Morphological Segmentation of Low-Resource Languages

Aaron Daniels
University of Cape Town, Department of Computer Science
Cape Town, South Africa

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

Sheldon Reay
University of Cape Town, Department of Computer Science
Cape Town, South Africa

KEYWORDS

Natural Language Processing (NLP); Computational Linguistics; Morphological Segmentation; Surface Segmentation; Canonical Segmentation; Supervised Machine Learning; Unsupervised Machine Learning; Nguni Languages; Low-resource Languages;

1 PROJECT DESCRIPTION

Morphological Segmentation is the Linguistic task of separating a word into its *morphemes.* Morphemes are the smallest building blocks of a language, that also have meaning [23]. This task is useful in the context of languages that are *agglutinative*, meaning languages in which the words are composed by aggregating morphemes without altering their spelling.

Low-Resource Languages are languages in which there is a lack of data, for a variety of reasons, such as they are languages unique to smaller countries or the language has a small group of speakers. The Nguni Family of Languages of Southern Africa is an example of Low-Resource Languages and they are also the focus of this project, with a specific focus on isiXhosa, isiZulu, isiNdelebe and siSwati. As a result of their status as a low resource language, there are few Natural Language Processing (NLP) applications, such as speech to text, that cater to them, and what NLP applications do exist have a tendency to be low accuracy.

This project will involve the morphological segmentation of the aforementioned languages through the implementation of Supervised and Unsupervised Machine Learning Models. Supervised Machine Learning involves the training of a model using data that is correctly labelled, the model then has to learn how to label unlabelled data based on information gained from the training data. Unsupervised Machine Learning entails the training of a model using unlabelled data, the model is able to learn by finding patterns in the data, and then it is able to apply these patterns to new data. The specific models being implemented will be Conditional Random Fields (CRF) and Sequence to Sequence (Seq2Seq), both of which are Supervised Models, and, Entropy Based and Likelihood based Unsupervised Models. The Supervised Models will be trained using annotated text corpora, and they will then be given a sequence of text which will have to be segmented into morphemes. The Unsupervised Models will have to perform the same task with an un-annotated text corpora. Both model types will then be evaluated on their performance throughout the task.

The development of these Morphological Segmenters can be useful for various reasons. They are useful in the development of NLP tools such as machine translation and text generation, because the segmenter can be used for pre-processing of the words being used as input to the tools. These tools could prove useful to students and speakers of the language, and linguists that are doing, or want to do work in the languages for which the segmenter is developed. The use of these languages is limited outside of the small communities of speakers, so developing these tools for the languages may also foster a reinvigorated use of the languages in the fields of education, business and in the growing online world.

2 PROBLEM STATEMENT & AIMS

There are a variety of ways in which Morphological Segmentation can be achieved. The first possible categorizations are **labelled** or **unlabelled**. If it is labelled, the substrings generated by the segmenting of the original sequence will each be given labels based on their function to the sequence or word in the sequence. These functions include amongst others root word, suffix and prefix. In unlabelled the substrings are simply segmented with no labels. Beyond this the word can be segmented using **Surface Segmentation** or **Canonical Segmentation**. Given a word, w. Using Surface Segmentation, w will be separated into a sequence of substrings which when aggregated will form the original word, w[21]. Using Canonical Segmentation w will be analyzed as a sequence of canonical morphemes, these morphemes will be based upon word forms that will have been learnt through the annotated data 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[21].

Given the word "*durability*":

- **Surface Segmentation**: *dur-abil-ity*
- **Canonical Segmentation**: *dur-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*.
This project aims to determine whether or not morphological segmentation can be successfully applied to the Nguni languages, using CRFs, Seq2Seq, and Unsupervised models. Specifically, we would like to determine which model is best able to accomplish this task based on the model's F1 score as a primary measure. Each member will implement a baseline model, and after implementing their planned model they will compare the results of their model against their baseline. To be certain our models represent a real improvement our models need to perform above their respective baselines in terms of F1 score.

Conditional Random Fields. This approach involves the implementation of three CRFs, to determine which one performs the best. A Traditional CRF to be used as a baseline against which to compare the other two implementations,a Neural CRF and a Semi-Markov CRF. These models will be tasked with unlabelled surface segmentation and labelled segmentation of the orthographic underlying representation. We hypothesize that the Neural CRF and Semi-Markov CRF will consistently outperform the Traditional CRF in terms of F1 score.

Sequence to Sequence. This approach involves implementations of different types of Seq2Seq models in order to reveal which yields the best results in terms of F1 Score. We hypothesize that a RNN-based Seq2Seq model with attention and a Transformer-based Seq2Seq model will perform better than a baseline made up of a standard Seq2Seq model.

Unsupervised. For the unsupervised approach, an entropy-based and likelihood-based model will be developed for unlabelled surface segmentation in the given languages. We will compare the performance of the models we develop to that of Morfessor-Baseline. Morfessor-Baseline is part of the Morfessor family of unsupervised morphological segmenters [19] developed by Creutz and Lagus. Morfessor-Baseline is the simplest model from the family which uses the minimum description length principle. It considers morph frequency but does not model context. We hypothesize that the unsupervised models will outperform the Morfessor-Baseline

3 METHODS AND PROCEDURES

Data Sets

The annotated training and testing data sets for the relevant languages will be sourced from the National Centre for Human Language Technology (NCHLT) Annotated Text Corpora, a publicly available resource[20]. This data set includes all the Nguni languages that are in the scope of the project.

Preprocessing. The data will need to undergo a standard preprocessing stage in order to be used for training of the models.

Supervised

The goal of the of the supervised aspect of this project is to improve current supervised learning models. This will be done by modifying and improving current models. We will be exploring two types of supervised models for this task: **Conditional Random Fields** and **Sequence to Sequence** Models.

Conditional Random Fields

CRFs are a class of Discriminative Probabilistic models used to segment and label sequence data, such as text corpora that will be the focus of this project[3]. Discriminative models are used to directly model the conditional probability $p(\mathbf{Y}|\mathbf{X})$, meaning the probability of $\mathbf{Y}$ given $\mathbf{X}$[3]. Discriminative Models are also able to learn direct mappings from an input to a label, giving them an advantage over Generative models[12]. Generative Models can do the same thing, but they require an extra step using Bayes Rules to get to the conditional probability step, which is the reason Discriminative models are widely considered to be superior[12]. Probabilities are also a large component of the model: The way the label sequence is assigned is through the maximization of probabilities of sequences until the best one is found.
CRFs take data sequences as input, and output label sequences with all the individual labels being defined in some predefined label alphabet. In this project the input sequence will be a sequence of text from a text corpora, and the output will be a sequence of labels. The words are segmented by reading in words character by character and each character is marked according to whether it is the start of a new segment, a part of the current segment, the end of the current segment, or a single character segment[23].
CRFs will be trained using an annotated text corpora, then given text sequences to be segmented and labelled using the probabilities learnt in the training stage.
There are three variations of the CRF that we will be looking at: Traditional CRFs, Neural CRFs and Semi-Markov CRFs.

Traditional CRFs. This is the base form of CRFs upon which the extensions are based. This model calculates the conditional probability of a sequence of labels given an input sequence using a linear chain that conforms to Markovian properties of label states being neighbours[13]. This model takes into account both the current label and the previous label.

Neural CRFs. This is a CRF that is augmented by implementing two Long-Short Term Memory Networks (LSTM), one that reads the sequence forward, known as a *forward LSTM* and one that reads the sequence backwards, known as a *backwards LSTM*[9]. These when used together create what is known as a Bi-Directional LSTM (BLSTM) develop a representation of each position in the sequence in context from both sides[9].

Semi-Markov CRFs. This subset of CRFs is different because transitions between labels can occur in a non-Markovian way[21]. Additionally, this model allows for features that traditional CRFs are unable to, such as looking at entire segments, determining segment length and segment similarity[21].

Implementation. The traditional CRF will be implemented based upon an existing PyTorch implementation and it will have to be extended to segment words. The neural CRF will be implemented based upon an existing model using PyTorch. It will have to be extended to segment words as opposed to tagging them. The semi-markov CRF will be an extension on the traditional CRF.

Analysis of the CRF Models.

1. Determine average performance of each CRF implemented through several runs based on different variables such as amount of data used in training and language amongst others. Score will be measured using Recall, Precision and F1 score.
2. Compare the performances to the traditional CRF Baseline
3. Compare the performances of the CRF implementations against each other
4. Compare the performances of the implementations against Sequence to Sequence and Unsupervised Models

Sequence to Sequence Models

Sequence to Sequence models (Seq2Seq) are an approach at using the power of Deep Neural Networks (DNN) for the tasking of mapping an input sequence to an output sequence. They were developed due to limitations of standard DNNs, which were unable to map a variable length input to a variable length output. The idea of sequence processing is especially important in language processing tasks as in many cases, the input length of the sequence will differ to the output length of the sequence, such as in Machine Translation. [11]

Sequence models frequently implement an encoder-decoder architecture. The encoder encodes the input sequence into a context-vector and the decoder uses this context-vector representation of the input sequence to produce an output sequence. [7] The encoder-decoder is made up with a number of recurrent neural network units (RNN). Lately, the usage of LSTMs and Gated Recurrent Units (GRU) has taken off due to allowing better long term dependencies over standard RNNs. [6, 11]
Attention is another idea which has closely been related to implementations of Seq2Seq models. Attention is used to dynamically determine which parts of the input sequence the model should focus on. This helps with long term dependencies between input words or characters. The use of an attention mechanism in different types of language processing tasks has led to an increase in performance accuracy. [5]

Transformers. Another widely used implementation of Seq2Seq models is the Transformer. The Transformer drops the usage of the RNN units and makes use solely of the attention mechanism. The performance accuracy of the Transformer has improved over RNN-based models at tasks such as machine translation. [1] Being a new model architecture, there are few existing implementations in Morphological Segmentation.

Implementation. A character-level sequence model will be implemented using existing models in PyTorch. The sequence-to-sequence models will consist of a RNN-based approach using LSTMs and a Transformer-based approach using the attention mechanism. These models will be based off existing implementations which deal with Parts-of-Speech tagging, another sequence problem. Both these approaches will be compared to a baseline of a non-neural CRF and a basic Seq2Seq model without attention.

Analysis of Models.

1. Evaluate the Word Accuracy and F1 Score using the test data via various runs over the data while adjusting hyper-parameters.
2. Determine the average performance of each type of Seq2Seq model implementation.
3. Compare the performance of the new implemented models to that of the baseline mentioned above.
4. Compare the performance of the implementations against CRFs and the Unsupervised models.

Experimental setup

Both Supervised Models will follow similar experimental setups: We will start by implementing and training the models. Then we will do some testing of the models which will aid us in optimising the hyper-parameters of each model. Once sufficiently optimised we will introduce the full training data to determine the recall, precision and F1 score. Upon completion of this we will be able to compare the various models against their individual baselines, and against each other.

Unsupervised

Two unsupervised approaches are considered: a branching entropy character-level recurrent neural network language model (BE) and a character-based Segmental Neural Language model with length regularization (SNLMR).

The BE model is predicated on the assumption that the predictability of the successive letter in a word increases as one moves through the letters in a word. An increase in predictability means a decrease in uncertainty. This uncertainty is measured as entropy which is expressed in terms of probabilities. A drastic increase in entropy implies that there is a morpheme boundary at a given position in a word.

SNLMR uses a neural network to generate a sequence of segments consisting of characters where each segment represents a possible morpheme. Each segment is generated character-by-character from a sequence model. This model defines a distribution over the input sequence as the marginal distribution over all morphological segmentations. Possible morphemes are generated by an LSTM through sampling a letter and feeding it back into this LSTM which gives the probability of a morpheme boundary. The model's parameters are optimized such that when summed over all morphological segmentation in the input, the marginal likelihood of the training data is maximized. Length regularization will be implemented to prevent overfitting to the training data set and thus poor generalization.

Experimental setup. We begin by preparing the training and testing data sets from the NCHLT Annotated Text Corpora. The surface segmentation will have to be derived heuristically through minimal edit distance between the input and the canonical segmentation provided in the corpora. Morfessor-Baseline will be implemented, trained, tested on all four languages, and its performance results will be used as a benchmark. The performance will be measured on the accuracy of the morpheme boundary or non-boundary classification, and the F1 score.

Implementation. Existing implementation will be adapted to develop the models. A character-based LSTM [15], will be trained and the entropy-based and likelihood-based objective functions will then be implemented on top of this in python with PyTorch. The result will be applied to segmentation. After development, models are tested to ensure they function as intended.

Analysis. Results from the evaluation are organized and formatted to prepare data for analysis. The purpose of this stage is to situate the models' performance relative to other models performing the same task and to draw insights about the nature of the task and given languages. To do that we will compare the performance of the following:

1. Entropy-based and likelihood-based models
2. Unsupervised models developed on various hyper-parameter configuration
3. Unsupervised models developed and the Morfessor-Baseline
4. Existing unsupervised models and those we have developed Seq2Seq and CRF and the unsupervised models developed

Evaluation

The corpora will be split into a *Training Set* and a *Testing Set*. During training, the models will be given more of the training set to learn from in increments of 25%. At each increment, the models will be fed testing data to assess its performance at the given amount of data. The described process is also applied to the baseline models. While our models do not outperform the respective baselines, model hyper-parameters will be fine-tuned to optimize performance if there is sufficient time. *Precision*, *recall* and *F1* score will be used to measure performance:

$$precision = \frac{TruePositives}{PredictedPositives}$$

$$recall = \frac{TruePositive}{TruePositive + FalseNegatives}$$

$$F1\ Score = 2 * \frac{precision * recall}{precision + recall}$$

Precision measures of the accuracy of predictions, indicating how many of predicted positives are positives. *Recall* measures how many positives the model is able to correctly predict. *F1 Score* is a function of precision and recall that will balance precision and recall because those can be thrown off.

4 ETHICAL, PROFESSIONAL AND LEGAL ISSUES

The data-set being used is a publicly available resource released under a Creative Commons license [4] allowing for fair use providing the source is correctly referenced. [20] The models will be constructed by modifying the open source code of selected existing implementations of morphological segmenters. The resultant constructed models will be compared to baselines which will be implemented from online resources such as PyTorch. We, therefore do not require any special permissions or legal engagement for the use of the data sets, code, and baselines.

It should be noted to future users of the software that Morphological Analysers do not guarantee a perfect and accurate measure of the morphological structure of a language, and any further use of the implemented models should take this into account.

5 RELATED WORK

In this section we have analysed how morphological segmentation has been achieved by others before us, making use of these models.

Conditional Random Fields

Ruokolainen, T. et al[23] approached the problem of Morphological Segmentation using Traditional CRFs in a Low Resource context. They compared theirs to a specialised Semi-Supervised version of Morfessor models, and in the vast majority of the test the CRF outperformed the Morfessor model. They noted that the CRF is suited to the task of Morphological Segmentation, provided that one has the necessary annotated data. Moreover, it was noted by them that CRFs are capable of yielding "state-of-art" results even when only using small amounts of data. The languages tested with these models were Arabic and Hebrew, which were both considered low resource at the time the paper was written.

Lample, G. et al.[9] used Neural CRFs for Named Entity Recognition through the use of BLSTMs. They compared their model to a variety of alternative models and the Neural CRF was either the first or the second best model in terms of F1 score. Whilst this is not the purpose for which we re using the Neural CRF we anticipate that the score will be similar in the Morphological Segmentation context. This paper tested the model on English, German, Dutch and Spanish.

Finally, Andrew, G[8] wrote a paper on semi-markov CRFs being used for Sequence Segmentation. As mentioned previously semi-markov CRFs are useful because they are able to incorporate features that traditional CRFs are unable to. Using a variation of the same semi-markov CRF model, but using different features. Overall the model was able to obtain an average F1 score of above the 94th percentile. This task is similar to the morphological segmentation we will be performing, and as such we believe that the results will be similar.

Overall, there have been several implementations of various forms of the CRF over time. These forms have been used for a variety of different tasks with the commonality between them all being that the model performs consistently and it performs consistently well.

Sequence to Sequence

Sequence-to-sequence models are a younger model architecture than CRFs. They were first implemented by Cho, K. et al. [7] and Sutskever, I. et al. in 2014 [11] and have since been successfully proven to deal with language tasks such as Machine Translation and POS tagging. [5, 11] When it comes to their usage for Morphological Segmentation, there are several instances where it has been successfully implemented. The first major attempt at canonical segmentation using Seq2Seq models was by Kann, K. and Cotterell, R. in 2016. [6] This implementation included a RNN-based encoder decoder with a mechanism of attention [5], as well a neural re-ranker to assist selecting the most probable candidates generated by the encoder-decoder. This approach, at the time, led to state of the art results.

Ruzsics, T. and Samardžić, T. [22] explored learning canonical morphological segmentation using a character-level RNN-based encoder decoder, supplemented by a language model trained over morphemes sequences. The application was focused on morphological rich languages in a low resource setting. Being compared to the model by [6] this approach eliminates the need for external resources outside of the provided corpora, which is required by the neural re-ranker. It performed better than the state of the art at the time without requiring additional resources thereby better suiting its application in low resource settings.

Unsupervised

Unsupervised segmentation is attractive as it is flexible and can accommodate changes in the language it models [2]. Also, it does not require annotated data sets for its training [2], eliminating the need for a human to prepare the data sets used [17] which can be costly and time-consuming.

The branching entropy model is partially inspired by the IsiXhosa Branching Entropy Segmenter (XBES-BE) in [17], except we will use a character-level Recurrent Neural Network language model. XBES-BE is used for morphological segmentation in isiXhosa and is comparable to the Long Short Term Memory (LSTM) surprisal approach in [14]. Models implemented in [17] were trained on the isiXhosa version of the South African Constitution, while NCHLT IsiXhosa Text Corpus was used as a testing data set. XBES-BE and its variants were compared to a random segmenter and Morfessor-Baseline. The models were evaluated on accuracy which measures how many boundaries and non-boundaries the segmenter identified correctly in word. XBES-BE achieved an accuracy of 66.9% which is significantly worse than Morfessor-Baseline's accuracy of 79.1%. However, it performed better than the random segmenter which had an accuracy of 60.72%.

In 2019, the authors reevaluated the performance of these models in [16] for the same task in [17] but included the F1 score in performance measurements. In this evaluation, XBES-BE improved to an accuracy of 71.6% with an F1 score of 55.3%, while Morfessor-Baseline achieved 77.2% and 48.9% on accuracy and F1 score, respectively. This study used similar training and testing data sets as in [17].

For the likelihood-based model, our efforts are motivated by those of Kawakami, Dyer, and Blunsom in [14], work which was done in 2019. Models in [14] were implemented for the task of Chinese and English word segmentation. The following corpora were split into training, validating, and testing data sets: Brent Corpus and English Penn Treebank for English, and Beijing University Corpus and Chinese Penn Treebank for Chinese. We shall implement an adaptation of the character-based Segmental Neural Language model (SNLM) with length regularization for our problem. However, we shall not include the variant of the model with lexical memory. On the same corpus, the SNLM with length regularization scored an F1 score of 49.5% whereas the LSTM surpisal model scored 55%. However, SNLM with lexical memory scored the best with an F1 score of 79.3%.

6 ANTICIPATED OUTCOMES

This section contains all the anticipated outcomes of the project including the software we aim to have deployed by the end of the project timeline, the impact that we anticipate it will have and the measures that we will use to determine whether or not the project can be considered a success.

Software

Given that the goal of this project is to compare different approaches to the task of morphological segmentation of the Nguni language group, models will need to be developed for this. This will be done with a combination of modifying existing model implementations, as well as implementing new ones where necessary.

Upon completion of the project, we anticipate that both supervised and unsupervised approaches will have been developed for the task of Morphological Segmentation. These models will be optimised with low resource languages in mind, specifically the Nguni family of languages as these are the focus of the project.

Challenges we anticipate include constraints in training of the models due to the limited amount of training data available, limited resources related to morphological segmentation of Southern African languages and familiarising ourselves with the new technology inherent with these models. It may also be beneficial for us to familiarise ourselves with basic elements of the languages for better understanding of the structure of the language and the results we will obtain. Finally, existing model implementations that we use may be limited which will make implementing them more difficult.

Impact

We hope that through the completion of this project, we will have contributed to the study of Morphology within the Nguni family of languages, as well as other languages with similar linguistic properties, namely being agglutinative or polysynthetic. We expect accurate results from our models, based on their F1 scores, and we hope that these results are helpful in the morphological analysis of this family of languages. These results will come from evaluating how accurately, using the F1 score, the models are able to segment the data. We expect that the resulting models will be useful to future researchers and the development of NLP tools.

The results of the models will also be useful insofar as they will add to the knowledge base about the effectiveness of Machine Learning approaches to morphological segmentation, and more generally the effectiveness of Machine Learning models in field of linguistics. This ability to automate linguistics work may allow us to learn more about the underlying structure of languages than ever before.

Key Success Factors

We will use a number of Key Success Factors in order to measure how successful the project has been:

- Whether all deliverables and milestones are met on time.
- Whether all models have been successfully implemented.
- The majority of the model implementations perform better than their respective baselines in terms of average F1 scores.

7 PROJECT PLAN

Risks

We have identified several risks associated with this project which are identified in **Appendix A**. Each risk has been defined using the following ways; The risk is described. It is then given a number from 0 to 10 corresponding to the probability of the risk occurring (0 being not possible, and 10 being definite). It is given a number from 0 to 10 corresponding to the impact that the risk occurring would have (0 being no impact, and 10 being critically impactful). Finally, we have identified a mitigation strategy, a way of monitoring whether the associated risk is going to occur and finally a management strategy for if the risk does occur.

Project Timeline

A Gantt chart that will assist with time management can be found in **Appendix B**. It is comprehensive and includes tasks that have already been completed. The chart also details all Milestones to be reached and important dates important throughout the project timeline. This chart is tentative and subject to change.

Resources Required

We have identified the following resources that will be needed over the course of this project. Access to PyTorch for use as a basis as well as for developing some of the models. Access to a cluster of Graphics Processing Units (GPUs) provided by UCT for use in training and testing the models once they have been implemented. We will also need access to some NCHLT text corpora that will be used to train the models once they have been implemented. We will need a stable internet connection and access to the UCT Virtual Private Network system to be able to make use of GPU resources. Finally, we will need access to computationally adequate computers to be used for implementing the model and software on those computers that will allow us to program the models.

Deliverables

The final list of deliverables along with their associated due dates can be found in **Appendix C**. The main deliverables include the writeup for the project and research as they progress. The list also includes deliverables for the code, the final report, presentations, and a poster.

Milestones

To ensure the project remains on track we have created several points by which we should have completed certain objectives which aligns with our Gantt Chart. These milestones can be found in **Appendix D**. These goals will allow the group members to know where the project is and what should have been completed by any stage in time, as well as what needs to be completed next. To avoid repetition we have left the official deliverables off of this list, but these deliverables also represent milestones for the project.

Work Allocation

All group members will work on the implementation of a baseline model for each of their models which will be used as a field of comparison for their models.

- Tumi Moeng will study Conditional Random Fields, specifically looking at unlabelled segmentation and labelled segmentation of the orthographic underlying meaning.
- Sheldon Reay will study Sequence to Sequence models, specifically looking at labelled segmentation and canonical segmentation.
- Aaron Daniels will study two techniques known as Entropy and Likelihood based models.

REFERENCES

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems, pages 6000–6010

[2] Burcu Can and Suresh Manandhar. 2014. Methods and Algorithms for Unsupervised Learning of Morphology. International Conference on Intelligent Text Processing and Computational Linguistics (2014), 177–205.

[3] Charles Sutton and Andrew McCallum. 2011. An introduction to conditional random fields. Foundations and Trends in Machine Learning 4, 4: 267–373. https://doi.org/10.1561/2200000013

[4] Creative Commons: https://creativecommons.org/licenses/by/3.0/legalcode

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. CoRR, ICLR 2015 abs/1409.0473, 2014.

[6] 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, Association for Computational Linguistics, Austin, Texas, 961–967.

[7] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bah- danau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 1724–1734.

[8] Galen Andrew. 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: 465–472. https://doi.org/10.3115/1610075.1610140

[9] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference, 260–270. https://doi.org/10.18653/v1/n16-1030

[10] Harald Hammarström and Lars Borin. 2010. Unsupervised learning of morphology. Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning 37, 2 (2010), 309–350. DOI:https://doi.org/10.1162/COLI_a_00050

[11] Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. Sequence to sequence learning with neural networks. In Advances in Neural Information Processing Systems, pages 3104–3112, 2014.

[12] Jing Hao Xue and D. Michael Titterington. 2008. on discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. Neural Processing Letters 28, 3: 169–187. https://doi.org/10.1007/s11063-008-9088-7

[13] John Lafferty and Andrew McCallum. 2014. Conditional Random Fields. Computer Vision 2001, June: 146–146. https://doi.org/10.1007/978-0-387-31439-6_100233

[14] Kazuya Kawakami, Chris Dyer, and Phil Blunsom. 2019. Learning to Discover, Ground and Use Words with Segmental Neural Language Models. (2019), 6429–6441. DOI: https://doi.org/10.18653/v1/p19-1645

[15] LSTM: https://github.com/salesforce/awd-lstm-lm

[16] Lulamile Mzamo, Albert Helberg, and Sonja Bosch. 2019. Evaluation of combined bi-directional branching entropy language models for morphological segmentation of isiXhosa. CEUR Workshop Proceedings 2540, (2019), 77–89.

[17] Lulamile Mzamo, Albert Helberg, and Sonja Bosch. 2019.Towards an unsupervised morphological segmenter for isiXhosa.Proceedings - 2019 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa, SAUPEC/RobMech/PRASA 2019 (2019),166–170.

[18] Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. Morphological and Phonological Learning: Proceedings of the 6th Workshop of the ACL Special Interest Group in Computational Phonology July (2002), 21–30. DOI:https://doi.org/10.3115/1118647.1118650

[19] Mathias Creutz and Krista Lagus. 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

[20] NCHLT Speech Corpus: https://sites.google.com/site/nchltspeechcorpus/

[21] Ryan Cotterell, Tim Vieira, and Hinrich Schütze. 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: 664–669. https://doi.org/10.18653/v1/n16-1080

[22] Tatyana Ruzsics and Tanja Samardzic. 2017. Neural Sequence-to-sequence Learning of Internal Word Structure. In Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Association for Computational Linguistics, Vancouver, Canada, 184–194.

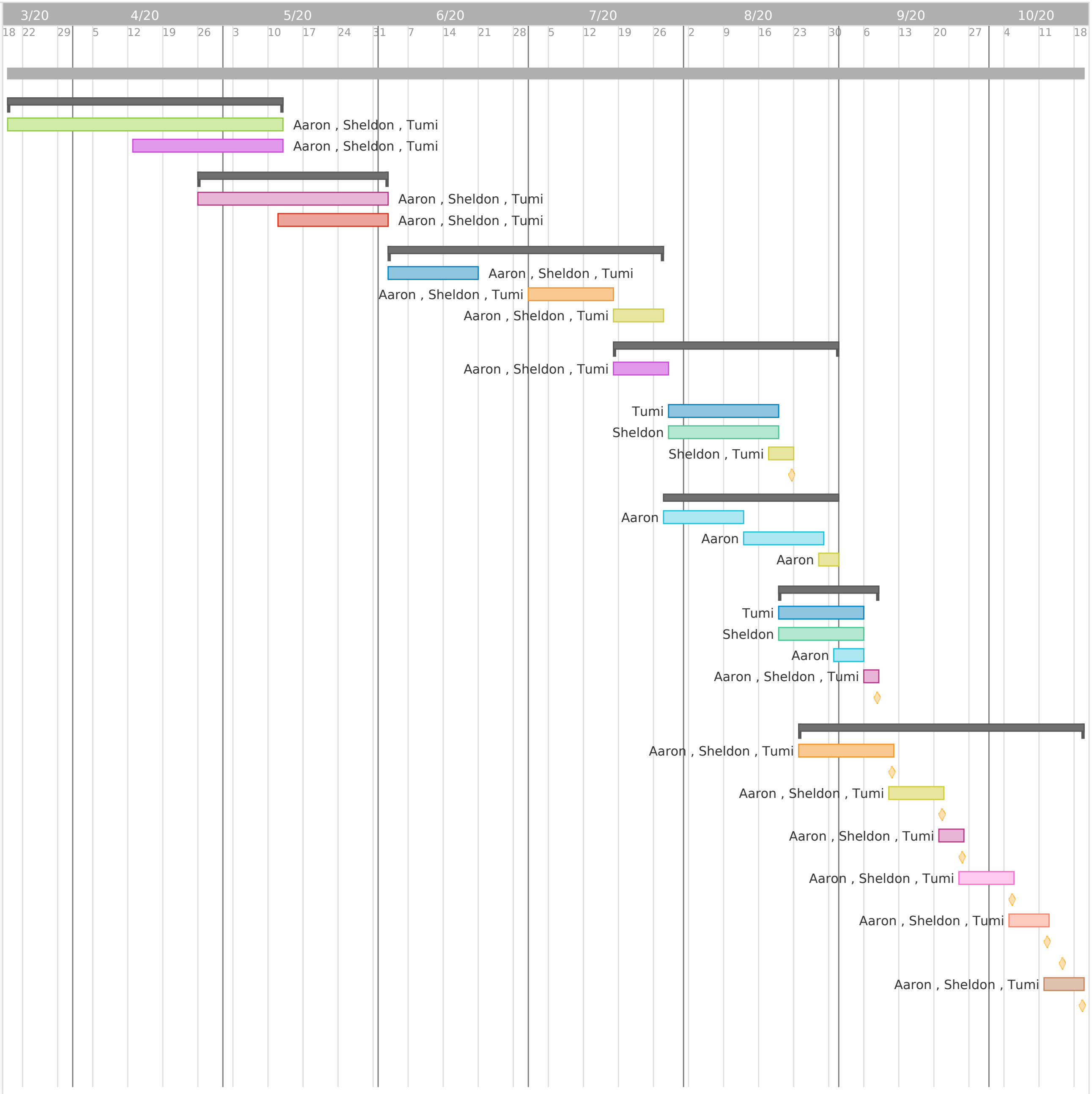
[23] Teemu Ruokolainen, Oskar Kohonen, Sami Virpioja, and Mikko Kurimo. 2013. Supervised morphological segmentation in a low-resource learning setting using conditional random fields. CoNLL 2013 - 17th Conference on Computational Natural Language Learning, Proceedings, 2008: 29–37.

APPENDIX A - RISK TABLE

Risk	Probability (0 - Low and 10 - High)	Impact (0 - Low and 10 - High)	Mitigation	Monitoring	Management
Unable to Access Resources for training	4	7	Ask project supervisor early on to ensure we are able to access the GPU and Text Corpora	No access to the GPUs and/or corpora by the time we start implementing the models	Make use of other online cloud based GPU services or alternative text corpora
Inability to Access Necessary Code Bases	4	8	Find suitable code bases early on	No suitable code bases identified by the time implementation starts	Ask project supervisor for recommendations and access to appropriate code bases
Poor Time Management by the Team	3	9	Start project work early to ensure time isn't too tight a constraint	Team members fall behind Gantt chart by more than 5 days	Team members that are ahead can assist those that appear to be falling behind
Long Model Training Times	3	6	Use a subset of available training data	Our models take longer to train than our baseline models	Ensure enough time is set aside for training model through Gantt Chart
Team Members Dropping Out	4	9	Team members should remain healthy both physically and mentally	Members of the team make little to no progress	The scope of the project would need to be decreased
Models Performing Worse than Baselines Leading to Incorrect Conclusions	4	8	Ensure our implementations are as close to theoretical implementations as possible, and as optimized as possible	Baselines performs better than our models in initial tests based on F1 scores	Review literature for where mistakes could have been made, check for optimization errors and ask project supervisor for advice
Lack of Communication Between Team and/or project supervisor	3	7	Ensure regular engagement between the team and project supervisor	More than 2 weeks pass by without interaction with project supervisor and/or team meeting	Re-engage team/project supervisor through a discussion of the project matter
Difficulty Implementing Models	6	9	Ensure we leave enough time for implementation, and ask for help when necessary	One week into time set for implementation with no progress being made	Ask project supervisor if there are aspects that can be scaled back for simpler implementation

APPENDIX B - GANTT CHART

Project	start	end
Literature review	**19/03/20**	**12/05/20**
Research	19/03	12/05
Writing	13/04	12/05
Project Proposal	**26/04/20**	**02/06/20**
Research	26/04	02/06
Writing	12/05	02/06
Initial Implementation	**03/06/20**	**27/07/20**
Initial Research	03/06	20/06
Model Research	01/07	17/07
Baseline Model Implementation	18/07	27/07
Model Implementation	**18/07/20**	**31/08/20**
Data Preparation	18/07	28/07
Supervised	-	-
Conditional Random Fields	29/07	19/08
Sequence to Sequence	29/07	19/08
Hyperparameter tuning	18/08	22/08
Complete Model Implementations	22/08	22/08
Unsupervised	**28/07/20**	**31/08/20**
Entropy-based	28/07	12/08
Likelihood-based	13/08	28/08
Hyperparameter tuning	28/08	31/08
Model Testing & Comparisons	**20/08/20**	**08/09/20**
CRF vs. Baseline	20/08	05/09
Seq-to-Seq vs. Baseline	20/08	05/09
Entropy || Likelihood Based vs. Baseline	31/08	05/09
Comparison of Models	06/09	08/09
Complete Testing	08/09	08/09
Final	**24/08/20**	**19/10/20**
Final report draft	24/08	11/09
Complete draft due	11/09	11/09
Final Report	11/09	21/09
Final paper due	21/09	21/09
Code Cleaning & Documentation	21/09	25/09
Code submission due	25/09	25/09
Project Presentation Preparation	25/09	05/10
Project Presentation Due	05/10	05/10
Web Page Development	05/10	12/10
Project web page due	12/10	12/10
School of IT Showcase	15/10	15/10
Poster Making	12/10	19/10
Project poster	19/10	19/10



APPENDIX C - DELIVERABLES

Deliverable	Due Date
Literature Reviews covering all three models	12/05/2020
Project Proposal	02/06/2020
Feasibility Demonstration	10/08/2020
Project Write Up Draft	11/09/2020
Final Project Write Up	21/09/2020
Code Submission	25/09/2020
Project Presentation	05/10/2020
Project Web Page Due	12/10/2020
Project Poster	19/10/2020

APPENDIX D - MILESTONES

Milestones	Due Date
Model Research Complete	17/07/2020
Baseline Models Successfully Implemented	27/07/2020
Data Preparation	28/07/2020
Models Successfully Implemented	19/08/2020
Hyper-parameter Tuning Complete	31/08/2020
Models Compared To Baselines	05/09/2020
Models Compared To Each Other	08/09/2020
Testing Completed	08/09/2020
Code Cleaning and Commenting	25/09/2020
School of IT Showcase	15/10/2020