eds.). Association for Computational Linguistics, Atlanta, Georgia, 1185–1195. https://aclanthology.org/N13-1138

[20] Tanja Gaustad and Martin J. Puttkammer. 2022. Linguistically annotated dataset for four official South African languages with a conjunctive orthography: IsiNdebele, isiXhosa, isiZulu, and Siswati. *Data in Brief* 41 (2022), 107994. https://doi.org/10.1016/j.dib.2022.107994

[21] John Goldsmith. 2000. Linguistica: An automatic morphological analyzer. In *Proceedings of 36th meeting of the Chicago Linguistic Society*. Citeseer.

[22] Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, Wentao Han, Minlie Huang, Qin Jin, Yanyan Lan, Yang Liu, Zhiyuan Liu, Zhiwu Lu, Xipeng Qiu, Ruihua Song, Jie Tang, Ji-Rong Wen, Jinhui Yuan, Wayne Xin Zhao, and Jun Zhu. 2021. Pre-trained models: Past, present and future. *AI Open* 2 (2021), 225–250. https://doi.org/10.1016/j.aiopen.2021.08.002

[23] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (11 1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735 arXiv:https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf

[24] Rani Horev. 2018. BERT Explained: State of the art language model for NLP. https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

[25] Bo Huang, Jiahao Zhang, Jiaji Ju, Ruyan Guo, Hamido Fujita, and Jin Liu. 2023. CRF-GCN: An effective syntactic dependency model for aspect-level sentiment analysis. *Knowledge-Based Systems* 260 (2023), 110125. https://doi.org/10.1016/j.knosys.2022.110125

[26] Katharina Kann, Ryan Cotterell, and Hinrich Schütze. 2016. Neural Morphological Analysis: Encoding-Decoding Canonical Segments. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Jian Su, Kevin Duh, and Xavier Carreras (Eds.). Association for Computational Linguistics, Austin, Texas, 961–967. https://doi.org/10.18653/v1/D16-1097

[27] Lauri Karttunen, Ronald M Kaplan, and Annie Zaenen. 1992. Two-level morphology with composition. In *COLING 1992 Volume 1: The 14th international conference on computational linguistics*.

[28] Matej Klemen, Luka Krsnik, and Marko Robnik-Šikonja. 2023. Enhancing deep neural networks with morphological information. *Natural Language Engineering* 29, 2 (2023), 360–385. https://doi.org/10.1017/S1351324922000080

[29] Kimmo Koskenniemi. 1984. A general computational model for word-form recognition and production. In *10th International Conference on Computational Linguistics and 22nd Annual Meeting of the Association for Computational Linguistics*. 178–181.

[30] Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. *CoRR* abs/1808.06226 (2018). arXiv:1808.06226 http://arxiv.org/abs/1808.06226

[31] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML '01)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 282–289. https://api.semanticscholar.org/CorpusID:219683473

[32] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Kevin Knight, Ani Nenkova, and Owen Rambow (Eds.). Association for Computational Linguistics, San Diego, California, 260–270. https://doi.org/10.18653/v1/N16-1030

[33] N. P. MAAKE. 1991. LANGUAGE AND POLITICS IN SOUTH AFRICA WITH REFERENCE TO THE DOMINANCE OF THE NGUNI LANGUAGES. *English Studies in Africa* 34, 2 (1991), 55–64. https://doi.org/10.1080/00138399108690880 arXiv:https://doi.org/10.1080/00138399108690880

[34] Alexandre Magueresse, Vincent Carles, and Evan Heetderks. 2020. Low-resource Languages: A Review of Past Work and Future Challenges. arXiv:2006.07264 [cs.CL]

[35] Peter H. Matthews. 1991. *Morphology* (2 ed.). Cambridge University Press, Cambridge, UK.

[36] 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)*, Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue (Eds.). ELRA and ICCL, Torino, Italia, 12247–12258. https://aclanthology.org/2024.lrec-main.1071

[37] Andrei Mikheev, Marc Moens, and Claire Grover. 1999. Named entity recognition without gazetteers. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*. 1–8.

[38] Tumi Moeng, Sheldon Reay, Aaron Daniels, and Jan Buys. 2021. Canonical and Surface Morphological Segmentation for Nguni Languages. arXiv:2104.00767 [cs.CL]

[39] David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticæ Investigationes* 30, 1 (2007), 3–26. https://doi.org/10.1075/li.30.1.03nad

[40] 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*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 116–126. https://aclanthology.org/2021.mrl-1.11

[41] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. arXiv:1802.05365 [cs.CL]

[42] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. *Improving language understanding with unsupervised learning*. Technical Report. OpenAI. https://api.semanticscholar.org/CorpusID:49313245

[43] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. arXiv:1606.05250 [cs.CL]

[44] Biffie Viljoen Rigardt Pretorius and Laurette Pretorius. 2005. A finite-state morphological analysis of Tswana nouns. *South African Journal of African Languages* 25, 1 (2005), 48–58. https://doi.org/10.1080/02572117.2005.10587248 arXiv:https://doi.org/10.1080/02572117.2005.10587248

[45] Teemu Ruokolainen, Oskar Kohonen, Sami Virpioja, and Mikko Kurimo. 2013. Supervised Morphological Segmentation in a Low-Resource Learning Setting using Conditional Random Fields. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, Julia Hockenmaier and Sebastian Riedel (Eds.). Association for Computational Linguistics, Sofia, Bulgaria, 29–37. https://aclanthology.org/W13-3504

[46] Jasdeep Singh, Bryan McCann, Richard Socher, and Caiming Xiong. 2019. BERT is Not an Interlingua and the Bias of Tokenization. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, Colin Cherry, Greg Durrett, George Foster, Reza Haffari, Shahram Khadivi, Nanyun Peng, Xiang Ren, and Swabha Swayamdipta (Eds.). Association for Computational Linguistics, Hong Kong, China, 47–55. https://doi.org/10.18653/v1/D19-6106

[47] Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021. Masked Language Modeling and the Distributional Hypothesis: Order Word Matters Pre-training for Little. arXiv:2104.06644 [cs.CL]

[48] Adam Skory and Maxine Eskenazi. 2010. Predicting cloze task quality for vocabulary training. In *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*. 49–56.

[49] Abdel Aziz Taha and Allan Hanbury. 2015. Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. *BMC Medical Imaging* 15, 1 (12 Aug 2015), 29. https://doi.org/10.1186/s12880-015-0068-x

[50] Knut Tarald Taraldsen. 2010. The nanosyntax of Nguni noun class prefixes and concords. *Lingua* 120, 6 (2010), 1522–1548. https://doi.org/10.1016/j.lingua.2009.10.004 Contrast as an information-structural notion in grammar.

[51] Wilson L. Taylor. 1956. Recent Developments in the Use of "Cloze Procedure". *Journalism Quarterly* 33, 1 (1956), 42–99. https://doi.org/10.1177/107769905603300106 arXiv:https://doi.org/10.1177/107769905603300106

[52] Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL*

2003. 142–147. https://aclanthology.org/W03-0419

[53] Lisa Torrey and Jude Shavlik. 2010. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI global, 242–264.

[54] Reut Tsarfaty, Djamé Seddah, Sandra Kübler, and Joakim Nivre. 2013. Parsing Morphologically Rich Languages: Introduction to the Special Issue. *Computational Linguistics* 39, 1 (03 2013), 15–22. https://doi.org/10.1162/COLI_a_00133 arXiv:https://direct.mit.edu/coli/article-pdf/39/1/15/1798976/coli_a_00133.pdf

[55] Yulia Tsvetkov. 2017. Opportunities and Challenges in Working with Low-Resource Languages.

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention Is All You Need. arXiv:1706.03762 [cs.CL]

[57] 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*, Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (Eds.). Association for Computational Linguistics, Online, 473–482. https://doi.org/10.18653/v1/2021.naacl-main.40

[58] Ahmet Üstün, Murathan Kurfalı, and Burcu Can. 2018. Characters or morphemes: how to represent words? https://doi.org/10.18653/v1/w18-3019

A APPENDIX: EXPERIMENT IMPLEMENTATION DETAILS

A.1 Software Used

This project was written in Python (version 3.8.12) mainly from the Hugging Face "transformers" library. To leverage the parallelization of the Nvidia V100 GPUs, CUDA was used for the models using their PyTorch implementations. The CUDA standard development kit was version 11.5.1 while the Pytorch version was 1.10 to match.

A.2 Custom classes

Three custom Python classes were created for the PLMs in the project: MorphParseDataset, MorphParseModel and MorphParseArgs for the datasets, model loading/training and program command-line arguments respectively. The MorphParseDataset has the capacity of loading the train and test files and perform subword tokenization on them. The data can either be loaded in at sentence-level or at word-level. The MorphParseModel class has a custom training loop and loads in a trainer from the Hugging Face library. It collates the data into batches for training. The arguments class is simply for command-line arguments to specify directories, hyperparameters and logging information.

A.3 Reproducibility

In order to reproduce the results for this project, initialisation seeds were used for the Python modules used (numpy, pandas, transformers and torch). For the main part of the experimenting, including the grid search, the default seed of 42 was used. For testing, five random seeds were select 1 through 5 inclusive.

A.4 Scripts and GPU Usage

To train the models on the CHPC cluster, simple bash scripts were used. These scripts used the Portable Batch System (PBS) and ran the grid search and testing runs while saving the results to a configured JSON file. Each training script made use of 1 GPU core and 1 CPU core for the maximum cluster time of 12 hours. The JSON results file for the grid search contain a JSON object for each model run with IDs in the form language code_learning rate_batch size_number of epochs_model initial. The language codes were NR: isiNdebele, SS: siSwati, XH: isiXhosa and ZU: isiZulu. The model initials were x: XLM-R, a: Afro-XLMR, n: Nguni-XLMR. For the testing, another results script was created and the IDs were in the form language code_model initial_seed number.

A.5 Grid Search Limits

Grid search could not be optimised through PyTorch hyperparameter searching libraries like Ray and Optuna due to storage and training time requirements exceeding the storage capacity and time contracts of the CHPC's GPU cluster. Furthermore, the brute-force grid search space was reduced to make training and testing times feasible for the project timeline.

B APPENDIX: MAXIMUM F1 ALIGNMENT ALGORITHM

Algorithm 1 The maximal alignment algorithm

```
function ALIGN_SEQS(s, l)
    d ← LENGTH(l) − LENGTH(s)
    m ← LENGTH(l) − 1
    possible_indices ← GENERATE_INDICES(d, m)
    best_correct ← 0
    best_indices ← []
    for all i ∈ possible_indices do
        p ← PAD_AT_INDICES(s, i)
        c ← COUNT_MATCHES(p, l)
        if c ≥ best_correct then
            best_correct ← c
            best_indices ← i
        end if
    end for
    return PAD_AT_INDICES(s, best_indices)
end function

function GENERATE_INDICES(length_diff, max_index)
    if length_diff = 0 then
        return []
    else
        indices ← []
        for f ← max_index to 0 do
            i ← [f] ++ GENERATE_INDICES(length_diff − 1, f)
            indices ← indices ++ [i]
        end for
        return indices
    end if
end function
```