Deep Learning for Morphological Parsing of Nguni Languages

Supervisor: Francois Meyer

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

Cael Marquard
mrqcae001@myuct.ac.za
University of Cape Town
Cape Town, Western Cape, South Africa

1 INTRODUCTION

In natural language processing (NLP), morphological parsing is the tagging of each morpheme in a word with its grammatical role [13]. A morpheme is the smallest unit of linguistic meaning into which a word can be split, while its respective tag is a description of its grammatical role [14, 22]. This information is important for further tasks like dependency parsing, translation, and text filtering [17]. For example, "aliqela" (meaning "they are several" in isiXhosa) is split into the morphemes "a-li-qela", which could be parsed as "a[RelConc6]-li[BPre5]-qela[NStem]". The morpheme "qela" is the word's noun stem in this example. The goal of the task is to predict these morpheme tags at the sentence level. Morphological parsers will be produced for all four Nguni languages: isiNdebele, siSwati, isiXhosa, and isiZulu [14].

Few parsers exist for the Nguni languages, none using deep learning methods [12, 38]. Morphological information is important for NLP in Nguni languages due to their linguistic features:

(1) Nguni languages are agglutinative, meaning many words are created by aggregating multiple morphemes [4, 24, 35].
(2) Nguni languages are written conjunctively, meaning that morphemes are concatenated into a single word [35]. 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".

Furthermore, the results from the research would help increase the resources for these four languages since they are low-resource and need more resources to be modelled for translation and filtering, as mentioned above [17, 21].

2 RELATED WORK

Morphological parsing refers to tagging each morpheme in a sentence with its grammatical role [13]. Full morphological parsing includes segmenting a word into its constituent morphemes and subsequently labelling them [31]. Therefore, morphological parsing can be seen as the last step in this pipeline.

Morphological segmentation is the task of splitting a word into its morphemes [24]. There are two types of segmentation: surface and canonical. Surface segmentation splits the word into the surface form of its morphemes which can be concatenated to form the word itself [24, 33]. Canonical segmentation extracts actual underlying morphemes, which may differ from the surface form due to sound changes [33]. The full canonical segmentation of a word may also include morphemes not visible on the surface, as shown in Table 1 [37].

Word	Surface	Canonical
zobomi	zo-bo-mi	za-u-bu-o-mi

Table 1: Difference between the surface and canonical segmentations of the same word.

Many common 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 [4]. Since the individual morphemes of a word combine to create the whole word's meaning, morphological features are important to NLP [31]. By developing improved tools for morphological parsing, these NLP tasks could be better solved by incorporating these features.

Because of the agglutinativity and conjunctive orthography of the Nguni languages, a vocabulary constructed from a text's morphemes may be more general than one constructed simply from whole words [7]. This could help models generalise to text containing words not seen in the same exact form in their training data [4, 24, 35].

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 [4]. This approach requires linguists to define grammatical and morphological rules for each language manually. Therefore, these models require a high degree of linguistic expertise, are time-consuming to make, and do not generalise well to dissimilar languages [8, 10].

2.1 Training from scratch

Morphological parsing for the Nguni languages has been implemented by Puttkammer and Du Toit [31] using an existing trainable pipeline known as MarMoT [25]. MarMoT uses Conditional Random Fields (CRFs), which make up a probabilistic model used to estimate the probability of a given labelling sequence for a given input sequence [18, 25]. MarMoT considers features such as the input's bi-gram representation and other surface features of the surrounding text [25]. It does not incorporate any features generated by a neural network. Morphological parsing is an example of a sequence labelling task. In NLP, there are two prominent sequence labelling tasks: named entity recognition (NER) and part-of-speech (POS) tagging.

NER refers to the identification of sequences in text such as names (proper nouns) and numeric expressions [26]. It relates to morphological parsing in that they can both be treated as sequence-labelling tasks [19]. Lample et al. [19] apply bidirectional Long Short-Term Memory networks (bi-LSTMs) and CRFs with bi-LSTM features.

POS tagging refers to the task of labelling each word in a sentence with its lexical category, thus classifying the grammatical role it

plays [29]. This is similar to morphological parsing but operates at a word level. Pannach et al. [29] implement part-of-speech tagging for the Nguni languages with an approach similar to the one taken by Lample et al. [19] for NER.

Given that CRFs and bi-LSTM models have been effective in solving related tasks, these architectures are likely to yield good results when applied to morphological parsing. Notably, CRFs with neural features (e.g. bi-LSTM features) have not yet been evaluated for this task.

Sequence-to-sequence models based on Transformers [38] and bi-LSTMs have also been used to solve the related problem of canonical morphological segmentation by Moeng et al. [24]. Additionally, Akyürek et al. [2] achieved good results in applying a bi-LSTM-based, encoder-decoder model to the task of morphological parsing for European languages.

2.2 Fine-tuning

Fine-tuning PLMs and transfer learning have long been explored in NLP and progress is notable with the advent of the Bidirectional Encoder Representations from Transformers (BERT) by Devlin et al. [12] and the Generative Pre-trained Transformer (GPT) by Radford et al. [32] in 2018. Since then, multiple tasks have been modelled from these to outperform machine learning models trained on the task from scratch. Models like multilingual BERT (mBERT) [12, 15] and cross-lingual RoBERTa (XLM-R) [5] have been developed and exhibit excellent performance for downstream tasks such as NER [1] and POS tagging [15].

Similar tasks have been pursued in African languages by the Masakhane group[1] in two sequence labelling tasks: MasakhaPOS by Tsarfaty et al. [37] and MasakhaNER by Adelani et al. [1]. Two of the Nguni languages, isiXhosa and isiZulu, were utilised in the fine-tuning of the models and it was discovered that PLMs performed better than competing convoluted neutral networks like bi-LSTMs. However, it is noted that the languages' absences [21] in pre-training corpora especially lowered performance in models like AfriBERTa [27].

PLMs have also been applied to morphological analysis in other languages. Inoue et al. [16] make use of the PLMs for morphosyntactic tagging and one of their key findings was that while being low-resourced, fine-tuning the PLMs with all the dialects and then the target dialect produced better results than just fine-tuning on the target dialect. McCarthy et al. [23] explored the shared task of morphological inflection in 2019, and the project involved multiple teams. The best results came from the two teams that employed PLMs, namely mBERT and BERT, illustrating the advantage of fine-tuning and transfer learning of embeddings [15, 23] over the traditional neural networks present.

3 RESEARCH QUESTIONS

The principal research aim of the project is to develop new morphological parsers for each of the Nguni languages and compare the performance of different deep-learning techniques for this task. This will assist in the enhancement of downstream NLP tasks [4, 17].

The main research question to be answered is as follows:

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

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 2: Three examples from the isiXhosa part of the dataset collected by Gaustad and Puttkammer [14].

(1) **Can the selected neural models outperform current, state-of-the-art probabilistic models on the task of morphological parsing for all of the Nguni languages?**

This question will help to assess the feasibility of using neural models in morphological parsing for the Nguni languages. Each model developed will be evaluated on the same metrics—precision, recall and F_1—as described in Section 4.3. The aim is to develop models that will perform the task successfully for augmentation of the morphological data for these languages. Therefore, using these key NLP metrics allows them to be objectively compared to other models.

Aside from that, there are some other questions that should be answered from the research:

(2) **Do the selected pre-trained models or the selected models trained from scratch perform better for Nguni morphological parsing?**
(3) **Do the models perform better when trained on surface segmentations or canonical segmentations of the words?**

The hypotheses for the research questions are as follows:

(1) **Neural models will outperform probabilistic models.** This is predicted because neural models have outperformed probabilistic models on related NLP tasks.
(2) **Pre-trained models will outperform models trained from scratch.** The reasoning for this is discussed further in subsection 2.2.
(3) **Models trained on canonical segmentations will outperform those trained on surface segmentations.** This is predicted because canonical segmentation provides more information to the model than surface segmentation.

4 METHODS

4.1 Data Preparation

The corpus collected by Gaustad and Puttkammer [14] will be used to train the model. It includes the canonical segmentation and tags for each morpheme in each word, as seen in Table 2.

The final model will be organised into a pipeline, as per Figure 1, so that each portion is substitutable. First, the sentence will be read. Next, it will be segmented. Lastly, the segmentation will be fed into the morpheme classification/parsing model to produce a final output.

Splitting the segmentation and parsing model has many advantages. Firstly, by simplifying the task that the model must learn to achieve, there is a higher chance of the parameters converging quickly, and the resulting model being smaller in size. Secondly, because the parsing model will be independent of the segmentation model, its internal segmentation performance will not limit it. If a

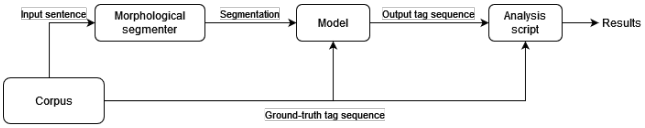


Figure 1: The trainable morphological analysis pipeline, which allows the morphological segmenter and parsing model to be swapped out. We are developing the parsing model part of the pipeline.

better segmenter is developed later, it may easily be substituted in to improve performance. Lastly, this simplifies the development of the model itself.

There are various ways in which a pre-existing segmenter may be incorporated into the parsing pipeline. We will investigate three options:

(1) **The model is trained on the gold-standard canonical segmentation.** While this would *not* result in a complete pipeline from sentence to morpheme tags, it would allow the accuracy of each classification model to be determined independent of the segmentation quality.
(2) **The model is trained using the canonical segmentation model.** This is similar to the previous case but yields a full pipeline usable on unseen text. It therefore may perform better on unseen text by using consistent segmentation during training and evaluation. It may perform worse, especially on text in the dataset, due to the poorer segmentation model performance compared to the gold standard.
(3) **The model is trained using the surface segmentation model.** This approach has the advantage that the surface segmentation developed by Moeng et al. [24] is very accurate. However, the model will not be able to use any information present only in the canonical segmentation of the word.

4.2 Operating the Models

Given that the project's basis is applying deep learning methods to morphological parsing, significant computational resources provided by graphical processing units (GPUs) will be required to handle the models' training. This processing power has been requested and acquired from two separate high-performance computing (HPC) clusters: the National Integrated Cyberinfrastructure System's Centre for High-Performance Computing (NICIS CHPC)[2] and the University of Cape Town's HPC cluster[3]. The primary cluster will be the NICIS CHPC, with UCT's serving as a backup in case of any issues.

The project is split into two parallel explorations for each team member.

4.2.1 Subtask 1 - Training models from scratch. Both sequence labelling and sequence-to-sequence approaches have been applied to the task of morphological parsing.

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

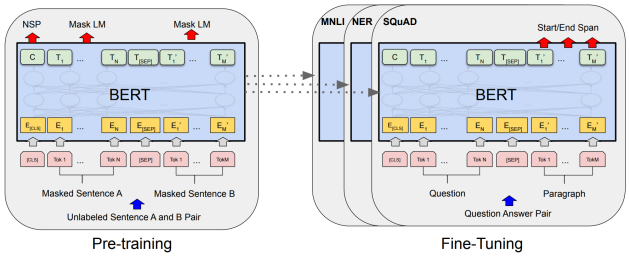


Figure 2: The operation of BERT illustrates how the model is first pre-trained as an MLM and then fine-tuned by a classification layer for various natural language understanding tasks like NER and POS tagging [12].

Sequence labelling models such as Conditional Random Fields (CRFs) and bi-directional Long Short-Term Memory networks (bi-LSTMs) have been effective at solving related problems in NLP, including in the Nguni languages, but have not yet been extensively applied to morphological parsing [19, 25, 29, 31]. While CRFs with statistical features have been used for morphological parsing of the Nguni languages, CRFs with bi-LSTM features have not yet been applied to this task [31]. Therefore, CRFs with bi-LSTM features and bi-LSTM sequence labelling models will be developed for the morphological parsing task.

Sequence-to-sequence models such as Transformers and encoder-decoder models have also been applied to similar problems in NLP [2, 24, 38]. Therefore, a Transformer-based model will also be developed.

The models will be implemented in Python using the PyTorch library [30].

4.2.2 Subtask 2 - Fine-tuning pre-trained models. This will make use of the vast library of pre-trained NLP models (PLMs) available on the Hugging Face library[4]. The selection pool is from the BERT-based [12] models like RoBERTa [5, 6, 20] and AfriBERTa [27, 28] with the actual selections being:

(1) XLM-R by Conneau et al. [5]: a large scale cross-lingual PLM
(2) AfroXLMR by Alabi et al. [3]: XLM-R further pre-trained on 20 African languages including isiXhosa and isiZulu.
(3) Nguni-XLMR: XLM-R adapted for the 4 Nguni languages for part-of-speech tagging.

The training data is supplied to the model for fine-tuning in a machine learning paradigm known as "transfer learning" [12, 15, 36]. PLMs like BERT are fine-tuned by adding a new task-specific classification layer to an MLM pre-trained model, as shown in Figure 2. All the work will be done in a Python environment with Jupyter Notebooks. Once each model is pre-configured to fit the task's requirements, the code will be gathered into executable files for the CHPC clusters and trained on the cluster. After each training, the model will be evaluated according to the evaluation parameters as explained in Section 4.3. The model's hyper-parameters will be fine-tuned for the new task on each training iteration to optimise the performance of the models on the F_1 score.

[4]https://huggingface.co/models

4.3 Evaluation

For each model developed in sections 4.2.1 and 4.2.2, evaluation scripts will be designed to assess performance on three metrics: precision, recall and F_1. Precision is the proportion of positive identifications that were correctly identified [11]. In the case of morphological parsing, this is the number of correctly identified tags in proportion to all the tags in the predicted output. Recall is the proportion of the correctly identified tags to all the tags in the expected output [11]. Both precision and recall can mathematically be described in terms of true and false positives and negatives (TP, FP, TN, and FN), as shown in Equation 1 and Equation 2.

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

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

F_1 results from combining precision and recall into a harmonic mean given by Equation 3 [9, 34]. Since precision and recall are both important metrics of a model's performance, it is necessary to balance them into a single objective metric that reflects them equally.

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

For each of the metrics, a higher score would indicate better performance as measured by that metric and would help compare the parsing ability of each one relatively.

5 ETHICAL CONSIDERATIONS

The most prominent ethical consideration is the issue of copyright. The SADiLaR dataset [14] acquired for this research is publicly available under the Creative Commons Attribution 4.0 International license, meaning it is viable for adaptation and redistribution in the project. Furthermore, the code bases for the project, including the entirety of the Hugging Face models, are open source and available for use. Lastly, to encourage further engagement for the task, the results (models and code base) produced here are intended to be open source.

6 ANTICIPATED OUTCOMES

The primary anticipated outcome of the project will be the trained models themselves, which will be released publicly for further research and application. A by-product of this will be the source code for preparing the data and training the models. A secondary outcome will be the analysis of the models' performances. Lastly, a final report will be produced, discussing the project as a whole, including the relevant prior work, methodology, model architectures, performance analysis, and a critical discussion of how well the aims were met and how the models compare with each other and previous work.

The models themselves may be evaluated by calculating their F_1 scores for the task of morphological tag prediction, as discussed in subsection 4.3. A model is successful if its performance exceeds the baseline probabilistic model on the same task. Overall, the premier research question may be answered affirmatively if at least one of the resulting models outperforms the baseline.

Additionally, the training and runtime performance of the models will be considered. While this is not a key focus area, the models should neither be prohibitively expensive to train nor too slow to use.

If the project is successful, then the most successful models could be used to provide context to other Nguni language models to improve their performance on downstream tasks, such as question answering, text summarisation, and machine translation. Additionally, comparing different model architectures and their performance will assist in determining which architectures are best suited to similar tasks, such as POS tagging.

7 PROJECT PLAN

7.1 Risks and Management

The key risks affecting the project are shown in Table 3.

Risk	Probability	Impact	Mitigation Strategy
HPC cluster downtime impeding progress	4	7	Weekly checking of the cluster's status to plan ahead of time for cluster availability.
Scope creep - more and more kinds of models are planned.	6	5	Continual re-assessment of priorities on which models would most likely be the most promising.
Loadshedding disrupts development of model source code.	8	3	Use of laptops, inverters, and judicious planning using the EskomSePush[5] app.
Team member drops out due to sickness or other reasons.	2	5	Splitting the project up into two disjointed parts, permitting individuality within the team.
Models requiring more time than expected to train.	5	9	Implementing the first models as soon as possible to extrapolate the time needed for training and adjusting the plan if needed.
Trained models being over-fitted to the dataset and not working well on real-world data.	4	7	Training the models with the canonical and surface segmentations, and employing techniques like dropout and early-stopping.

Table 3: Risk matrix for the project

7.2 Timeline

The project timeline is represented as a Gantt chart in Appendix A.

7.3 Required Resources

The principal resources for the project are the dataset, computers for development, and access to a high-performance computing cluster for model training.

The dataset [14] is readily available for download, and is distributed under an open license.

The computers will be used to write the code for the models. Desktop computers are readily available in the Honours Lab 24/7 and both members additionally own personal laptops to use during the project.

Access to the NICIS CHPC and UCT HPC clusters has already been granted to both team members under the supervisor's (Francois Meyer) project. Access has been granted for periods of one year and six months respectively giving the project ample time for training. Both team members have access to the internet via UCT's eduroam and home Wi-Fi networks which allows for ssh access into both clusters.

7.4 Deliverables

The project deliverables consist of the following items:

- **Review of literature (individually submitted)**. A review of the related works in the subject area produced by each team member to illustrate understanding and demonstrate the project's significance.
- **Project proposal (this document)**. The project proposal must be submitted and approved before the project can commence.
- **Prototype demo.** Near the beginning of the 3rd block, the work-in-progress project will be showcased.
- **Final report.** A report documenting the project overview and activities. This will include an introduction to the research area, a review of the literature, a discussion of the methodology and model architectures, and a discussion of results.
- **Final demo.** After the final report is submitted, the operation of the final models will be demonstrated.
- **Poster.** A poster giving a high-level summary of the project must be created.
- **Project web-page.** A web page must be developed to explain the project in more detail for the Computer Science department showcase.
- **Source code and trained models**. All source code relevant to the project (e.g. data preparation, training of models, and evaluation scripts), and the models themselves, will be delivered to the supervisor. Documentation for the code will also be provided extensively

7.5 Milestones

Key milestones in the project include:

- Project proposal
- Implementation of the first gold-standard-segmentation-based models (pre-trained and from scratch). This will include developing the generic training pipeline and data pre-processors. At this point, the other models that are planned will be prioritised accordingly to ensure that there is enough to present at the first demo.
- First/Half-point demonstration. After this, feedback will be incorporated and the plan for the rest of the project will be revised accordingly.
- Completed implementation of all models. Once the selected models are all implemented, they will then be analysed. More emphasis in the analysis will be placed on models which perform particularly well or poorly, as well as those which yield unexpected results.
- Final report first draft. Needed for feedback from the supervisor on formatting.
- Presentation of the project at School of IT showcase.

7.6 Work Allocation

While the project is split into two parallel streams for the two different approaches, there are shared tasks among the team members. As discussed earlier, the dataset must be pre-processed into a form that is a common input for both approaches. We will also need to assess the performance of the models following the project's metrics. Simbarashe will code scripts to execute these two tasks for common use.

The final report paper and website are additional collaborative efforts needing to be shared between the team members. Since the sections of the report are not rigid, the work allocation on the report will be established closer to the time.

The main split of the work is the development of the models. Cael will train models from scratch from several available NLP models while Simbarashe will fine-tune pre-existing PLMs from the Hugging Face library. The team members are expected to mould each model to fit within the common training pipeline.

REFERENCES

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

[2] 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

[3] Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. 2022. Adapting Pre-trained Language Models to African Languages via Multilingual Adaptive Fine-Tuning. In *Proceedings of the 29th International Conference on Computational Linguistics*, Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen

Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na (Eds.). International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 4336–4349. https://aclanthology.org/2022.coling-1.382

[4] 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

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

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

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

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

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

[10] 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

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

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

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

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

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

[16] Go Inoue, Salam Khalifa, and Nizar Habash. 2022. Morphosyntactic Tagging with Pre-trained Language Models for Arabic and its Dialects. In *Findings of the Association for Computational Linguistics: ACL 2022*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 1708–1719. https://doi.org/10.18653/v1/2022.findings-acl.135

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

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

[19] 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

[20] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs.CL]

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

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

[23] Arya D. McCarthy, Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sabrina J. Mielke, Jeffrey Heinz, Ryan Cotterell, and Mans Hulden. 2019. The SIGMORPHON 2019 Shared Task: Morphological Analysis in Context and Cross-Lingual Transfer for Inflection. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, Garrett Nicolai and Ryan Cotterell (Eds.). Association for Computational Linguistics, Florence, Italy, 229–244. https://doi.org/10.18653/v1/W19-4226

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

[25] 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

[26] 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

[27] Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. 2021. Small Data? No Problem! Exploring the Viability of Pretrained Multilingual Language Models for Low-resourced Languages. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 116–126. https://aclanthology.org/2021.mrl-1.11

[28] Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. 2021. Small Data? No Problem! Exploring the Viability of Pretrained Multilingual Language Models for Low-resourced Languages. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 116–126. https://aclanthology.org/2021.mrl-1.11

[29] 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

[30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[31] Martin Puttkammer and Jakobus Du Toit. 2021. Canonical Segmentation and Syntactic Morpheme Tagging of Four Resource- scarce Nguni Languages. *Journal of the Digital Humanities Association of Southern Africa (DHASA)* 3 (01 2021). https://doi.org/10.55492/dhasa.v3i03.3818

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

[33] 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

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

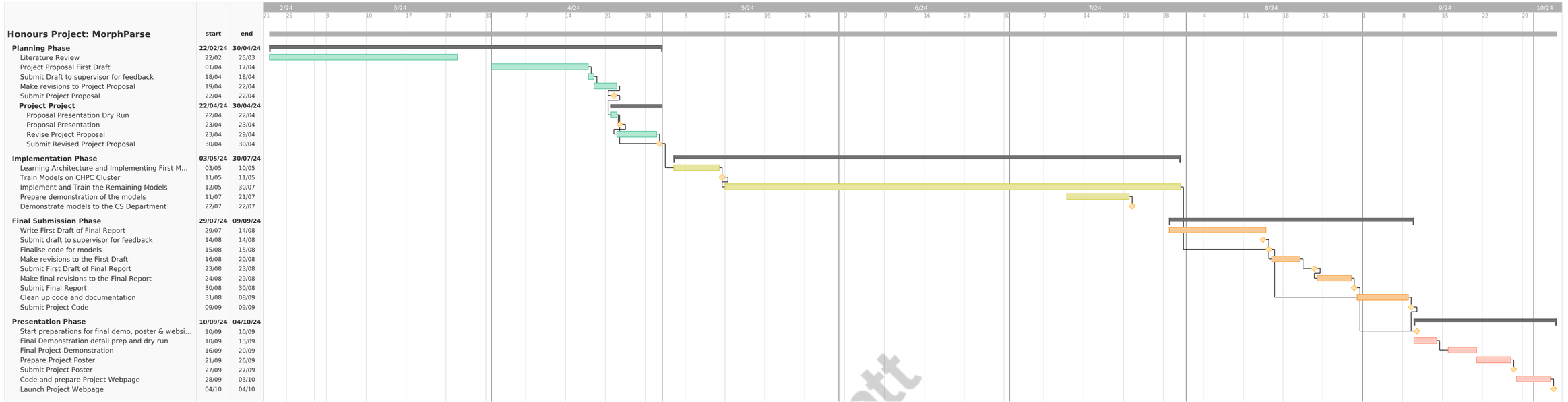
[35] Elsabe Taljard and Sonja Bosch. 2006. A Comparison of Approaches to Word Class Tagging: Disjunctively vs. Conjunctively Written Bantu Languages. *Nordic Journal of African Studies* 15 (01 2006). https://doi.org/10.53228/njas.v15i4.37

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

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

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

A PROJECT TIMELINE



Honours Project: MorphParse	start	end
Planning Phase	**22/02/24**	**30/04/24**
Literature Review	22/02	25/03
Project Proposal First Draft	01/04	17/04
Submit Draft to supervisor for feedback	18/04	18/04
Make revisions to Project Proposal	19/04	22/04
Submit Project Proposal	22/04	22/04
Project Project	**22/04/24**	**30/04/24**
Proposal Presentation Dry Run	22/04	22/04
Proposal Presentation	23/04	23/04
Revise Project Proposal	23/04	29/04
Submit Revised Project Proposal	30/04	30/04
Implementation Phase	**03/05/24**	**30/07/24**
Learning Architecture and Implementing First M...	03/05	10/05
Train Models on CHPC Cluster	11/05	11/05
Implement and Train the Remaining Models	12/05	30/07
Prepare demonstration of the models	11/07	21/07
Demonstrate models to the CS Department	22/07	22/07
Final Submission Phase	**29/07/24**	**09/09/24**
Write First Draft of Final Report	29/07	14/08
Submit draft to supervisor for feedback	14/08	14/08
Finalise code for models	15/08	15/08
Make revisions to the First Draft	16/08	20/08
Submit First Draft of Final Report	23/08	23/08
Make final revisions to the Final Report	24/08	29/08
Submit Final Report	30/08	30/08
Clean up code and documentation	31/08	08/09
Submit Project Code	09/09	09/09
Presentation Phase	**10/09/24**	**04/10/24**
Start preparations for final demo, poster & websi...	10/09	10/09
Final Demonstration detail prep and dry run	10/09	13/09
Final Project Demonstration	16/09	20/09
Prepare Project Poster	21/09	26/09
Submit Project Poster	27/09	27/09
Code and prepare Project Webpage	28/09	03/10
Launch Project Webpage	04/10	04/10