Comparison of Language modelling techniques for Nguni Languages

Luc Hayward
University of Cape Town
Cape Town, South Africa
hywluc001@myuct.ac.za

Stuart Mesham
University of Cape Town
Cape Town, South Africa
stuart.mesham@gmail.com

Jared Shapiro
University of Cape Town
Cape Town, South Africa
jar.shapiro@gmail.com

CCS CONCEPTS

• **Computing methodologies** → Neural networks; **Machine translation**; • **Information systems** → *Language models.*

KEYWORDS

language modelling, neural networks, low-resource languages, LSTM, N-Gram, Transformer

1 PROJECT DESCRIPTION

Natural language processing has many applications in areas such as information retrieval, voice recognition, machine translation, spelling correction and question answering [1, 6, 9, 13]. A prerequisite for these applications is the development of a language model. Language models assign a probability to a sequence of words [4]. More formally, a language model is defined as a probability distribution of a set of strings over a given context [2, 13]. Recent advances in language modelling have yielded significant improvements in performance, however, newer models are typically trained on large, high quality datasets which are not available for most South African languages. In this project, we aim to determine whether these advances, which have led to performance improvements in English, will also improve performance in low-resource South African languages. We plan to evaluate three major classes of language models: n-grams, LSTMs and Transformers.

N-gram language models are a traditional class of language models that have historically been the best performing. We plan to implement n-gram models with smoothing and interpolation algorithms including the modified Kneser-Ney algorithm [2].

Long Short-Term Memory (LSTM) language models are a class of artificial neural networks and have recently pushed the state of the art in language modelling. While LSTMs show a clear performance improvement over n-grams in a high-resource setting, it remains to be observed whether LSTMs will yield an improvement in the context of low-resource South African languages. We plan to evaluate a type of LSTM called the AWD-LSTM [8].

Transformer models are another class of language models based on artificial neural networks which have more recently achieved state-of-the-art results. Transformer models are also typically trained on large datasets and thus their performance on low-resource languages is not yet thoroughly explored. We plan to evaluate the GPT-2 transformer model [12].

2 PROBLEM STATEMENT AND RESEARCH QUESTIONS

African languages are typologically[1] very different to the languages typically studied for language modelling. African languages can be classed as morphologically rich languages [11], languages in which grammatical relations (such as Subject, Object, etc) are indicated by changes in the words rather than the relative position of words in the sentence. Moreover african lanugages are agglutinative, wherein the words within the language are made up by combination of smaller morphological units. This leads to potentially very large vocabulary sizes where each word appears relatively few times, in spite of the "sub-words" being more prevalent across the vocabulary.

The goal of the project is to compare the performance of three classes of language models - N-grams, LSTMS, and Transformers. We seek to evaluate whether the same techniques which have led to improvements in English can lead to improvements in South African languages. The following specific research questions are posed:

1. Which of the three classes of language models - N-grams, LSTMS, and Transformers - achieve the best results when evaluated intrinsically on entropy?
2. For each model class, to what extent do the different modelling choices matter?

In addition to the typological differences, the relatively small datasets available for these languages may make training more difficult for the newer model architectures. The LSTM and Transformer architectures are designed for large datasets and have a large number of parameters, following the recent trend in neural language modelling where larger models are trained on larger datasets to achieve improved performance. In a low-resource setting, it is difficult to train large models without over-fitting the training data. As such, careful regularisation, hyperparameter tuning and optimization strategies are necessary to achieve optimal performance with each type of model.

3 PROCEDURES AND METHODS

There are several main procedures required for the successful completion of the project. First, datasets for each language will need to be obtained and suitably cleaned. Open source implementations of each type of model will then be adapted for training and evaluation on South African languages. Model hyperparameters will need to be optimised, requiring many training runs. Finally, the resulting models will be evaluated based on their perplexity scores on a test set for each language class, Nguni and Sotho-Tswana to

[1] Typology refers to the linguistic properties and characterization of a language.

determine whether a significant performance improvement over existing methods has been achieved.

3.1 Datasets

Following standard practice for research in language modelling, the datasets will be split into training, validation and test sets. In order to effectively evaluate the performance of each type of language model we require South African language corpora of sufficient size and quality. Below we outline the properties of existing datasets that we have identified as well as the potential for dataset combination. A comparison of the various dataset sizes is available in table 1.

3.1.1 NCHLT Text Project. The South African Centre for Digital Language Resources (SADiLaR) [2] provides monolingual corpora for all 11 of South Africa's official languages. Corpora for Sesotho, Sepedi, isiZulu, Siswati, Setswana, isiNdebele, tshiVenda and isiXhosa were collected for the 2014 National Centre for Human Language Technology (NCHLT) Text project [5]. A significant proportion of the texts are scraped from governmental websites. The corpora range in size from 1 to 3 million tokens.

3.1.2 MuST Corpora . Previous work on South African languages by Scarcella [13] made use of proprietary corpora prepared by the Multilingual Speech Technology Group (MuST) [3] from North West University. Multiple sources were combined to create Xitsonga and isiZulu corpora with 2.6M and 2.9M words respectively.

3.1.3 Newstools Isolezwe Corpus. News articles from the isiZulu Isolezwe newspaper have been scraped and consolidated for the newstools initiative[4]. The process has been automated and the repository of news articles is updated on a continuous basis as news articles are published. At the time of writing, we estimate that this corpus contains under 1.2M words of acceptable quality.

3.1.4 Other Corpora. Other datasets have been collected for previous projects such as the Autshumato project[5] (2010) and the African Speech Technology project (2004). However, compared to the NCHLT Text project, these datasets contain relatively small corpora and in the case of the Autshumato project, corpora are only available for a subset of the South African languages.

3.1.5 Proposed Dataset Use . The Sotho-Tswana and Nguni language families are two of the largest language families in South Africa. We expect the model's relative performance to be similar within each of the language families due to the similarities between the languages. Specifically, given datasets of equal size and quality, we expect that the models would achieve similar perplexity test scores, relative to the other models, on languages within the same family. Thus, as a starting point, a representative language from each of these two families will be selected and used in the initial training procedures. In later stages of the project, provided there is sufficient time, the models will be evaluated on the remaining languages.

Sepedi will be used to represent the Sotho-Tswana languages due to having the largest available corpus within this family. We hypothesise that available corpora from the NCHLT and Autshumato projects can be combined to yield a corpus of with an estimated size of 3.6M words.

Similarly, **isiZulu** will be used to represent the Nguni languages due to the availability of corpora. In this case we estimate that the NCHLT, Autshumato and Newstools corpora can be combined to yield a 3.18M word corpus. It should be noted that for isiXhosa the NCHLT and Autshumato datasets can be combined to yield a larger 4.21M word corpus, but we have chosen isiZulu due to the availability of the Newstools corpus which presents an opportunity to test whether our models can generalise across domains.

It should be noted that the NCHLT and Autshumato datasets were both collected from government websites so there is a potential for overlap. When creating combined corpora we will need to detect and prevent duplicate texts. The Newstools dataset, by contrast, is entirely comprised of news articles. Unlike government websites, the news articles likely contain much informal or colloquial language, especially since they contain many quotations of speakers/sources. We aim to evaluate the performance of our language models on both the government website and news article domains. Furthermore we will evaluate whether adding training data from the news domain will improve test performance in the government text domain and similarly, vice versa.

We will request access to the MuST Xitsonga and isiZulu corpora as these are among the largest and potentially highest quality available. However, Xitsonga is of the Tswa-Ronga family and thus will not be used in our initial evaluation.

3.1.6 Dataset Pre-Processing. Since most of the datasets originate from web-scraped articles there are many unwanted artifacts. For example the datasets contain some URLs, HTML tags and menu labels. The Newstools dataset contains links to videos, Twitter tags and mentions, and some English web-related sentences such as JavaScript warnings. The datasets will therefore need to be cleaned before use. The NCHLT dataset does provide cleaned versions of corpora. In this case the quality of the cleaning will need to be assessed and additional cleaning performed if necessary.

3.2 Tokenization

We plan to test three different tokenization strategies: character-level, word-level and Byte-Pair encodings [14]. We hypothesise that Byte-Pair encoding will yield the best performance due to the extensive use of agglutination in South African languages. We expect that word-level tokenization will result in large vocabulary sizes forcing increased use of Out of Vocabulary (OOV) tokens. On the other hand, character-level encoding will require models to learn very long token sequences. Byte-Pair encoding presents a middle ground where common groups of characters may grouped into single tokens resulting in shorter token sequences than than character-level tokenization and a smaller vocabulary size than word-level tokenization.

[2] Datasets are available at www.sadilar.org

[3] This dataset is not in the public domain, although we have made a request for access from http://engineering.nwu.ac.za/must

[4] Available on Github at https://github.com/newstools

[5] Available at the following URLs:
https://sourceforge.net/projects/autshumato/files/
https://repo.sadilar.org/handle/20.500.12185/524
https://repo.sadilar.org/handle/20.500.12185/418?show=full
https://repo.sadilar.org/handle/20.500.12185/413

3.3 Model Implementations

3.3.1 N-Gram Implementation. The n-gram with modified Kneser-Ney smoothing estimation and inference implementation known as KenLM [6] will be used. This implementation supports high performance estimation by map-reduce parallelisation. We will also use the Stanford Research Institute Language Modelling Toolkit (SRILM)[7] to evaluate other smoothing algorithms such as Jelinek-Mercer and Katz Smoothing.

3.3.2 LSTM Implementation. An open source implementation AWD-LSTM [8] released by the authors will be used for our LSTM language model implementation. This implementation uses the Pytorch neural network library which supports training on CUDA-enabled GPUs. There are also implementations for architectures such as Quasi-Recurrent Neural Networks (QRNNs) and Morgrifier networks, two alternative LSTM type models which have also shown good results in previous work.

3.3.3 Transformer Implementation. We will be using the open source huggingface Transformers library's implementation of GPT-2. This implementation also uses the Pytorch neural network library, enabling training on CUDA-enabled GPUs.

3.4 Model Benchmarks

In order to test the model implementations and training procedures, the models will initially be evaluated on the Penn Treebank (PTB) and Wikitext-2 (WT2) test sets. The results will be compared with known performance scores for each type of model to determine whether there are any implementation issues affecting performance.

3.5 Model Training

Due to the small dataset size, we expect only modest hardware to be required to achieve practically acceptable training times. For the n-gram models, we expect CPU-based training on a regular workstation computer to be sufficient. For the LSTM and Transformer models, we will require GPU hardware acceleration to achieve practical training times.

3.6 Model Optimisation

For each class of model, many training runs will be required since many hyper-parameter combinations must be tested. These include parameters such as the hidden layer sizes for the neural models. On each run the models will be trained on the training set and evaluated on the validation set.

3.7 Model Evaluation

Finally, each model will be evaluated on the test set. We expect neural models to achieve lower perplexity scores than n-gram models. We do not have an expected quantity of improvement, because we are using uncommon datasets of varying sizes in different languages to those used for comparisons in existing literature. It is therefore difficult to give a quantitative estimate of the expected improvement since we have no direct reference to compare against.

3.8 Potential Further Experiments

Should the project proceed faster than expected, it may be feasible to perform additional experiments. For example, the output of each type of model could be interpolated to potentially improve performance. Furthermore, transfer learning could be attempted using techniques such as pretraining models on related languages and training cross-lingual embeddings.

4 ETHICAL, PROFESSIONAL AND LEGAL ISSUES

The project does not require external participants for evaluation of model performance at any stage.

Most of the datasets that will be used to train our language models will be sourced from the Language Resource Management Agency of SADiLaR, who, under the Creative Commons Attribution 2.5 South Africa License, allows us to create derived work from their datasets. Another source of text corpora may come from Isolezwe newspaper articles, effort has been made to reach out to the news site to obtain explicit permission for the use of their data in this research. As this is a non-commercial use, the terms of the site do not expressly prevent our use of their articles. Should Isolezwe request that their data not be used, it will be removed from the study. MuST Xitsonga and isiZulu corpora will be used only if we acquire explicit permission to do so. Finally, we have not identified any ethical or legal implications associated with using the Autshumato dataset.

All software used in this project will be used in compliance with the third-party use specifications of the software libraries. This approach, combined with our approach to data usage, leads us to anticipate no legal issues. All code written by us will be open source.

5 RELATED WORK

Despite the research field of language modelling making great strides forward in recent history, the majority of the research has been concentrated on languages with significantly large datasets [13]. Thus, there has been a lack of research and results on how these significant improvements relate to low-resource languages.

A 2014 study by Gandhe, Metze and Lane [6] investigated and compared the performance of a modified Kneser-Ney [7] n-gram language model, Feedforward Neural Network Language Model (FFNNLM), and a Recurrent Neural Network Language Model (RNNLM) on a significantly limited amount of language model training data. The low-resource languages modelled on were Tagalog, Pashto, Cantonese, Turkish, and Vietnamese. The results of this study demonstrated that, under low-resource conditions (approx. 100k training tokens), FFNNLMs perform better than RNNLMs. The study also demonstrated that the relative improvement from the modified Kneser-Ney model to Neural Network models increases with the size of the training data.

A 2018 study by Scarcella [13] similarly investigated and compared modified Kneser-Ney N-gram language models to basic RNNs on a severely limited amount of language model training data. The languages modelled on were Xitsonga, IsiZulu, Afrikaans, and English. This study demonstrated that the n-gram models performed better across all languages except Afrikaans, with the n-gram model having a particularly significant improvement over the RNN model

[6] KenLM is available at https://github.com/kpu/kenlm
[7] SRILM is available at http://www.speech.sri.com/projects/srilm/

on the IsiZulu dataset. This study also demonstrated that there is a significant improvement in performance over the individual models when the two models are linearly interpolated.

Our research may reach different conclusions than these two studies, as although we will be investigating modified Kneser-Ney N-gram language models, we will be using an LSTM model rather than a basic RNN, as LSTM models have seen significant improvements in language modelling performance over basic RNNs and n-gram models in high-resource settings [8]. Past research has also not looked at the performance of Transformer models when modelled on low-resource languages, which we plan to investigate.

6 ANTICIPATED OUTCOMES

6.1 System

Given that the objective of our work relates to an investigative and comparative process of three different language models, all three models need to be implemented accurately. This is required for a fair evaluation and comparison between the language models, which will guide our research. In order for accurate implementation and testing, the same datasets used by each language model needs to be pre-processed into tokens so that each model can use the data for training purposes. To ensure fair evaluation in a closed-vocabulary setting, the same vocabulary has to be used across all three models. In both a closed-vocabulary and an open-vocabulary setting, such as byte-pair encoding, we will have to normalize over the number of tokens in the test set.

6.2 Impact

We expect to be able to measure each models perplexity when tested on the same set of test data for all three models. According to this measure of performance, we expect to see the Transformer model perform the best out of all three language models, as it has recently clearly shown to have state of the art performance in high-resource settings, as well as similar state of the art performance when trained on small datasets [3].

6.3 Success Factors

The project will be successful if we are able to evaluate the effectiveness of the three language models investigated, when applied to low resource South African languages, and if we are able to compare each language model to one another in terms of their performance in this low-resource setting. In order for this to be possible, each language model will need to be implemented and tested correctly according to some test data. For the neural language models, thorough hyperparameter tuning to obtain optimal performance from each model is critical for a fair and successful comparison. The perplexity of each model must be calculated correctly, which will also ensure a fair comparison between the models. If we are able to evaluate and make clear comparisons between the models in question, according to their respective performance, we will be able to make conclusions about how each model performs when applied to low-resource South African languages.

7 PROJECT PLAN

7.1 Risks

The potential risks we anticipate for this project are outlined in a Risk Matrix (see Appendix A) in which we describe our risk management strategies. We expect a low overall risk for the project.

7.2 Timeline and Milestones

The project timeline runs from the 30th of March to the 19th of October 2020. A detailed breakdown of the timeline and different milestones is available in the form of a Gantt chart in Appendix B. The implementation and testing of the models follows a largely sequential sequence of implementation, testing, evaluation and comparisons with each model being worked on in parallel. Additionally, the writeup of the report runs throughout the project in order to spread out the work required.

Key milestones for the project are:

- Datasets in chosen languages acquired
- Initial algorithms implemented
- Model development completed
- Model training completed
- Testing framework completed
- Submission of paper outline
- Submission of paper draft
- Submission of final paper
- Completion of poster
- Completion of web page

7.3 Resources Required

7.3.1 Datasets: Several initial datasets have been identified in the languages to be examined, see 3.1 for further details.

7.3.2 Hardware: We plan to apply for time on the UCT HPC cluster for the final testing and evaluation of the models. Additionally team members have access to their personal computers for development and supplemental training during testing and evaluation of the different models.

7.3.3 Software: The project will make use of open source implementations of the examined models.

The transformer model will be using the huggingface open source implementation of OpenAI's GPT-2 model [8] [12] built on Pytorch with numpy[9] [10]. The LSTM model will be using the AWD-LSTM model [10] built on Pytorch and Numpy again. Finally, the N-gram model implementations use the KenLM and SRILM at libraries.

7.4 Deliverables

The key deliverable for the project is the final paper presenting the results of the research. The key deliverables (some of which have already been completed) are given below:

- Three literature reviews (completed)
- Project proposal

[8] Available at https://github.com/huggingface/transformers

[9] Numpy is widely regarded to be the fundamental library for scientific computing in Python.

[10] Available at https://github.com/salesforce/awd-lstm-lm

- Software proof of concept
- Draft paper
- Final trained model parameters
- Detailed validation results
- Final paper
- Final open-source implementation
- Project poster
- Project website

7.5 Work Allocation

Work will be split up equally amongst the team with each member focusing on a specific class of model. Namely Jared Shapiro will cover N-grams, Luc Hayward will cover LSTMs and Stuart Mesham will cover Transformers. The identification of datasets has been started by Stuart and each member will be responsible for collecting the dataset for PTB and WT2, Nguni languages, and Sotho-Tswana languages. These will then be shared amongst the group. Finally the results of each members testing will be shared to allow for cross comparison of results as needed. This allows for work to proceed in parallel without any one member disrupting the work of any other should they fall behind or drop out of the project.

ACKNOWLEDGMENTS

This work is based on the research supported in part by the National Research Foundation of South Africa as well as the University of Cape Town.

REFERENCES

[1] Catherine Chavula and Hussein Suleman. 2016. Assessing the Impact of Vocabulary Similarity on Multilingual Information Retrieval for Bantu Languages. In *Proceedings of the 8th Annual Meeting of the Forum on Information Retrieval Evaluation* (Kolkata, India) *(FIRE '16)*. Association for Computing Machinery, New York, NY, USA, 16–23. https://doi.org/10.1145/3015157.3015160

[2] Stanley F Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language* 13, 4 (1999), 359–394.

[3] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 2978–2988. https://doi.org/10.18653/v1/P19-1285

[4] Jurafsky Daniel and James H Martin. 2019. N-gram Language Models. In *Speech and Language Processing* (3 ed.). Chapter 3.

[5] Roald Eiselen and Martin Puttkammer. 2014. Developing Text Resources for Ten South African Languages. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA), Reykjavik, Iceland, 3698–3703. http://www.lrec-conf.org/proceedings/lrec2014/pdf/1151_Paper.pdf

[6] Ankur Gandhe, Florian Metze, and Ian Lane. 2014. Neural Network Language Models for low resource languages. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. Carnegie Mellon University, International Speech and Communication Association, 2615–2619.

[7] Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for M-gram language modeling. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, Vol. 1. IEEE, 181–184. https://doi.org/10.1109/icassp.1995.479394

[8] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2018. Regularizing and Optimizing LSTM Language Models. In *International Conference on Learning Representations*.

[9] B. Ndaba, H. Suleman, C. M. Keet, and L. Khumalo. 2016. The effects of a corpus on isiZulu spellcheckers based on N-grams. In *2016 IST-Africa Week Conference*. 1–10.

[10] Travis E Oliphant. 2006. *A guide to NumPy*. Vol. 1. Trelgol Publishing USA.

[11] Laurette Pretorius and Sonja Bosch. 2009. Exploiting Cross-Linguistic Similarities in Zulu and Xhosa Computational Morphology. In *Proceedings of the First Workshop on Language Technologies for African Languages*. Association for Computational Linguistics, Athens, Greece, 96–103. https://www.aclweb.org/anthology/W09-0714

[12] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. (2019). https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

[13] Alessandro Scarcella. 2018. *Recurrent neural network language models in the context of under-resourced South African languages*. Ph.D. Dissertation. University of Cape Town.

[14] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, 1715–1725.

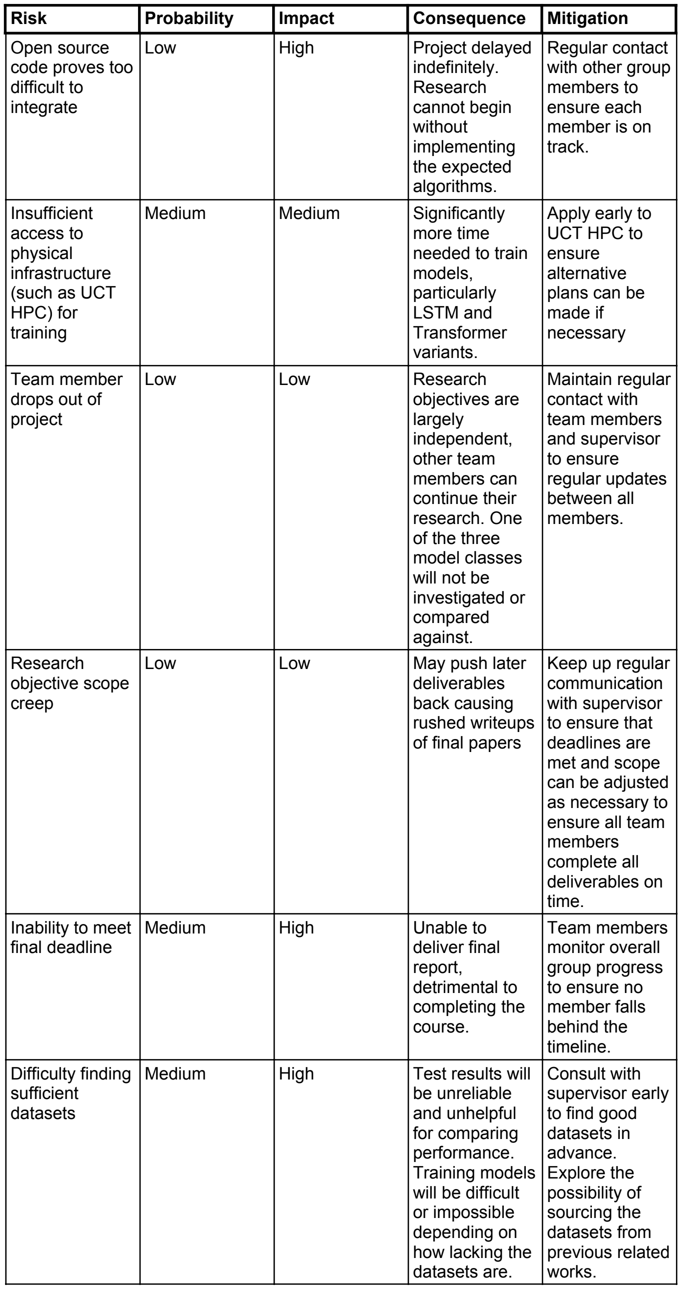
Appendices

A DATASET SIZES

Language	Family	Dataset Size (Millions of Tokens)			
		NCHLT	Newstools	Autshumato	Combined
Sepedi	Sotho-Tswana	2.77		2.39	3.61
isiZulu	Nguni	1.64	1.14	0.84	3.18
Sesotho	Sotho-Tswana	2.35		0.40	2.35
Setswana	Sotho-Tswana	1.91		0.90	3.25
isiXhosa	Nguni	1.71		2.50	4.21
Xitsonga	Tswa-Ronga	1.64		0.50	2.14
tshiVenda	Venda	1.26			1.26
isiNdebele	Nguni	1.19			1.19
Siswati	Nugni	1.15			1.15

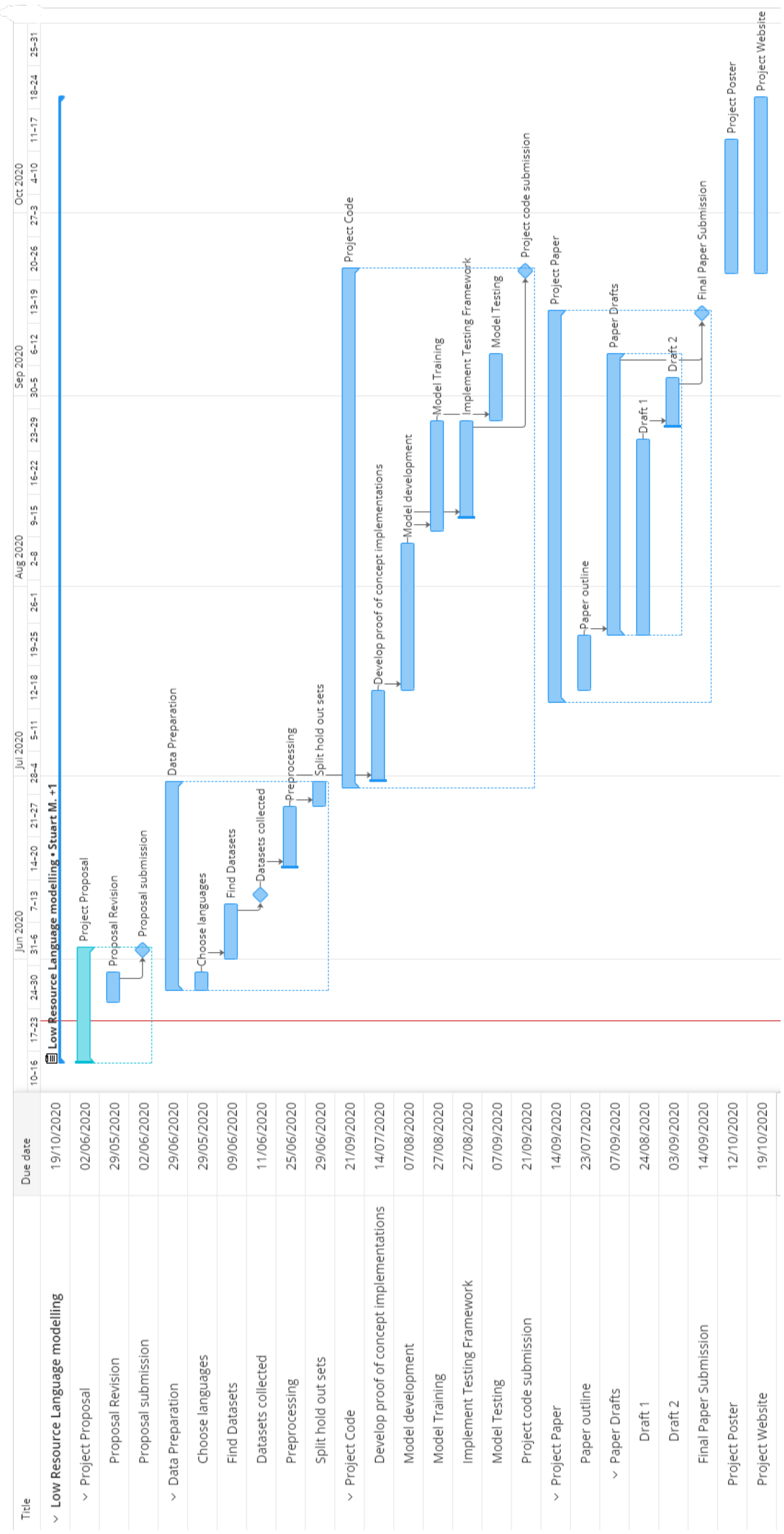
Table 1: A table showing the sizes (in millions of word tokens) of available datasets. The sizes of the NCHLT datasets are as reported by the original authors [5]. The sizes of the Newstools and Autshumato datasets are upper-bounds estimated by preliminary inspection. The combined column shows the estimated corpus size resulting from the concatenation of available corpora for each language. The combined dataset size should be interpreted as an upper bound since the effective size will be reduced by dataset cleaning and duplicate removal.

B RISK MATRIX



Risk	Probability	Impact	Consequence	Mitigation
Open source code proves too difficult to integrate	Low	High	Project delayed indefinitely. Research cannot begin without implementing the expected algorithms.	Regular contact with other group members to ensure each member is on track.
Insufficient access to physical infrastructure (such as UCT HPC) for training	Medium	Medium	Significantly more time needed to train models, particularly LSTM and Transformer variants.	Apply early to UCT HPC to ensure alternative plans can be made if necessary
Team member drops out of project	Low	Low	Research objectives are largely independent, other team members can continue their research. One of the three model classes will not be investigated or compared against.	Maintain regular contact with team members and supervisor to ensure regular updates between all members.
Research objective scope creep	Low	Low	May push later deliverables back causing rushed writeups of final papers	Keep up regular communication with supervisor to ensure that deadlines are met and scope can be adjusted as necessary to ensure all team members complete all deliverables on time.
Inability to meet final deadline	Medium	High	Unable to deliver final report, detrimental to completing the course.	Team members monitor overall group progress to ensure no member falls behind the timeline.
Difficulty finding sufficient datasets	Medium	High	Test results will be unreliable and unhelpful for comparing performance. Training models will be difficult or impossible depending on how lacking the datasets are.	Consult with supervisor early to find good datasets in advance. Explore the possibility of sourcing the datasets from previous related works.

C PROJECT TIMELINE (GANTT CHART)



Title	Due date
⌄ Low Resource Language modelling	19/10/2020
⌄ Project Proposal	02/06/2020
Proposal Revision	29/05/2020
Proposal submission	02/06/2020
⌄ Data Preparation	29/06/2020
Choose languages	29/05/2020
Find Datasets	09/06/2020
Datasets collected	11/06/2020
Preprocessing	25/06/2020
Split hold out sets	29/06/2020
⌄ Project Code	21/09/2020
Develop proof of concept implementations	14/07/2020
Model development	07/08/2020
Model Training	27/08/2020
Implement Testing Framework	27/08/2020
Model Testing	07/09/2020
Project code submission	21/09/2020
⌄ Project Paper	14/09/2020
Paper outline	23/07/2020
⌄ Paper Drafts	07/09/2020
Draft 1	24/08/2020
Draft 2	03/09/2020
Final Paper Submission	14/09/2020
Project Poster	12/10/2020
Project Website	19/10/2020