Transformer-based Machine Translation for Low-resourced Languages embedded with Language Identification

Tshephisho J. Sefara
Next Generation Enterprises and Institutions
Council for Scientific and Industrial Research
South Africa
tsefara@csir.co.za

Skhumbuzo G. Zwane
Department of Computer Science
University of Zululand
South Africa
201144122@stu.unizulu.ac.za

Nelisiwe G. Gama
School of Computer Science and Applied Mathematics
University of the Witwatersrand
South Africa
nellygrattie@gmail.com

Hlawulani Sibisi
Next Generation Enterprises and Institutions
Council for Scientific and Industrial Research
South Africa
hlawusibisi@gmail.com

Phillemon N. Senoamadi
Next Generation Enterprises and Institutions
Council for Scientific and Industrial Research
South Africa
phillemon@aims.ac.za

Vukosi Marivate
Department of Computer Science
University of Pretoria
South Africa
vukosi.marivate@cs.up.ac.za

***Abstract*—Recent research on the development of machine translation (MT) models has resulted in state-of-the-art performance for many resourced European languages. However, there has been a little focus on applying these MT services to low-resourced languages. This paper presents the development of neural machine translation (NMT) for low-resourced languages of South Africa. Two MT models, JoeyNMT and transformer NMT with self-attention are trained and evaluated using BLEU score. The transformer NMT with self-attention obtained state-of-the-art performance on isiNdebele, Siswati, Setswana, Tshivenda, isiXhosa, and Sepedi while JoeyNMT performed well on isiZulu. The MT models are embedded with language identification (LID) model that presets the language for translation models. The LID models are trained using logistic regression and multinomial naive Bayes (MNB). MNB classifier obtained an accuracy of 99% outperforming logistic regression which obtained the lowest accuracy of 97%.**

***Index Terms*—machine translation, low-resourced languages, neural network, language identification**

I. Introduction

Machine translation on African languages has received awareness in terms of online communities and organisations such as:

- Deep Learning Indaba[1]: an organisation that focuses on strengthening machine learning for African communities.
- BlackinAI: a transcontinental community that strengthen the presence of Black individuals in the field of artificial intelligence.
- Zindi[2]: a data science platform for competitions and hackathons initiated in Africa.
- Lanfrica: an online open-source database system for simplifying access to existing machine learning research and results for African languages [1].
- Masakhane[3]: an online community focused on building machine translation for African languages [2].

Machine translation (MT) has found great applications for many users in products such as Google Translate, which enable a way of translating text from one language to another. This technique is integrated into many websites. MT is one of the major parts of Artificial Intelligence and Natural Language Processing (NLP). In NLP, MT can solve problems of cross-lingual information retrieval or extrinsic evaluation of more basic tasks. Good examples of MT has been implemented.

- Google Translate[4]
- IBM Translate[5]
- Microsoft Translator[6]

[1] https://deeplearningindaba.com/

[2] https://zindi.africa/

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

[4] https://translate.google.com

[5] https://www.ibm.com/demos/live/watson-language-translator/self-service/home

[6] https://www.bing.com/translator

- Translate.com[7]

However, these MT systems do not include all low-resourced languages used in South Africa excluding Afrikaans, Sesotho, isiZulu and isiXhosa which are included in Google Translate while Microsoft Translator and Translate.com only include Afrikaans. A summary of these systems and language coverage is shown in Table I. Hence, creating MT models for low-resourced South African languages will play an important part in various sectors like teaching and learning, and communication between native and non-native speakers.

TABLE I
MT SYSTEMS AND THE LANGUAGES COVERED

Languages	Google	IBM	Microsoft	Translate.com
Afrikaans	yes	no	yes	yes
Sepedi	no	no	no	no
Sesotho	yes	no	no	no
Setswana	no	no	no	no
isiZulu	yes	no	no	no
isiXhosa	yes	no	no	no
Siswati	no	no	no	no
Tshivenda	no	no	no	no
Xitsonga	no	no	no	no
isiNdebele	no	no	no	no

as at 28/Oct/2020

The architecture of the proposed system is shown in Figure 1. This paper aims to build an English to local language MT toolkit using neural networks and embed it with a language identification (LID) interface that presets a language before translation continues.

The main contributions of this paper are as follows:

- We train and compare Neural Machine Translation (NMT) models built using JoeyNMT [3] and NMT with self-attention[8].
- We benchmark with the recent development of MT systems for low-resourced languages.

The paper is organised as follows: Section II discusses the literature review about MT systems. Section III details the methodology. Section IV discusses the experimental results and findings. The implementation of the proposed system is discussed in Section V and the paper is concluded in Section VI.

II. LITERATURE REVIEW

This section review MT models trained for low-resourced languages of Africa.

Nekoto et al. [4] used JoeyNMT to train MT models for African languages using JW300 [5] and Autshumato [6] dataset. Authors trained from English to African language in most cases and good results were achieved for some languages with Tshivenda being at the top obtaining 49.57 BLEU. On the other hand, Martinus et al. [7] used JoeyNMT to train MT models for South African languages. Authors obtained good results on Tshivenda with BLEU score of 52.27 outperforming Nekoto et al. who achieved a BLEU score of 49.57.

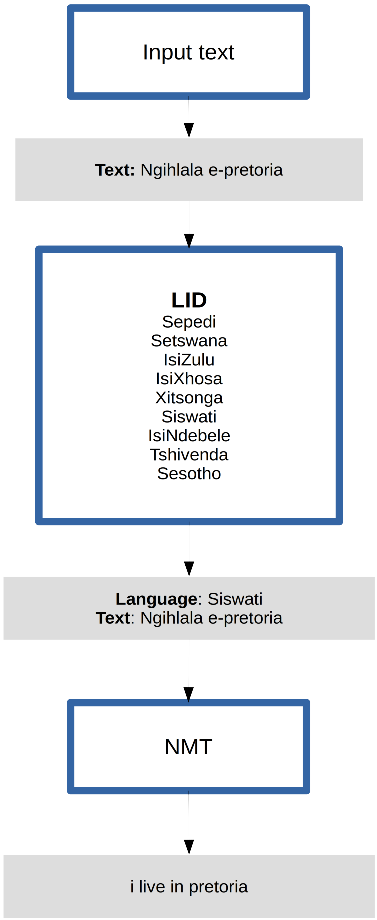


Fig. 1. Proposed architecture

Recently, Lakew et al. [8] created NMT models for five African languages, Swahili, Amharic, Tigrigna, Oromo, and Somali. Authors trained the NMT models from and to English using OpenNMT [9] based on transformer mechanism. Authors obtained poor performance results when testing the models with out-of-domain data and recommended the use of robust multilingual, transfer-learning, and semi-supervised modelling approaches.

Duh et al. [10] explored MT for low-resourced languages by creating MT for Somali-to-English and vice versa. Authors compare NMT and statistical MT under low resource conditions and ascertain that NMT can perform well but requires careful parameter fine-tuning and can benefit well when training data is increased. For improvement in translation services for African languages, authors suggest robust semi-supervised, multilingual, and transfer-learning modelling approaches.

Ahia et al. [11] created multiple NMT for Nigerian Pidgin using Transformer mechanism. Authors used JW300 to train supervised and unsupervised NMT models. Authors obtained BLEU of 24.29 for the supervised model (Byte Pair Encoding) when translating from English to Pidgin. Performance increased when authors tested NMT models from Pidgin to

[7] https://www.translate.com/
[8] https://www.tensorflow.org/tutorials/text/transformer

English where the supervised model (Word-Level) obtained the highest BLEU of 24.67.

Akinfaderin [12] created HausaMT which is an MT for Hausa: a language spoken in western Africa. The author used JW300, Tanzil, Tatoeba and Wikimedia dataset. The data was pre-processed using both Byte Pair Encoding and word-level tokenization. The highest BLEU of 45.98 was obtained for word-level tokenisation. This shows that selecting the right pre-processing technique may improve the performance of the MT models.

Sánchez-Martínez et al. [13] created an NMT between English and Swahili using news data. Authors obtained BLEU of 27.42 for English-to-Swahili translation when using the data that included part-of-speech tags outperforming Google translate. For Swahili-to-English, a BLEU of 30.55 was obtained when using part-of-speech tags outperforming Google translate. This shows that the inclusion of part-of-speech tags improves the performance of the NMT models.

III. Methodology

This section presents the methodology used in this paper. The first section outlines data gathering and collection, Section III-B presents feature extraction and data pre-processing. The transformer-based translation model is presented in Section III-C, then finally the language identification models are presented in Section III-D.

A. Data Collection

Data collection of parallel bilingual data is a challenging task when building neural translation models. Mostly for low-resourced languages, such data may be limited. Due to the lack of data, it was necessary to search and scrape data from websites that have English and SA languages dictionaries and other popular NLP data sources identified through research. South African languages do not have a rich presence on the internet. This work uses the Bible data from YouVersion website[9], Jehovah Witness Bible (JW300 site) [5] and Autshumato data [6].

1) We obtained data from the JW300 website which contains (Jehova's Witness Bibles) parallel corpus for low-resource languages including all South African languages already aligned with English.
2) More data is obtained from the Autshumato project which is supported by the South African Government. The project aims to provide more translation technologies for South African indigenous languages.
3) Another good source of data is the YouVersion Bible website where we obtained extra data. For English, we used the Good News Bible version for translations.

The following Bible translation are selected and used to create the datasets[10].

- IBhayibhili Elingcwele (Ndebele Contemporary Version 2003)
- SiSwati Bible (1996)
- IBHAYIBHELI ELINGCWELE (Zulu 1997)
- IZIBHALO EZINGCWELE (Xhosa 1996)
- BIVHILI KHETHWA Mafhungo Madifha (Tshivenda 1998)
- BIBELE Mahungu Lamanene (Xitsonga 1989)
- Setswana Bible (Setswana 1992)
- Sesotho Bible (Southern Sotho 1989)
- BIBELE (Sepedi or Sesotho sa Leboa 2000)

B. Data Cleaning and Pre-processing

The datasets are cleaned by firstly removing the null values from the datasets. This may be caused by when some verses exist in the English Bible but not found in the translated Bible. Duplicates and conflicting translations are removed and then the entire dataset is converted to a lower case. The unnecessary whitespaces are removed by striping each sentence in the dataset. Some of the translations are long so instead of using them as they are, we removed verses with length longer than 20 words and only used verses with fewer words.

For the JoeyNMT model, we pre-processed the data into Subword BPE Tokens. Usually, translations are tokenised by sentences, words and characters but using Subword BPE tokens which are known to improve results. Subword BPE is a technique that compresses data and replaces the most frequent byte pairs in a sequence with a single unused byte [14]. Table II presents the cleaned data we obtained and used to train the MT models.

For LID models, we extract verses from all the Bible translations where each translation is labelled by its natural language. We create TFIDF vectorizers using word n-grams of size 1 to 3 for each language.

TABLE II
South African languages JW300, Bible, and Autshumato data

Languages	JW300 + Bible	Autshumato + Bible
Sepedi	646057	31775
Siswati	139469	33146
isiNdebele	88110	29884
Xitsonga	876791	32905
isiZulu	1116879	32862
Sesotho	646057	33146
Tshivenda	254669	32578
isiXhosa	90795	32569
Setswana	91373	33147

C. The Transformer-based translation Model

We trained two types of ML models

- **Transformer NMT with attention:** The Transformer model is different from that of RNNs and CNNs in that it uses attention mechanism. Internally, the architecture of RNNs, CNNs and Transformers is the same but the Transformer has 6 encoders and decoders. Each encoder consisting of two layers: Multi-Head Attention and Feed Forward Neural Network. This is so that when there is an input, it flows through the self-attention layer first helping the encoder to look at other words in the input sequence

[9] www.bible.com

[10] /www.biblesociety.co.za

as it encodes a specific word. The decoder has three layers: the Multi-Head Attention, Feed Forward Neural Network and between those two layers, it has an Extra Multi-Head Attention that helps the decoder focus on the relevant parts of the input sequence. The architecture for the Transformer model is represented in Fig. 2.

- **JoeyNMT:** is a light NMT toolkit created for educational purposes based on PyTorch. It allows novices to easily and quickly learn to train custom translation models under well or low resource environment.

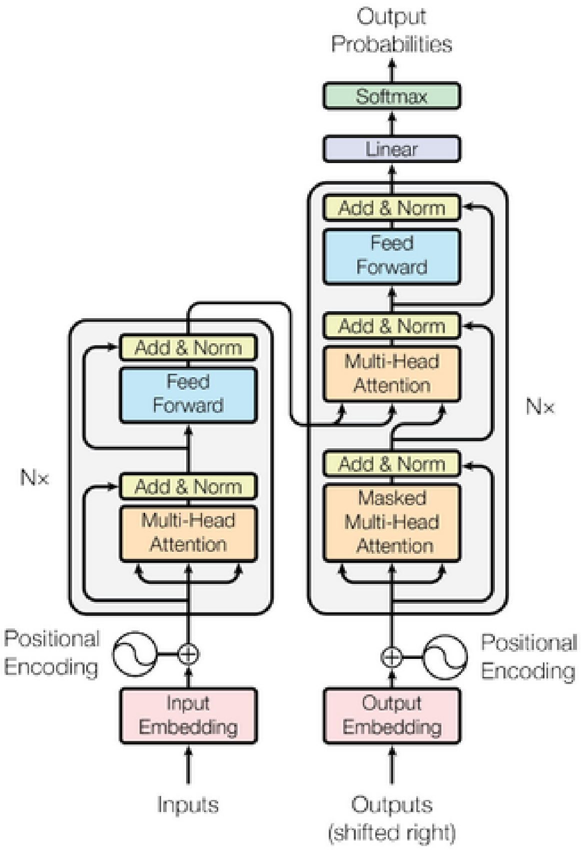


Fig. 2. Transformer Architecture

D. Language Identification

A machine translation system requires automatic LID of the input string to generate accurate results. The LID model acts as a pre-processing step to NMT. The aim of the automatic LID is to automatically preset the source language so that NMT can continue with the translation process. Two ML models are trained to act as LID:

- Logistic Regression model: is a form of logistic regression used to predict a target variable having more than two classes [15].
- Multinomial Naive Bayes model: is a specialized version of Naive Bayes that is designed more for text documents, it captures word frequency information in documents [16].

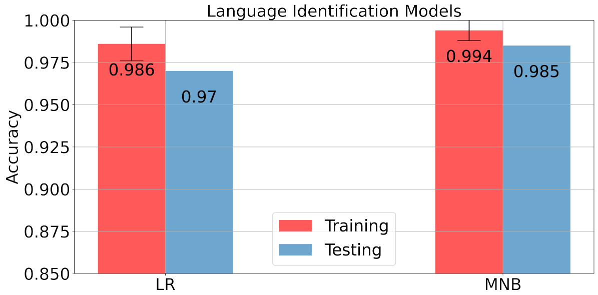


Fig. 3. Accuracy for LID.

IV. Results and Discussions

This section presents the results obtained from the different experiments conducted using the data presented in Table II. To evaluate the performance of the trained models we use the following measurement metrics:

- Accuracy metric to measure the LID model.
- The BLEU metric [17] to measure the quality of the NMT.

Firstly, Section IV-A presents the results of the LID model. Secondly, Section IV-B presents the results obtained using the transformer NMT model and Section IV-C presents the results using JoeyNMT model.

A. Language Identification

The data is divided into 90% for training and 10% for testing. We performed 5 fold cross validation experiments on identifying the language. Logistic Regression model obtained training accuracy of (0.986±0.01) while Multinomial Naive Bayes (MNB) model obtained training accuracy of (0.994±0.006) as shown in Figure 3. When testing the final model we used the 10% testing data, Logistic Regression model obtained accuracy of 0.97 and Multinomial Naive Bayes model obtained accuracy of 0.985.

B. Transformer NMT with Self-Attention

The transformer models were trained using the Bible data combined with the Autshumato data for each of the languages presented in Table III and we used 100 epochs during training. The calculation of BLEU was done using NLTK's BLEU score module[11]. We observed all the languages presented in Table III obtained state-of-the-art BLEU score of above 60. These results to date are the highest for these languages compared to [4], [7].

C. JoeyNMT

We used 30 epochs to train each language because the larger the number of epochs, the longer the JoeyNMT model takes to train. The BLEU scores for the JoeyNMT Model tested on some of the English to South African languages are shown in Table IV.

[11] https://www.nltk.org/_modules/nltk/translate/bleu_score.html

TABLE III
BLEU Scores Generated by Transformer NMT with Self-Attention

Languages	Train BLEU	Test BLEU
isiNdebele	64.10	62.51
Siswati	64.39	63.02
Setswana	62.84	61.74
Tshivenda	65.17	62.93
isiXhosa	64.97	63.51
Sepedi	63.70	62.12

TABLE IV
BLEU Scores Generated by JoeyNMT using Bible (B), JW300 (J) and Autshumato (A) Data

Languages	Train Bleu	Test Bleu	Data
Sepedi	43.96	0.43	B+J+A
Siswati	21.25	33.34	J
isiNdebele	11.07	24.22	J
Xitsonga	36.12	16.7	B+J+A
Tshivenda	35.75	0.45	B+J+A
isiZulu	36.65	47.39	J
isiZulu	43.08	51.22	J+A

TABLE V
BLEU scores by using JW300 (J) and Autshumato (A) data

Languages	Test Bleu	Data	Reference
Sepedi	19.56	A	Nekoto et al. [4]
Sepedi	45.95	J	Martinus et al. [7]
Setswana	46.91	J	Martinus et al. [7]
isiNdebele	26.61	J	Nekoto et al. [4]
Xitsonga	7.28	J	Martinus et al. [7]
Xitsonga	46.41	A	Martinus et al. [7]
Tshivenda	52.27	J	Martinus et al. [7]
isiXhosa	13.32	A	Nekoto et al. [4]
isiXhosa	37.11	J	Martinus et al. [7]
isiZulu	1.96	A	Nekoto et al. [4]
isiZulu	44.07	J	Martinus et al. [7]

Table V shows related studies of machine translation for African languages. We compare our results with Nekoto et al. [4] and Martinus et al. [7].

- **Sepedi** NMT outperformed Martinus et al. [7] with difference of 16.27 (62.22-45.95) when we combined the Bible and Autshumato data.
- **Setswana** NMT outperformed Martinus et al. [7] with difference of 14.83 (61.74-46.91) when we combined the Bible and Autshumato data.
- **IsiNdebele** NMT outperformed Nekoto et al. [4] with difference of 38.29 (62.51-26.61) when we combined the Bible and Autshumato data.
- **Xitsonga** NMT performed less to Martinus et al. [7] with difference of 29.71 (46.41-16.7) when we combined the Bible, Autshumato and JW300 data.
- **Tshivenda** NMT outperformed Martinus et al. [7] with difference of 10.66 (62.93-52.27) when we combined the Bible, Autshumato and JW300 data.
- **IsiXhosa** NMT outperformed Martinus et al. [7] with difference of 26.4 (63.51-37.11) when we combined the Bible and Autshumato data.
- **IsiZulu** NMT outperformed Martinus et al. [7] with difference of 7.15 (51.22-44.07) when we combined the JW300 and Autshumato data.

V. Implementation

Figure 4 shows the implemented system. The system runs on Python Django framework[12] as a web application. The GUI interact with the backend using application programming interfaces (APIs). The GUI has a text box for user text input. The LID prediction happens when there is a change in input text to preset the language. We selected MNB since is the best performing LID model to predict input text. The GUI shows the prediction probabilities as a bar graph where the highest language is selected for translation.

VI. Conclusion

This paper presented a machine translation system for low-resourced languages. The system uses a LID to identify the language of the input text automatically then the NMT continues with translation. The LID was trained on the Bible data for nine languages and obtained state-of-the-art accuracy of 99%. The NMT with attention obtained state-of-the-art performance results for isiNdebele, Siswati, Setswana, Tshivenda, isiXhosa, and Sepedi when using the Bible data. The future work will focus on improving translation models for these languages and building more MT models for languages not trained in this work. And curating and collecting bilingual data for these low-resourced languages.

References

[1] C. C. Emezue and B. F. Dossou, "Lanfrica: A participatory approach to documenting machine translation research on african languages," *arXiv preprint arXiv:2008.07302*, 2020.

[2] I. Orife, J. Kreutzer, B. Sibanda, D. Whitenack, K. Siminyu, L. Martinus, J. T. Ali, J. Abbott, V. Marivate, S. Kabongo *et al.*, "Masakhane–machine translation for africa," *arXiv preprint arXiv:2003.11529*, 2020.

[3] J. Kreutzer, J. Bastings, and S. Riezler, "Joey nmt: A minimalist nmt toolkit for novices," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, 2019, pp. 109–114.

[4] W. Nekoto, V. Marivate, T. Matsila, T. Fasubaa, T. Kolawole, T. Fagbohungbe, S. O. Akinola, S. H. Muhammad, S. Kabongo, S. Osei *et al.*, "Participatory research for low-resourced machine translation: A case study in African languages," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, 2020, pp. 2144–2160.

[5] Ž. Agić and I. Vulić, "Jw300: A wide-coverage parallel corpus for low-resource languages," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 3204–3210.

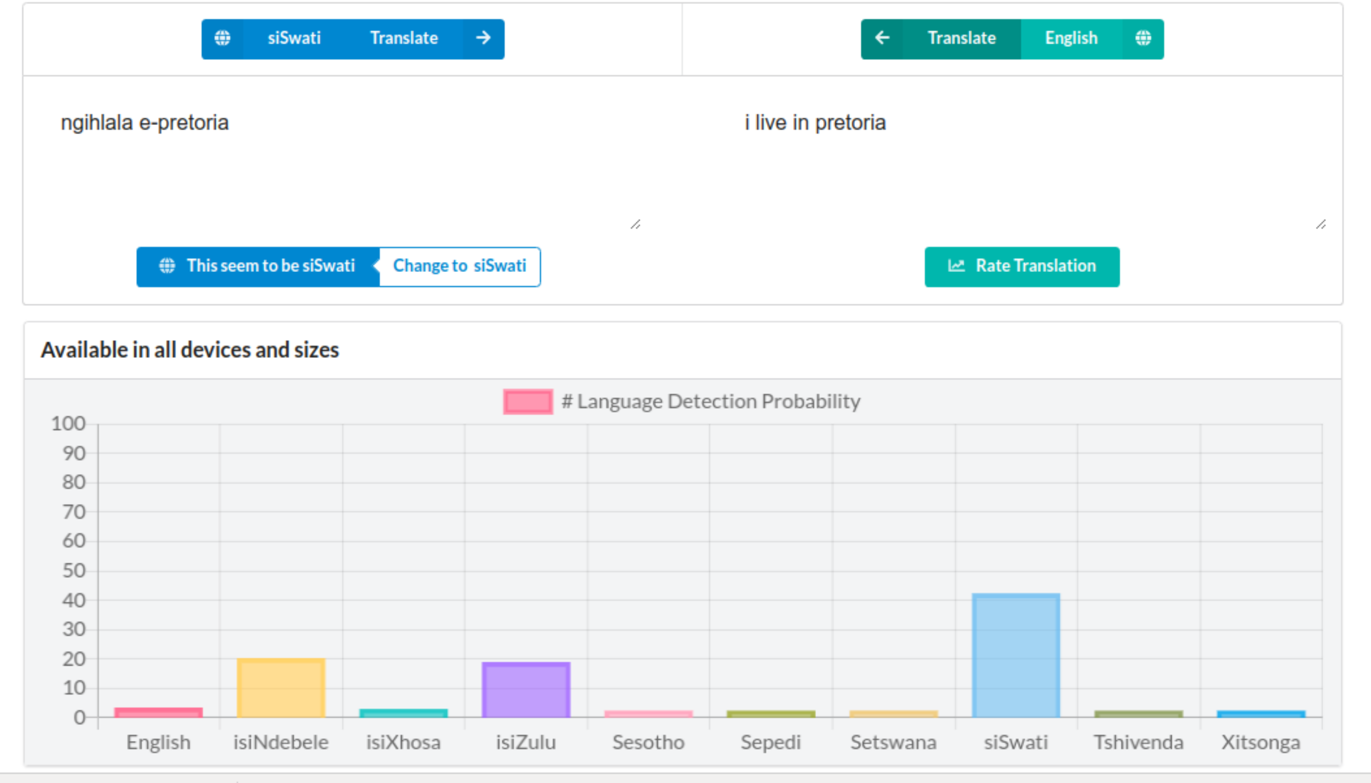
[6] H. J. Groenewald and L. du Plooy, "Processing parallel text corpora for three south african language pairs in the autshumato project," *AfLaT 2010*, p. 27, 2010.

[7] L. Martinus, J. Webster, J. Moonsamy, M. S. Jnr, R. Moosa, and R. Fairon, "Neural machine translation for south africa's official languages," *arXiv preprint arXiv:2005.06609*, 2020.

[8] S. M. Lakew, M. Negri, and M. Turchi, "Low resource neural machine translation: A benchmark for five african languages," *arXiv preprint arXiv:2003.14402*, 2020.

[9] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, "Opennmt: Open-source toolkit for neural machine translation," in *Proceedings of ACL 2017, System Demonstrations*, 2017, pp. 67–72.

[12] https://www.djangoproject.com/



ngihlala e-pretoria

i live in pretoria

Fig. 4. The interface of the proposed system

[10] K. Duh, P. McNamee, M. Post, and B. Thompson, “Benchmarking neural and statistical machine translation on low-resource african languages,” in *Proceedings of The 12th Language Resources and Evaluation Conference*, 2020, pp. 2667–2675.

[11] O. Ahia and K. Ogueji, “Towards supervised and unsupervised neural machine translation baselines for Nigerian Pidgin,” *arXiv preprint arXiv:2003.12660*, 2020.

[12] A. Akinfaderin, “Hausamt v1. 0: Towards english-hausa neural machine translation,” *arXiv preprint arXiv:2006.05014*, 2020.

[13] F. Sánchez-Martínez, V. M. Sánchez-Cartagena, J. A. Pérez-Ortiz, M. L. Forcada, M. Espla-Gomis, A. Secker, S. Coleman, and J. Wall, “An english-swahili parallel corpus and its use for neural machine translation in the news domain,” in *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, 2020, pp. 299–308.

[14] R. Sennrich, B. Haddow, and A. Birch, “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)*, 2016, pp. 1715–1725.

[15] D. Böhning, “Multinomial logistic regression algorithm,” *Annals of the Institute of Statistical Mathematics*, vol. 44, no. 1, pp. 197–200, 1992.

[16] A. M. Kibriya, E. Frank, B. Pfahringer, and G. Holmes, “Multinomial naive bayes for text categorization revisited,” in *Australian Conference on Artificial Intelligence*, 2004.

[17] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in *ACL*, 2001.