Deep Learning for Morphological Parsing

Cael Marquard
cael.marquard@gmail.com
University of Cape Town
Cape Town, Western Cape, South Africa

ABSTRACT

Morphemes are the smallest unit of meaning in a language. Morphological parsing refers to classifying the syntactic role of morphemes within a sentence. Nguni languages like isiXhosa and isiZulu are agglutinative, meaning that words often consist of many concatenated morphemes. Thus, many Natural Language Processing (NLP) tasks such as machine translation could be improved if morphological information were incorporated. This paper examines the existing literature surrounding morphological segmentation and parsing in order to explore the best ways to approach the problem of morphological parsing for the Nguni languages. The most promising approaches include sequence-labelling models which have been successfully applied to similar NLP tasks, such as bi-LSTM CRFs, bi-LSTMs, and Transformers.

KEYWORDS

Natural Language Processing, Machine Learning, Computational Linguistics, Morphological Parsing

1 INTRODUCTION AND MOTIVATION

Morphological parsing is the task of identifying the syntactic class of each morpheme within a word [23]. For example, in the sentence "There were many **windings** in the road", "wind" is a verb stem, "ing" marks the continuous tense, and "s" is a plural marker. The term "morphological parsing" is often used interchangeably with the terms "morphological analysis" and "morphological tagging". Full morphological parsing includes both the segmentation of a word into its constituent morphemes and their subsequent labelling [23].

Many common Natural Language Processing (NLP) tasks, such as machine translation and information retrieval rely on *lexical semantics*, or the meaning of individual words and their connections to other words [2]. Since the individual morphemes of which a word is composed create the whole word's meaning, morphological features are therefore important to NLP [23]. By developing improved tools for morphological parsing, these NLP tasks could be better solved through the incorporation of these features.

Morphological parsing is closely related to two other tasks in NLP: *morphological segmentation* and *part-of-speech tagging*.

Morphological segmentation is a task closely related to morphological parsing. Morphological segmentation refers to decomposing a word into its constituent morphemes. There are two important subcategories of this task: canonical and surface segmentation. Canonical segmentation refers to decomposing a word into its constituent *morphemes*, which are the smallest units of meaning in a language [6, 16, 25]. Surface segmentation refers to decomposing a word into its constituent *morphs*, which are the surface-form of morphemes as they appear in the word [16, 25]. A word's morphs may differ from its morphemes due to spelling changes that occur during morpheme attachment [16, 25].

Part-of-speech tagging refers to the task of labelling each word in a sentence with its lexical category, thus classifying the grammatical role that it plays in the sentence [21]. This is similar to morphological parsing, but operates at a word level rather than at a morpheme level. Some existing models, such as MarMoT [17], even combine part-of-speech tagging and morphological parsing into a single task. Because of this similarity, approaches used in part-of-speech tagging may be applicable to the task of morphological parsing.

Two features of Nguni languages make them especially receptive to modelling with morphemes: their agglutinativity and their conjunctive orthographies.

(1) Nguni languages are agglutinative, meaning that many words are created by aggregating morphemes [2, 16, 27].
(2) Nguni languages are written conjunctively, meaning that morphemes are concatenated together in the same word [27]. For example, in isiXhosa, "andikambuzi" means "I haven't yet asked him", and is composed of the morphemes "a", "ndi", "ka", "m", "buza", and "i".

Because of these features, a vocabulary constructed from a Nguni text's morphemes may be more general than one which is constructed simply from its words [3]. This could help models generalise to text containing words not seen in the same exact form in their training data. For instance, the word "ndimbuzile" means "I have asked him", and is composed of the morphemes "ndi", "m" "buza", "ile". Suppose that a model has come across this word in its training data and learned its meaning. A model which is aware of the morphological features of this word may generalise better to the unseen word "andikambuzi". This is because while it differs significantly in its surface form, it still contains the common morphemes "ndi", "m", and "buza". Otherwise, the model would need to learn the morphological analysis of each word on its own. Hence, incorporating morphological features into Nguni NLP models may have an out-sized impact due to the agglutinativity and conjunctive orthography of Nguni languages [2, 16, 27].

This project will examine two approaches to parsing morphemes in Nguni languages:

(1) The first approach will be to use a pre-trained Large Language Model (LLM), such as BERT, and fine-tune it on the morphological parsing task. This approach has been delegated to my partner.
(2) The second approach will be to train a model from scratch to perform the task. This task has been delegated to me.

2 PRESENTATION OF PRIOR WORK

2.1 Modelling

2.1.1 Rule-based approach. Traditionally, morphological parsing is done by manually incorporating grammatical and morphological rules about the language into a model. Often, Finite State Transducers (FSTs) are used. For example, the ZulMorph analyser is a morphological parser for isiZulu based on FSTs [2].

This approach requires linguists to define grammatical and morphological rules about each language which are then incorporated into the model. Therefore, these models require a high degree of linguistic expertise, are time-consuming to produce, and do not generalise well to dissimilar languages [4, 6]. Solutions based on machine learning, such as sequence-to-sequence and sequence labelling models, address these issues as they do not require any linguistic expertise to produce, need only existing annotated data to produce, and may be created in a language-independent manner.

2.1.2 Sequence-to-sequence models. One way to approach the tasks of morphological segmentation, part-of-speech tagging, and morphological parsing is with neural sequence-to-sequence models [16]. This means that an input sequence of words or characters is given to a model, which is tasked with creating an output sequence of labels. The output sequence does not necessarily have to be equal in length to the input sequence. For instance, Moeng et al. [16] applied Transformers [28] and bi-directional long short-term memory models to the task of canonical segmentation.

Long short-term (LSTM) neural networks may be used to solve sequence processing tasks [10, 21]. Recurrent neural networks (RNNs) are a type of neural network able to represent recent past events for use in future computations [10]. This allows them to incorporate the context of the past sequence while processing its items one by one. Hochreiter and Schmidhuber [10] propose LSTMs as a way to avoid issues such as vanishing and exploding gradients which are present in older RNN architectures. Specifically, LSTMs introduce a number of *memory cells* by which past information may be 'remembered' by the LSTM in future computations, thus forming an internal hidden state [10, 21]. This state is governed by learned *gate* functions which dictate what the LSTM will remember and what it will forget [10, 21].

Bi-directional LSTMs (bi-LSTMs) are a combination of two separate LSTMs, where one reads the input from the start to the end, and the other reads the input from the end to the start in reverse [21]. The hidden states produced by each LSTM are concatenated and then often fed through another fully-connected layer to create the final output [21]. This allows the model to take into account both the future and past of the sequence, which is useful for many sequence-processing tasks [9].

Transformers, as introduced by Vaswani et al. [28], are a type of a machine learning architecture which use an attention mechanism to provide context when processing input sequences. Notably, they do not have any sort of 'memory' as LSTMs or RNNs do. Attention provides a mechanism for the network to receive weighted context about the input sequence at each processing step. It allows dependencies to be modelled independently to their distance in the sequence. Attention is more amenable to parallelisation than recurrence as seen in RNNs and LSTMs. This allows training to be sped up.

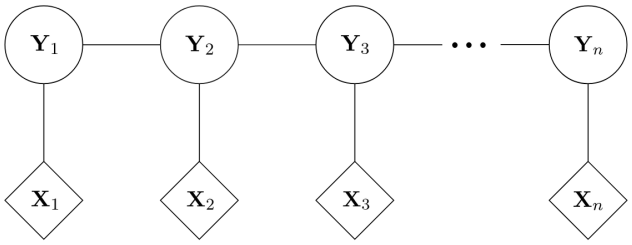

Figure 1: An example of a CRF organised as a chain [11]. Diamonds represent input variables $\mathbf{X_i}$ and circles represent output variables $\mathbf{Y_i}$, while edges represent statistical interdependence.

2.1.3 Sequence labelling models. Morphological segmentation, part-of-speech tagging, and morphological parsing are all often treated as a sequence labelling task [16, 23, 25]. This means that an input sequence of words or characters is given to a model which is tasked with creating a corresponding label sequence. For instance, a part-of-speech tagging model may be presented with the sentence "the dog runs" and be tasked with outputting the sequence "article; noun; verb".

Bi-LSTMs may be used to solve sequence labelling tasks, since they model a broad context of the sequence in both forward and backward directions [21].

Conditional Random Fields (CRFs) are a probabilistic model often used for these kind of sequence labelling tasks [11]. A CRF models the probability of a given input sequence $\mathbf{X}$ being assigned any possible output (labelling) sequence $\mathbf{Y}$. For example, the input sequence could be a vector of words, with the output sequence being a vector of part-of-speech tags which label the input. A CRF estimates the probability of a given output sequence's correctness by modelling the interdependence of the individual labels on each-other as a graph $G = (V, E)$. Each vertex v in the graph is a random variable from the output sequence or the input sequence (e.g. a random variable representing the part-of-speech tag for a given word in the sequence, or a random variable representing the given word itself). An edge from a vertex $\mathbf{V}_a$ to a vertex $\mathbf{V}_b$ means that the probability of the outcome of $\mathbf{V}_b$ depends on $\mathbf{V}_a$. Each vertex representing an output label $\mathbf{Y}_a$ is always connected to the vertex representing the word in the input which it labels $\mathbf{X}_a$. In the simplest example of a CRF, the graph is arranged as a chain, in which there is an edge connecting each $\mathbf{Y}_i$ to the next label $\mathbf{Y}_{i+1}$, and each label $\mathbf{Y_i}$ is connected to its input element $\mathbf{X_i}$ (Figure 1). Thus, the probability of a label depends only on the adjacent labels and its input.

CRFs have an advantage over LSTMs for sequence labelling, since they model the interdependence of the output labels on one another [11]. An LSTM operates in a greedy fashion, with the most likely tag being chosen for each individual output label. In comparison, a CRF computes the probability of the *entire* label sequence. This means that unlikely label sequences may be rejected by a CRF, whereas an LSTM may not. For example, a sentence composed entirely of verbs

is highly unlikely, because this would be grammatically ill-formed. Because the CRF explicitly models the interdependence of adjacent labels, these unlikely sequences have a higher likelihood of being rejected by the model [11]. An LSTM still may learn to do this by using its memory, however this is not inherent to the architecture of the LSTM and must be learned [10].

Traditionally, CRFs use a set of hand-crafted binary features in order to assign the probability of an individual label [16]. However, instead of designing these features by hand, a neural network, such as a bi-LSTM, may instead be used to learn the relevant features from the data [12, 13, 16].

2.2 Existing morphological parsers

Morphological parsing for the Nguni languages is traditionally done by using Finite State Transducers (FSTs). The ZulMorph analyser is an example of a hand-crafted finite state morphological analyser [2]. It has also been adapted to Nguni languages other than isiZulu, such as Southern Ndebele [22]. However, as previously mentioned, rule-based approaches require a high degree of linguistic expertise and are time consuming to produce [6].

Morphological parsing may be implemented as a two-step process, in which words are first segmented into morphemes and then fed into a separate tagging model [23]. This allows the leveraging of existing morphological segmenters such as those developed by Puttkammer and Du Toit [23] and Moeng et al. [16]. The tagging model may also then be quite similar to a part-of-speech tagging model, except operating on morphemes rather than words. This separation also lowers the complexity of the tagging model itself by reducing its scope.

Puttkammer and Du Toit [23] use a two-step approach in order to parse the morphemes of Nguni languages. Firstly, a sentence is split into its morphemes (in their canonical form). Then, these morphemes are fed into an existing, trainable morphological parser, which is then used to classify the morphemes according to their grammatical function. MarMoT [17] was chosen by the authors as the existing parser due its successful application to part-of-speech tagging for Nguni languages.

Akyürek et al. [1] apply a sequence-to-sequence model to the task of morphological parsing (applied to a variety of European languages). The sentence is given to the model as a sequence of characters and it then outputs the word's *lemma* (stem) character-by-character, followed by a set of morphological features, which are also produced feature-by-feature. Three encoder models are used in conjunction with one decoder model in order to produce the final output. The three encoder models are as follows:

- **Word encoder.** The word itself is encoded by processing the embedding of its characters from left to right.
- **Output encoder.** The morphological features of the two prior words in the sequence are encoded by the 'output' encoder by processing the features' encodings with a uni-directional LSTM.
- **Context encoder.** The embeddings of all the words, as produced by the word encoder, are processed by a bi-LSTM, producing a contextual embedding for each word in the sentence. The embedding for each word is the concatenation of the forward and backward LSTMs' outputs.

The decoder is a two-layer LSTM which uses the hidden states produced by the three encoders in order to produce the lemma morphological features for the given word. Two additional variants of the models were created: one which produces only the morphological tags of the word, and one which selected from the outputs of existing, rule-based morphological analysers. Producing the lemma as well as the morphological features was found to be more difficult than simply producing the features, especially when applied to low-resource languages. The overall approach obtained excellent performance on all nine languages it was evaluated on.

2.3 Related tasks

Machine learning has only been applied to morphological parsing for the Nguni languages by Puttkammer and Du Toit [23]. However, it has been applied to a few closely related sequence labelling tasks which share some structural and linguistic similarity with morphological parsing.

Named entity recognition (NER) refers to the identification of sequences in text such as names (proper nouns) and numeric expressions [20]. It relates to morphological parsing in that they may both be implemented as sequence-labelling tasks [12].

Lample et al. [12] apply Conditional Random Fields and bi-LSTMs to NER. In their CRF model, a bi-LSTM is used to encode a word's embedding into a representation of the word in its context. These contextual representations are then given to a CRF in order to model the interdependence of the tags in the output sequence. Specifically, the bi-LSTM is used to calculate a matrix P of size $n \times k$, where n is length of the input sequence and k is the number of tags. The score $s(\mathbf{X}, \mathbf{y})$ for a given input sequence $\mathbf{X}$ and label sequence $\mathbf{y}$ is given by

$$s\left(\mathbf{X}, \mathbf{y}\right) = \sum_{i=0}^{n} A_{y_i, y_{i+1}} + \sum_{i=1}^{n} Pi, y_i$$

The trainable matrix A is used to express the interdependence of the output tag probabilities on one another, such that $A_{y_i, y_{i+1}}$ represents the score of a transition from the tag i to the next tag $i + 1$. This models grammatical features of the language and would, for example, account for the fact that an article such as "the" is more likely to occur before a noun than a verb.

This approach is also adopted by Pannach et al. [21] in their part-of-speech tagging model.

Pannach et al. [21] implemented part-of-speech tagging for the Nguni languages using bi-LSTMs and bi-LSTM CRFs. For the bi-LSTM model, a bi-LSTM is used to produce hidden states for all of the words in the input, which is then passed to a fully-connected layer used to produce the probability for each POS tag. For the CRF model, the CRF's feature function is parameterised with a bi-LSTM which produces scores for each position in the sentence, which is similar to the approach taken by Lample et al. [12]. The bi-LSTM CRF has an advantage over a simple bi-LSTM in that it explicitly models the interdependence of the labels in the final sequence [11].

Pannach et al. [21] found that subword-level systems models consistently outperformed word-based models. Therefore, the CRF and bi-LSTM models were only evaluated with subword-level encoding. These took the form of two different approaches. The first approach was to break the word down into a vector of character

embeddings. The second approach was to break down the word into a vector of the word's 2-gram embeddings. A word's 2-grams refer to every consecutive two-character sequence that it contains, often with special start-of-word and end-of-word tokens added. In order to combine these subword-level representations to form word-level representations, the subword-level vectors were simply added together. The authors note that, though this discards positional information, a bi-LSTM-based encoder which would take into account this information did not improve the overall model's performance. In the end, the authors found that the best approaches for the part-of-speech tagging task were CRFs using 2-gram features and bi-LSTM using character features.

Since the algorithm required to train CRFs has a runtime quadratic in the size of the possible tag-set, they are not well-suited to tasks which have a very large number of possible tags [7, 16, 17]. Mueller et al. [17] addressed this issue by employing coarse-to-fine encoding in the CRF objective function and pruning the CRF during training [7]. The authors then applied this approach to the task of part-of-speech tagging as well as combined part-of-speech and morphological tagging, and found that the training was significantly faster than training a standard CRF [17]. MarMoT's CRF uses features based on unigrams, bigrams, and other surface features of the surrounding text as suggested by Ratnaparkhi [24] and Manning [14].

MarMoT [17] has subsequently been applied successfully by du Toit and Puttkammer [7] to the task of part-of-speech tagging for the Nguni languages (as well as morphological tagging, as previously discussed).

2.4 Morphological segmentation

Morphological segmentation refers to the decomposition of words into the morphemes or morphs of which they are made [6, 16, 25]. Given that full morphological parsing includes segmentation as one of its steps and that morphological parsers may be trained on the output of existing morphological segmenters, it is a closely related task [23].

2.4.1 Surface segmentation. Surface segmentation, or the segmentation of words into the surface morphs of which they are composed, may be approached with both supervised and unsupervised machine learning approaches [16, 25].

Moeng et al. [16] implement several surface segmentation models for the Nguni languages, including a feature-based CRF, a bi-LSTM CRF, and an unsupervised entropy-based model. The best model was found to be a feature-based CRF. In this case, the task of surface segmentation is treated as a sequence-labelling task, in which the input sequence of characters is labelled by a tag representing how it contributes to the overall morphological structure of the word. Specifically, the tag 'B' is used to mean that the letter starts a morpheme, 'M' means that it continues the morpheme, 'E' means it ends the current morpheme, and 'S' means that the letter forms a morpheme by itself. The likelihood of a given tag sequence is predicted by a CRF. The features which the CRF uses are character n-grams ($0 \leq n \leq 6$), whether the character is a consonant or vowel, and the case of the character.

Unsupervised machine learning has also been used to segment and parse morphemes in different languages, such as Finnish, isiXhosa, isiZulu, isiNdebele, siSwati, and Bengali [3, 4, 6, 16, 19].

Creutz and Lagus [3, 4] apply an unsupervised probabilistic approach in order to segment words into morphs. The first model developed, known as Morfessor Baseline, uses a maximum a posteriori (MAP) estimate in order to maximise the likelihood of segmenting the given corpus correctly. At first, the model considers each word to be its own morph. Thereafter, the list of morphs is refined. This is done by generating every way that each morph may be split into two substrings, as well as the morph itself, unsplit. The most likely splitting (or lack thereof) is then chosen, and the model is updated accordingly. The likelihood of a morph is a maximum likelihood estimate and is given by its frequency in the corpus divided by the sum of the frequency of all morphs in the corpus.

Creutz and Lagus [4] note that the Baseline model, unlike later versions, does not incorporate any notion of grammar, meaning that morphemes are assigned the same probability regardless of where they are placed in a word. This is in contrast to approaches which use CRFs, given that they explicitly model statistical dependencies between labels [11].

Mzamo [19] uses a branching entropy approach as well as a probabilistic approach in order to segment isiXhosa into surface morphs. These models were both found to be comparable to Morfessor Baseline [3, 4] in terms of accuracy, but superior in terms of their F1 scores.

Moeng et al. [16] use an entropy model to segment isiXhosa, isiZulu, isiNdebele and siSwati, based on the assumption that successive letters inside of a morpheme are decreasingly less random. A character-level LSTM is used to estimate the probability of characters occurring in a sequence. One LSTM is trained in the forward (left-to-right) direction, and another is trained in the backward (right-to-left) direction. Then, the word is segmented wherever the sum of the left and right entropy exceeds an experimentally-established threshold. This approach did not perform nearly as well as the supervised learning approaches evaluated in the same paper.

Dasgupta and Ng [6] use unsupervised machine learning to segment Bengali (though the approach is language agnostic). During training, the model learns root words and affixes from the language. Then, at segmentation time, all possible segmentations are generated using the learned affixes and roots. These possible segmentations are reduced through a series of tests, eliminating the most unlikely options, until only one segmentation remains. This approach achieved relatively good results for Bengali, but further improvements would still be possible. For instance, the authors note that the algorithm does not deal well with spelling changes which may occur during morpheme attachment. This may limit its applicability to Nguni languages, in which words often undergo spelling changes during morpheme attachment [23].

2.4.2 Canonical segmentation. Canonical segmentation is a related task in which a word is decomposed into the canonical form of the morpheme [6, 16, 25]. For instance, while "andingotata" may be segmented into "a", "ndi", "ngo", "tata" at a surface level, the canonical segmentation would yield "a", "ndi", "nga", "u", "tata". Thus, a canonical segmenter must also recover the morphemes

IsiXhosa 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 collected by Gaustad and Puttkammer [8].

of the word *before* any sound changes due to morphotactics and morpho-phonological rules have occurred [6].

Puttkammer and Du Toit [23] use a memory-based supervised learning algorithm (IB2, provided by TiMBL [5]) to predict each point at which the word should be segmented, as well as the spelling changes that should occur following these points in order to recover the canonical morpheme. The canonical morphemes are then reconstructed by applying these changes to the surface segmentation of the word. The accuracy achieved ranges from 85-95% depending on the language. As the authors express the performance of their model in terms of accuracy, it is difficult to compare to the model developed by Moeng et al. [16], who expressed their models' performance in terms of their F1 scores.

Moeng et al. [16] also applied two sequence-to-sequence models, namely bi-LSTMs and Transformers, in order to segment words into their canonical morphemes. While Transformers were the more effective of the two, neither of the sequence-to-sequence models managed to achieve a degree of accuracy similar to what was achieved for the simpler surface segmentation task.

In general, canonical segmentation for Nguni languages is not yet as accurate as surface segmentation. This is possibly due to the task itself being more difficult, given that the model must be able to reverse many possible spelling changes for the same morpheme.

2.5 Data

Previously, the Ukwabelana corpus [26] has been used to train morphological segmenters for isiZulu [15]. This corpus is accompanied by its labelled surface segmentation [15, 26]. However, this corpus covers only isiZulu. This means that using this corpus to train morphological taggers for other Nguni languages could lead to poorer results, as the resulting model would need to generalise to unseen languages with potentially different morphologies, grammars, and lexicons.

Gaustad and Puttkammer [8] collected a corpus of four Nguni languages (isiZulu, isiXhosa, isiNdebele, and siSwati) which has been manually annotated with each word's morphemes and their grammatical functions. Since this corpus is accompanied by its canonical segmentation and morphological tagging, this dataset is suitable to use in the training of supervised morphological taggers. Table 1 illustrates the morphological analysis of three words from the dataset's trainset for isiXhosa.

3 COMPARISON

Existing morphological parsers are constructed by hand, meaning that they require a high degree of linguistic expertise, are time consuming to produce, and hard to generalise to other languages (even if they are related) [4, 6].

Full morphological parsing may be broken down into a two-step process, consisting of segmentation and then tagging [23]. Many morphological segmenters already exist for the Nguni languages [16, 19, 23]. Fewer morphological taggers exist for the Nguni languages, with only the tagger developed by Puttkammer and Du Toit [23] using machine learning methods. However, this tagger uses standard CRF features and does not incorporate neural features.

Surface segmentation has been implemented by Moeng et al. [16] to a very high degree of accuracy, but canonical segmenters do not perform as well. Therefore, a morphological parser based on a pre-existing canonical segmentation may perform worse due to the high error in the segmentation step. For example, the tagger trained by Puttkammer and Du Toit [23] achieved a consistently lower tagging accuracy than its segmentation accuracy. Puttkammer and Du Toit [23] did not evaluate their morphological tagger on surface segmentations.

Unsupervised machine learning, while able to leverage large amounts of readily-available, unlabelled text data, has not been particularly successful in NLP tasks relating to morphological parsing. Supervised models present a more promising approach to this task.

Morphological parsing may be implemented either as a sequence-to-sequence model or as a sequence tagging model. In the case of sequence-to-sequence models, Bi-LSTMs and Transformers have been used to implement canonical segmentation by Moeng et al. [16], with Transformers found to be better suited to that specific task [28]. Sequence-to-sequence models have not been applied to morphological parsing for the Nguni languages. This may prove to be a promising approach, given the results achieved by the sequence-to-sequence morphological parsing model developed by Akyürek et al. [1].

In the case of sequence tagging models, bi-LSTMs, feature-based CRFs, and bi-LSTM CRFs have been used to implement surface segmentation, part-of-speech tagging, and morphological parsing [16, 21, 23]. Models which incorporate CRFs may have an advantage over other sequence tagging models in that they explicitly model the interdependence of the labels in the output sequence on one another [11].

Gaustad and Puttkammer's [8] annotated corpus could be used to train a machine learning model to parse morphemes in Nguni languages, since it is annotated with the full morphological analysis of each word in the corpus. Each word is accompanied by its canonical segmentation, so if a model were to be trained on surface-segmented input, special care would need to be taken to ensure the correct alignment of the morphological tags with their surface morph forms.

Pannach et al. [21] found that subword-level systems outperform word-based models in performing part-of-speech tagging. This could be due to the fact that part-of-speech and morphological properties are mutually-dependent tasks in agglutinative languages [18, 23]. Regardless, this means that some existing part-of-speech taggers already operate on a subword level, which is what is required for morphological tagging [21]. This means that these specific architectures may be a good starting-point for morphological parsing.

4 CONCLUSIONS

Morphological parsing is an important task in NLP. With the incorporation of morphological features, as extracted by morphological parsers, many key NLP tasks such as information retrieval and machine translation could be improved. This applies especially to the Nguni languages, which are highly morphologically complex.

However, there is room for improvement in the existing Nguni morphological parsers. Most existing models are created through a manual process which requires a high degree of linguistic expertise, is time consuming, and the resulting models do not generalise easily to other languages. Solutions based on machine learning require little to no linguistic expertise to produce, may be language independent, and can leverage existing annotated corpora.

Both sequence-to-sequence models and sequence labelling models have achieved promising results on morphological tagging in other languages and related NLP tasks for the Nguni languages. Sequence-to-sequence models such as Transformers, encoder-decoder LSTM models, and bi-LSTMs could be used to implement parsing for the Nguni languages. Sequence labelling models such as bi-LSTMs and bi-LSTM CRFs are also promising.

Based on the prior work which has been reviewed, bi-LSTM CRFs seem to be a promising approach to the task, and as such will be evaluated first. Other models which will be evaluated include sequence-to-sequence bi-LSTMs, Transformers, and feature-based CRFs.

REFERENCES

[1] Ekin Akyürek, Erenay Dayanık, and Deniz Yuret. 2019. Morphological Analysis Using a Sequence Decoder. *Transactions of the Association for Computational Linguistics* 7 (2019), 567–579. https://doi.org/10.1162/tacl_a_00286

[2] Sonja E. Bosch and Laurette Pretorius. 2017. A Computational Approach to Zulu Verb Morphology within the Context of Lexical Semantics. *Lexikos* 27 (00 2017), 152 – 182. http://www.scielo.org.za/scielo.php?script=sci_arttext&pid=S2224-00392017000100007&nrm=iso

[3] Mathias Creutz and Krista Lagus. 2002. Unsupervised Discovery of Morphemes. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*. Association for Computational Linguistics, 21–30. https://doi.org/10.3115/1118647.1118650

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

[5] W. Daelemans, J. Zavrel, K. Sloot, and A. Bosch. 2004. *TiMBL: Tilburg memory-based learner, version 6.4: reference guide.* Technical Report. Tilburg University, Tilburg.

[6] 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

[7] Jakobus S. du Toit and Martin J. Puttkammer. 2021. Developing Core Technologies for Resource-Scarce Nguni Languages. *Information* 12, 12 (2021). https://doi.org/10.3390/info12120520

[8] Tanja Gaustad and Martin Puttkammer. 2022. Linguistically annotated dataset for four official South African languages with a conjunctive orthography: IsiNdebele, isiXhosa, isiZulu, and Siswati. *Data in Brief* 41 (02 2022), 107994. https://doi.org/10.1016/j.dib.2022.107994

[9] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks* 18, 5 (2005), 602–610. https://doi.org/10.1016/j.neunet.2005.06.042 IJCNN 2005.

[10] 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

[11] 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.

[12] 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

[13] Xuezhe Ma and Eduard Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Katrin Erk and Noah A. Smith (Eds.). Association for Computational Linguistics, Berlin, Germany, 1064–1074. https://doi.org/10.18653/v1/P16-1101

[14] Christopher D. Manning. 2011. Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics?. In *Computational Linguistics and Intelligent Text Processing*, Alexander F. Gelbukh (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 171–189.

[15] Sthembiso Mkhwanazi and Laurette Marais. 2024. Generation of segmented isiZulu text. *Journal of the Digital Humanities Association of Southern Africa* 5, 1 (Feb. 2024). https://doi.org/10.55492/dhasa.v5i1.5034

[16] Tumi Moeng, Sheldon Reay, Aaron Daniels, and Jan Buys. 2022. Canonical and Surface Morphological Segmentation for Nguni Languages. In *Artificial Intelligence Research*, Edgar Jembere, Aurona J. Gerber, Serestina Viriri, and Anban Pillay (Eds.). Springer International Publishing, Cham, 125–139.

[17] Thomas Mueller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient Higher-Order CRFs for Morphological Tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard (Eds.). Association for Computational Linguistics, Seattle, Washington, USA, 322–332. https://aclanthology.org/D13-1032

[18] Thomas Müller and Hinrich Schuetze. 2015. Robust Morphological Tagging with Word Representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Rada Mihalcea, Joyce Chai, and Anoop Sarkar (Eds.). Association for Computational Linguistics, Denver, Colorado, 526–536. https://doi.org/10.3115/v1/N15-1055

[19] Lulamile Mzamo. 2021. *Morphological segmentation of isiXhosa using unsupervised machine learning.* Ph. D. Dissertation. North-West University (South Africa).

[20] 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

[21] Franziska Pannach, Francois Meyer, Edgar Jembere, and Sibonelo Zamokuhle Dlamini. 2022. NLAPOST2021 1st Shared Task on Part-of-Speech Tagging for Nguni Languages. *Journal of the Digital Humanities Association of Southern Africa* 3, 01 (Feb. 2022). https://doi.org/10.55492/dhasa.v3i01.3865

[22] Laurette Pretorius and Sonja Bosch. 2012. Semi-automated extraction of morphological grammars for Nguni with special reference to Southern Ndebele. *Language Technology for Normalisation of Less-Resourced Languages* (2012), 73–78.

[23] Martin Puttkammer and Jakobus S. Du Toit. 2022. Canonical Segmentation and Syntactic Morpheme Tagging of Four Resource- scarce Nguni Languages. *Journal of the Digital Humanities Association of Southern Africa* 3, 03 (Feb. 2022). https://doi.org/10.55492/dhasa.v3i03.3818

[24] Adwait Ratnaparkhi. 1996. A Maximum Entropy Model for Part-Of-Speech Tagging. In *Conference on Empirical Methods in Natural Language Processing*. https://aclanthology.org/W96-0213

[25] 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

[26] Sebastian Spiegler, Andrew van der Spuy, and Peter A. Flach. 2010. Ukwabelana - An open-source morphological Zulu corpus. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, Chu-Ren Huang and Dan Jurafsky (Eds.). Coling 2010 Organizing Committee, Beijing, China, 1020–1028. https://aclanthology.org/C10-1115

[27] Elsabe Taljard and Sonja Bosch. 2006. A Comparison of Approaches to Word Class Tagging: Disjunctively vs. Conjunctively Written Bantu Languages. *Nordic Journal of African Studies* 15 (01 2006). https://doi.org/10.53228/njas.v15i4.37

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf