Morphological Segmentation of Low-Resource Languages Using Conditional Random Fields

Tumi Moeng
University of Cape Town Computer Science Department
Cape Town, South Africa

ABSTRACT

The Nguni languages of Southern Africa have had few Language Processing Applications developed for them, and what applications have been developed have been low accuracy. How can we facilitate the development of more high accuracy applications? Within the field of Computational Linguistics there is a study called Morphological Segmentation. This study deals with the breaking down of individual words into morphemes, the smallest unit of language with meaning. Using these techniques would facilitate the creation of better language processing applications because words could be broken down to their smallest units to determine their meaning. These languages are low-resource meaning there is little access to text data, both annotated and unannotated. Thus a model is needed that can segment and label words with high accuracy, and do that with low amounts of text data.

This task can be accomplished through the use of several Machine Learning models but the model that will be the focus of this paper is known as Conditional Random Fields (CRF). CRFs are a class of discriminative probabilistic models used to label sequence data such as text corpora and images, using Supervised Machine Learning. Based on a review of literature on CRFs and similar models, we have determined that due to their ability to perform with an *F-measure*, a balance of *Recall* and *Precision*, of above 90% with very little data, CRFs would be a wise choice of model to use. In future work, a comparison between CRFs and other models, such as Sequence to Sequence and Morfessor models, will be conducted to determine which model would be the best suited to this task.

KEYWORDS

Natural Language Processing (NLP); Conditional Random Fields (CRF); Computational Linguistics; Morphological Segmentation; Surface Segmentation; Supervised Machine Learning; Nguni Languages; Low-resource Languages; Sequence to Sequence; Morfessor;

INTRODUCTION

Morphological Segmentation is a task studied in linguistics and computational linguistics. It is most relevant in the context of polysynthetic languages. Polysynthetic language are languages in which the words are composed of multiple morphemes, which are units of language that cannot be further subdivided, and have meaning on their own[14]. This segmentation can be done manually but for efficiency purposes there has been a move to doing it through the use of Machine Learning both supervised and unsupervised. Supervised Machine Learning entails the training of a model through the use of data that is correctly labelled, the machine will then learn to label unlabelled data based on the information it has gained from the training data. Unsupervised Machine Learning entails the training of a model with unlabelled data, and then the model works to find patterns in the data. This paper will focus on supervised machine learning with a more specific focus on Conditional Random Fields (CRF). CRFs are a class of models that use conditional probability to determine the best fit label sequence to an input sequence[7].

Many languages unique to certain countries in the world could be considered low-resource. What this means is that due to a lack of data in that language, there is a lack of language processing applications, such as speech to text and translation services. This lack of data is exacerbated by a combination of few speakers and a lack of research being done into these languages. This project focuses on a family of low-resource Southern African languages called the Nguni languages, with a specific focus on isiZulu, isiXhosa, isiNdebele and siSwati. This is a group of languages that have had some Natural Language Processing (NLP) applications developed for them, however there are few of them, and they are frequently low accuracy applications. These languages are also considered morphologically rich and polysynthetic. What this means is that despite the lack of data, if the morphemes can be understood then NLP applications can be developed using this knowledge and machine learning techniques, mitigating the need for large corpora of text data.

This project is motivated by the fact that there are few applications conducting this research and work for low-resource languages. It is our hope that this project others like it will facilitate a movement towards the further development of more and superior NLP applications for these languages. This research and the tools that come from it would be of benefit to speakers and students of the language, and linguists that wish to do research in the language. Additionally, languages

are important, they facilitate communication between people and ultimately allow us to connect. Given enough time these low-resource languages may eventually die out being replaced by more commonly used languages, due to the decreasing number of speakers. The preservation of these languages can be aided by NLP applications and other forms of research into them.

MORPHOLOGICAL SEGMENTATION

The act of segmenting a word is achieved by separating the word into its various sub-components, morphemes. This is done so that the word can be studied at a granular level. Here we will discuss two ways in which words can be segmented. Once segmented these words can be further categorized through labelling. If the segmentation form is labelled then labels will be added to each of the substrings, these labels will define the function of that substring to the word as a whole such as word root, suffix and prefix.
Given a word, w, the word can be segmented using Surface Segmentation and Canonical Segmentation.

Surface Segmentation

Using Surface Segmentation, the word w will be segmented into a sequence of substrings, which when added back together will form the word w again[4].

Canonical Segmentation

Using Canonical Segmentation, the word w will be analyzed as a sequence of canonical morphemes, based on word forms that will have been annotated for Supervised Machine Learning. Each canonical morpheme c will correspond to a surface morpheme s as its Orthographic Manifestation. This means that c will be the same as s after applying editing operations such as insertion, deletion and modification[4].

Given the word "*attainability*":

- **Surface Segmentation**: *attain-abil-ity*
- **Canonical Segmentation**: *attain-able-ity*

As can be seen, to move from the Surface Segmentation form to the Canonical Segmentation form, two edit operation have been performed, one on the i in the middle segment and it has been converted to an l, and another on the l in the middle segment which has been converted to an e.
Canonical Segmentation has the advantage of being a superior linguistic analysis tool because the morphemes it creates are more accurate to the language being studied than those created by Surface Segmentation[4].
However, as CRFs are unable to predict canonical segmentation directly without extensions being made to the model, this review will focus on Unlabelled Surface Segmentation, and Labelled Segmentation of the Orthographic Manifestation.

CONDITIONAL RANDOM FIELDS

CRFs are a class of discriminative probabilistic models used to segment and label sequence data such as bodies of text[7, 18]. Discriminative Models are used in contrast to Generative Models, what sets them apart is that Discriminative Models are able to directly model the conditional probability $p(\mathbf{Y}|\mathbf{X})$, meaning the probability of $\mathbf{Y}$ given $\mathbf{X}$[18, 22]. Additionally, they are able to learn direct mappings from an input, $\mathbf{X}$, to a label, $\mathbf{Y}$[22]. In contrast, Generative Models construct joint probability models of $\mathbf{X}$ and $\mathbf{Y}$, of the form $p(\mathbf{X},\mathbf{Y})$, and then used Bayes Rules to calculate $p(\mathbf{Y}|\mathbf{X})$[18, 22]. It is generally agreed that Discriminative Models are superior because they are able to solve the classifier problem directly with no middle step[22]. Probability is an important part of the labelling process, the way that the correct label or sequence of labels is chosen is through the maximization of probabilities.
CRFs accept data sequences as inputs, generally in the form of text corpora. They output label sequences, with all the individual labels being defined within a label alphabet. They make predictions for the labels that will belong to each segment using conditional probabilities based on the input. They way they segment words is by being read in character by character, and then each character will be marked according to whether it is the start of a new segment, a part of the current segment, the end of the current segment and a single character segment[14].
This project will involve the CRFs being trained on annotated text corpora, then given sequences of text to be labelled, in the aforementioned languages. Using the probabilities learned from the training stage the CRF will then segment the words in the text and label all the segments.
CRFs are based on Markov Models however their performance is typically superior[7]. Two examples of Markov Based Models are Hidden Markov Models (HMM), and Maximum Entropy Markov Models (MEMM). HMMs are Generative Models that aim to assign joint probability to paired observations and label sequences, $p(\mathbf{X},\mathbf{Y})$[7]. This model assumes each label is only dependent on the previous label only and that each observed x is dependent on y[18]. With these $p(\mathbf{X},\mathbf{Y})$ can be calculated[18]. MEMMs are discriminative conditional probabilistic sequence models wherein each state takes features of input $\mathbf{X}$, and outputs a distribution over the possible following states[7]. The goal being probability maximization of a sequence. These models, whilst functional, share a problem that CRFs are able to overcome known as the *label bias problem*. This is a problem whereby if one of two similar words, '*pat*' and '*pet*' for example, are more common in the training set, the states will favour that word's transition and that word's sequence will be chosen

over the other word's meaning it will always mislabel the other word[7]. This fact contributes to CRFs outperforming other state based classifiers.

Traditional Conditional Random Fields

Given the random variable **X**, the input sequence for which the CRF needs to predict the output label sequence, **Y**. Given a finite set of labels, the CRF needs to be able to determine the conditional probability, for each segment, $p(\mathbf{Y}_i|\mathbf{X}_i)$ here meaning the probability of label $\mathbf{Y}_i$ given the segment $\mathbf{X}_i$. In broader terms the CRF will need to determine the highest probability $p(\mathbf{Y}|\mathbf{X})$, where **Y** is a sequence of labels given a sequence of input **X**. It will do this taking into account the current label, $\mathbf{Y}_1$, and a previous label ,$\mathbf{Y}_2$. Finally, in order to ensure the CRF is valid it also needs to obey the Markov Property that given $p(\mathbf{Y}_1|\mathbf{X},\mathbf{Y}_2)$, $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are neighbors[7]. The combination of all these factors ensures a valid CRF with the fixed graph G=(V,E) where Y is indexed by the vertices of G, and E being path between one node and the next. We say it is fixed because the graph is conditioned on the observation(s) **X**. In simple terms G is a linear chain [7]. Mathematically CRFs compute probabilities as follows:

Assuming the labels of **Y** conditioned on **X** form a chain,we define $\mathbf{Y}_0$, as the start state, and $\mathbf{Y}_{n+1}$, as the stop state. Given this chain structure, a Matrix, **M**, can be defined for which the probability of a label sequence is given[7]. For each position, *i*, in the sequence **X**, we define a |y|*|y| matrix with the random variable $M_i(x) = [M_i(y',y|x)]$ by:

$$\begin{aligned} M_i(y',y|x) &= exp\,(\Lambda_i(y',y|x)) \\ \Lambda_i(y',y|x) &= \sum_k \lambda_k f_k(e_i, Y|_{e_i} = (y',y), x) \\ &\quad + \sum_k \mu_k g_k(v_i, Y|_{v_i} = y, x) \end{aligned} \tag{1}$$

Here, x represents the input sequence, and y represents a label output sequence. Additionally, e_i is the edge with the labels (Y_{i-1},Y_i) and v_i is the vertex with the label Y[7]. Finally, f_k and g_k are features of input sequence, **X**, that are assumed to be given and fixed[7]. For example a vertex feature of boolean form g_k might be True if the segment at $\mathbf{X}_i$ is '-ing' and the tag is "Suffix". Simply put, this equation creates a matrix representing the probability of a label for a given position of the input sequence.

Unlike most Markov State Models where conditional probability distributions for each position, *i*, in sequence needs to sum to one, CRFs are globally normalized meaning conditional probability for position *i* does not need to sum to one[11]. As such, there is the need for a function, $Z_\theta(x)$, that will ensure the model will define a valid distribution. This comes at the cost that the probability distribution over *Y* is not valid till the whole sequence **X** has been processed[11]. This is known as a *normalization (partition)* function, and it is the product of all the matrices for the input sequence[7]. It is defined as follows:

$$\begin{aligned} Z_\theta(x) &= (M_1(x) * M_2(x) * ... * M_{n+1}(x)) \\ &= \prod_{i=1}^{n+1} M_i(x) \end{aligned} \tag{2}$$

Therefore, for the input sequence, **X**, the probability of a label sequence, **Y**, being assigned to that input sequence is given by:

$$p_\theta(Y|X) = \left(\frac{1}{Z_\theta(x)}\right) * \prod_{i=1}^{n+1} M_i(y_{i-1}, y_i|x) \tag{3}$$

Finally, θ is a parameter that needs to be determined, and it represents $(\lambda_1,\lambda_2,...,\lambda_n;\mu_1,\mu_2,...,\mu_n)$[7]. θ will be estimated using the training data, $D = \{(x^i, y^i)\}_{i=1}^n$, with the distribution $p(x, y)$[7]. θ is calculated as follows, using a log-likelihood objective function[7]:

$$O(\theta) = \sum_{i=1}^{N} log p_\theta(y^i, x^i) \tag{4}$$

Neural Conditional Random Fields

This subset of CRFs is an extension upon traditional CRFs achieved by combining them with Neural Networks (NN). This can be accomplished through the placement of a feed forward NN between inputs and the energy function, which calculates the cost of an observation ,**X**, to adopt a given label. The most probable assignment of labels will be when the energy of the model is minimized[23]. A feed forward NN is a simple implementation of a NN which takes input and each layer processes the input and pushes output to units of the next layer till it reaches the output layer which will, in this case, produce a number of quantities known as *energy outputs*. It should be noted that these outputs are different from the output of the Neural CRF model[6].

Alternatively, there are approaches involving another class of models known as Recurrent Neural Networks (RNN). RNNs are a group of models that operate on sequences, similar to CRFs[8]. They take in a sequence of vectors and output a sequence of data that represents the input vector at each corresponding position[8]. However, RNNs suffer from a problem that they are biased to recent inputs in a sequence which can lead to previous patterns being overlooked in favour of more recent ones, this is known as the vanishing problem[8, 9]. To counteract this problem Long Short-Term Memory Networks (LSTM) were developed which have a memory cell and as such can capture long term dependencies[8]. Normally an LSTM generates the left context of a sequence at every position, however it would be useful to also get the right context of a sequence at every position. This was achieved

by adding a second LSTM that reads the sequence in reverse, the first LSTM is known as the *forward LSTM* and the second is known as the *backward LSTM*[8]. The pair are combined to create what is known as a bidirectional LSTM (BLSTM) which can create a representation of words in context from both sides known as h_t[8]. To implement this with CRFs, the BLSTM is used to generate a n^*k matrix, called P, where n is the number of words in the sequence and k is the number of labels in the label alphabet. Each position P_{ij} is the score of the j^{th} label of the i^{th} word in the sequence for a given sequence of predictions[8].
The score is given by:

$$s\left(X, y\right) = \sum_{i=0}^{n} A_{y_i, y_{i+1}} + \sum_{i=1}^{n} P_{i, y_i} \tag{5}$$

In this equation A is a square matrix, of dimensions n+2 where n is equal to the number of labels in the label alphabet, and the 2 added represent the *start* label and *end* label of the sequence[8]. Each position $A_{i,j}$ represents the score of the transition from label i to label j[8]. Yx represents the set of all possible distributions of labels over the input sequence[8].
The final conditional probability equation is given by:

$$p(Y|X) = \frac{e^{s(x,y)}}{\sum_{\widetilde{y} \epsilon Yx} e^{s(x,\widetilde{y})}} \tag{6}$$

This calculates the score of this sequence divided by the score of all possible sequences to determine the probability of that exact sequence of labels. During Neural CRF Training, the model attempts to maximize the log probability of a supplied correct label sequence using the equation[8]:

$$log(p(Y|X)) = s(x, y) - log\left(\sum_{\widetilde{y} \epsilon Yx} e^{s(x,\widetilde{y})}\right) \tag{7}$$

This equation creates the probabilities that will be used by the model when operating on data.

Semi-Markov Conditional Random Fields

Semi-Markov CRFs are a subset of CRF that models segmentation jointly with sequence labelling[3]. These models allow for the integration of features that are not possible with traditional CRFs such as looking at a whole segment, and determining segment length and segment similarity[3, 15]. One deviation from traditional CRFs is that transitions from label to label can be non-Markovian, meaning that transitions can occur between non-neighboring states[15]. This is achieved by implementing a system whereby each state, s_i, persists for a period of time. After this time the model moves to a new state, s_j, which depends only on s_i. During the time before the transition, the model can behave in a non-Markovian way.
This model represents the input sequence, w, as a sequence of segments s=(s_1,s_2,...,s_n), and each segment being assigned a label, l_i[3]. When concatenated, all the segments should form the sequence w[3]. Given a feature function, g, that maps any given triple (j,x,s), where j is the position in the sequence, x is the input sequence and s is the label, to some measurement, then the function G(x,s) is a function that would count the number of segments in the sequence, x, with the label, s[15]. We also assume that each g^k is a function of x, s_j, and the label y_{j-1} to form an association with the previous segment[15]. Finally, W is a weight vector over all the components of G, and Z(x) is a normalization function where[15]:

$$Z(x) = \sum_{y'} e^{W \cdot G(x,s')} \tag{8}$$

This is a normalization function that ensures the model will make accurate predictions later on. Thus, a Semi-Markov CRF is an estimator of conditional probability of a given sequence, of the following form[15]:

$$p(s|x, W) = \frac{1}{Z(x)} e^{W \cdot G(x,s)} \tag{9}$$

ALTERNATIVE MODELS

Morphological Segmentation can be done using a variety of different models. Having discussed CRFs, we will now discuss one alternate Supervised Machine Learning Method, known as Sequence to Sequence models, and Unsupervised Machine Learning Approaches based on Morfessor Models.

Sequence to Sequence

Sequence to sequence models are an evolution from Deep Neural Networks (DNN) that are considered powerful Machine Learning models that perform well on tasks such as speech recognition[17]. However, they suffer from the problem that they are unable to deal with sequence inputs because DNNs require that the dimensions of the input are known fixed, but it is not possible to know how long an input is before it is processed[17]. One way in which to approach the problem is through the implementation of LSTMs, using one to read the input sequence, one time step at a time, to obtain a fixed sized vector representation of the input, and then use another LSTM to extract the output sequence from that vector[2, 17]. This architecture is known as an *Encoder-Decoder*, with the first LSTM being the encoder and the second being the decoder[2]. The Encoder takes in the input sequence, x and converts it to a vector c, known as the context vector, as follows[2]:

$$ht = f(x_t, h_{t-1}) \tag{10}$$

and

$$c = q(h_1, ..., h_{T_x}) \tag{11}$$

where h_t is a hidden state at time t, and f and q are some nonlinear functions [2]. The first equation meaning a function of the input at position t, and the hidden state one time step before. The second equation makes a vector using a function of all the hidden states produced by the input sequence. The Decoder is trained to predict the next label, y_t given the vector c and all the previously predicted labels[2]. The Decoder functions as a RNN conditioned on the input sequence, and it defines a probability over the label sequence y by using conditional probability with the current label, the context vector, and the vector of previously predicted words of the following form[2]:

$$p(y) = \prod_{t=1}^{T} p(y_t|y_1, ..., y_{t-1}, c) \tag{12}$$

Here, y is a sequence of labels. To calculate the conditional probability has the following form[2]:

$$p(y_t|y_1, ..., y_{t-1}, c) = g(y_{t-1}, s_t, c) \tag{13}$$

g here is a nonlinear function that outputs the probability of y_t given the previous labels and the context vector, and s_t is a hidden RNN state[2].

LSTMs were chosen due to their ability to learn on data with long range temporal dependencies, which is necessary due to the time taken for inputs to be converted to outputs[17]. Finally, a mechanism known as *attention* is implemented in this model due to the fact that previous hidden states need to be kept in memory because they are important to the deciding the label of the current state and thus generating the label sequence[2]. The decoder will decide which parts of the source sequence to pay attention to, which removes the burden of the encoder to have to encode everything into into the context vector[2].

Morfessor Models

These models are based on the unsupervised machine learning meaning that the data corpora that will be fed to these models will be unannotated. The job of the model is then to find patterns in the corpora from which it will make logical groupings.

We consider 2 methods of Unsupervised Machine Learning known as *Maximum Likelihood* and *Maximum Entropy*.

Maximum Likelihood models only consider the accuracy of the representation of the data with no regard for the model complexity, like model size[5]. Unless this model is modelled with some form of restrictive model search heuristics or model smoothing, then it tends to suffer from overlearning[5]. Overlearning is a problem faced by Machine learning algorithms where they learn so well from the training data, that it hinders their ability to generalize to new data[5].

Maximum entropy provides a framework for the estimation of probability distributions based on a set of training data[21]. The entropy of a specific distribution is a measure of uncertainty and it is maximized when the distribution is close to being a uniform distribution[21]. The principle of maximum entropy states that the only model that can be built from incomplete information is the one which has the maximum entropy subject to some constraints, this is because any other model would require assumptions that may change the nature of the model[21].

It is important to find a suitable model structure because this structure will set constraints on what the model can, or cannot learn, too restricting a model structure and close-to-optional models may be excluded[5]. In contrast too loose a model, and it will be harder to train because it will require large amounts of data and computational power[5]. We will be making use of *Morfessor*, which is a probabilistic model family designed specifically for Morphology Learning[5]. This model family consists of different components that can be interpreted to discover morphemes in words from a text corpora[5].

The task needed to be done is to induce the creation of a model of language, from only an unannotated text corpus[5]. We aim at finding the optimal model that will be able to generate a Morph vocabulary, a lexicon of morphemes, and a grammar[5]. A Lexicon is a database that contains one entry for every type of morpheme in the corpus, it is also able to store information such as relationships between morphemes[5]. A Grammar is a set of rules that governs how morphemes in the language of the corpus can be combined[5].

In an attempt to find the best model, Morfessor attempts to assign the best model using the conditional probability, p(M|Corpus) where **M** is the model being assigned[5]. This conditional probability is calculated in the following way:

$$P(M) = P(lexicon, grammar) \tag{14}$$

And

$$P(M|corpus) = P(corpus|M) \cdot P(M) \tag{15}$$

The first equation estimates the joint probability of the created lexicon and grammar, then this will be used to determine the probability of a model[5]. The second equation has to be maximized to find the best model, and the way it is modelled is by using the maximum likelihood estimate of the corpus based on the language model (M)[5].

DISCUSSION

CRFs are powerful models for the analysis of sequence data, from text corpora to images. That there are several possible ways to model CRFs is also advantageous to their performance with each way bringing its own benefits. Supervised Learning methods like CRFs and Sequence to Sequence models will be able to determine sequence labels through less data than unsupervised methods, but that data will have to

be annotated which may be harder to find. Comparatively, unsupervised methods will require more data, but that data will be unannotated which is generally easier to find.
Ruokolainen, T. *et. al.* conducted several comparison of CRFs and some other models, these models are ranked with 3 factors[14]: *Precision*: The percentage of True Positives over all positives identified by the model. *Recall*: The percentage of positives, that were correctly identified as positives. Finally, *F-Measure*, a number that combines and balances the two previous measures. The results of their comparisons are below:

Language	% Data Used	Precision	Recall	F-Measure
Arabic	25	95.5	93.1	94.3
	50	96.5	94.6	95.5
	75	97.2	96.1	96.6
	100	98.1	97.5	97.8
Hebrew	25	90.5	90.6	90.6
	50	94.0	91.5	92.7
	75	94.0	92.7	93.4
	100	94.9	94.0	94.5

Table 1: A summary of the CRF results discovered by Ruokolainen, T. *et. al.*[14]

The second column gives information about what percentage of the training data was used in training the model. These results show that the CRF performs well even when using only 25% of the training data, achieving over 90% for both languages. Using the F-measure as the comparison value, the CRF beat the other two models to which it was being compared at every point for data used. The models it was being compared to were a semi-supervised Morfessor implementation and a long-linear model. These results show that the CRF is able to perform well even when it is trained on limited amounts of data. Compared to the other models implemented in the study, at the 25% data usage point the semi-supervised Morfessor implementation achieved an F-measure of 79.2 and 77.6, where the log-linear model achieved 85.2 and 75.9, in Arabic and Hebrew respectively. According to these results the CRF provides an improvement of up to 15% at low data points. This is useful to us because we are dealing with low-resource languages that may not have access to much annotated data, so it will be beneficial to the results of future studies if we have a model that is able to perform well using minimal amounts of data. The discriminative nature of this model is also beneficial, because when running, this model will be less computationally heavy than a generative model would be because it will not have to do added middle steps for the Bayes Rules calculations.
Much research has been done into fields like named entity recognition, and speech recognition, but not much has gone into Morphological Segmentation which could make all those other tasks far simpler to design. By using CRFs we can enable a model to determine morphemes, and use these morphemes to break up a word and by doing so determine the meaning of the word, this means that even the meaning of new words, could theoretically be determined.

CONCLUSIONS

We presented CRFs, a discriminative probabilistic model for the labelling of sequence data. We also presented variations of it that each yield their own benefits and some form of improvement over traditional CRFs. We discussed the strengths of CRFs and why they would make a good model for use in Morphological Segmentation. We discussed Morphological Segmentation, and the different ways in which words can be segmented. Finally, we briefly discussed the notion of Sequence to Sequence Models and Unsupervised techniques based on Morfessor models. These represent the three models that we will be implementing, for comparison, in future. Future work to be done includes the implementation of the morphological segmentation tools for the Nguni Languages using all three models described in this paper. Following the implementation of the models, we will run comparisons of the models to determine which model was the best at the task, using various factors such as speed, the precision and the recall.
The value of this work lies in the ways that it could be extended for the further development of NLP tools for the languages being studied, and even more Low Resource Languages. Additionally, in the fact that it encourages the study of low-resource languages that previously have had limited studies being done about them.

ACKNOWLEDGEMENTS

The author thanks his family for their support throughout his studies. He thanks his friends for helping him get through it all. He thanks Jessica for everything. He thanks his project partners, Aaron and Sheldon, for their contribution and all the interesting work related conversations. This work is based on research supported in part by the National Research Foundation of South Africa (Grant Number: MND190922478537).

REFERENCES

[1] Andrew, G. 2006. A hybrid Markov/semi-Markov conditional random field for sequence segmentation. COLING/ACL 2006 - EMNLP

2006: 2006 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference. July (2006), 465—472. DOI:https://doi.org/10.3115/1610075.1610140.

[2] Bahdanau, D. et al. 2015. Neural machine translation by jointly learning to align and translate. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings. (2015), 1—15.

[3] Cotterell, R. et al. 2015. Labeled morphological segmentation with semi-markov models. CoNLL 2015 - 19th Conference on Computational Natural Language Learning, Proceedings. (2015), 164—174. DOI:https://doi.org/10.18653/v1/k15-1017.

[4] Cotterell, R. et al. 2016. A joint model of orthography and morphological segmentation. 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference. (2016), 664—669. DOI:https://doi.org/10.18653/v1/n16-1080.

[5] Creutz, M. and Lagus, K. 2007. Unsupervised Models for Morpheme Segmentation and Morphology Learning. ACM Transactions on Speech and Language Processing. 4, 1 (2007), 1—34. DOI:https://doi.org/10.1145/1187415.1187418.

[6] Do, T. and Artieres, T. 2010. Neural conditional random fields. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, PMLR. 9, (2010), 177—184.

[7] Lafferty, J. and McCallum, A. 2014. Conditional Random Fields. Computer Vision. 2001, June (2014), 146—146. DOI:https://doi.org/10.1007/978-0-387-31439-6_100233.

[8] Lample, G. et al. 2016. Neural architectures for named entity recognition. 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference (2016), 260—270.

[9] Ma, X. and Hovy, E. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers. 2, (2016), 1064-–1074. DOI:https://doi.org/10.18653/v1/p16-1101.

[10] Müller, T. et al. 2013. Efficient higher-order CRFs for morphological tagging. EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference. October (2013), 322—332.

[11] Murphy, K.P. 2012. Undirected Graphical Models (Markov Random Fields). Machine Learning: A Probabilistic Perspective. 661—705.

[12] Mzamo, L. et al. 2019. Evaluation of combined bi-directional branching entropy language models for morphological segmentation of isiXhosa. CEUR Workshop Proceedings (2019), 77-–89.

[13] Peng, F. et al. 2004. Chinese segmentation and new word detection using conditional random fields. (2004), 562–568. DOI:https://doi.org/10.3115/1220355.1220436.

[14] Ruokolainen, T. et al. 2013. Supervised morphological segmentation in a low-resource learning setting using conditional random fields. CoNLL 2013 - 17th Conference on Computational Natural Language Learning, Proceedings. 2008 (2013), 29-–37.

[15] Sarawagi, S. and Cohen, W.W. 2005. Semi-markov conditional random fields for information extraction. Advances in Neural Information Processing Systems. (2005).

[16] Sha, F. and Pereira, F. 2003. Shallow parsing with conditional random fields. June (2003), 134—141. DOI:https://doi.org/10.3115/1073445.1073473.

[17] Sutskever, I. et al. 2014. Sequence to sequence learning with neural networks. Advances in Neural Information Processing Systems (2014), 3104—3112.

[18] Sutton, C. and McCallum, A. 2011. An introduction to conditional random fields. Foundations and Trends in Machine Learning. 4, 4 (2011), 267-–373. DOI:https://doi.org/10.1561/2200000013.

[19] Sutton, C. et al. 2007. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. Journal of Machine Learning Research. 8, (2007), 693—723.

[20] Tseng, H. et al. 2005. A Conditional Random Field Word Segmenter for Sighan Bakeoff 2005. Proceedings of the fourth SIGHAN workshop on Chinese language Processing. X (2005), 168—171.

[21] Wallach, H.M. 2004. Conditional random fields: An introduction. Neural Computation. 18, 4–5 (2004), 1—9. DOI:https://doi.org/10.1162/jmlr.2003.3.4-5.993.

[22] Xue, J.H. and Titterington, D.M. 2008. on discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. Neural Processing Letters. 28, 3 (2008), 169-–187. DOI:https://doi.org/10.1007/s11063-008-9088-7.

[23] Zheng, S. et al. 2015. Conditional random fields as recurrent neural networks. Proceedings of the IEEE International Conference on Computer Vision. 2015 Inter, (2015), 1529-–1537. DOI:https://doi.org/10.1109/ICCV.2015.179.