

From N-grams to Pre-trained Multilingual Models For Language Identification

Thapelo Sindane¹ and Vukosi Marivate^{1,2}

¹Data Science for Social Impact,
University of Pretoria, South Africa

²Lelapa AI
sindane.thapelo@tuks.co.za,
vukosi.marivate@cs.up.ac.za

Abstract

In this paper, we investigate the use of N-gram models and Large Pre-trained Multilingual models for Language Identification (LID) across 11 South African languages. For N-gram models, this study shows that effective data size selection remains crucial for establishing effective frequency distributions of the target languages, that efficiently model each language, thus, improving language ranking. For pre-trained multilingual models, we conduct extensive experiments covering a diverse set of massively pre-trained multilingual (PLM) models – mBERT, RemBERT, XLM-r, and Afri-centric multilingual models – AfriBERTa, Afro-XLMr, AfroLM, and Serengeti. We further compare these models with available large-scale Language Identification tools: Compact Language Detector v3 (CLD V3), AfroLID, GlotLID, and OpenLID to highlight the importance of focused-based LID. From these, we show that Serengeti is a superior model across models: N-grams to Transformers on average. Moreover, we propose a lightweight BERT-based LID model (za_BERT_lid) trained with NHCLT + Vukzenzele corpus, which performs on par with our best-performing Afri-centric models.

1 Introduction

Automatic language identification (LID) is the task of determining the underlying natural language used in a written or spoken corpus (McNamee, 2005). This is a challenging problem, especially for languages with insufficient training examples and closely related languages, particularly low-resourced languages (Haas and Derczynski, 2021). For South African languages, building quality LID technologies is significantly important for sourcing internet data, which has served as a de-facto repository for many low-resourced languages, especially from public domains such as news websites (Marivate and Sefara, 2020; Adelani et al., 2021; Dione

et al., 2023; Adelani et al., 2023; Lastrucci et al., 2023).

Statistical approaches for automatic LID such as N-grams (Dube and Suleman, 2019), and more classical machine learning models such as Logistic Regression, Naive Bayes, Random Forest, Boosting machines, Support Vector Machines, and Clustering techniques (e.g K Nearest Neighbors) have been proposed (Haas and Derczynski, 2021). Moreover, contemporary neural-based architectures such as deep neural networks and convolutional neural networks have also been tested. In all cases, not enough work for the South African languages is reported.

On the other hand, recent algorithmic advancements such as transformer architectures have made a significant impact on the Natural Language Processing landscape (Devlin et al., 2018; Conneau et al., 2019). With this sudden shift in perspective, many works have proposed automatic LID using large pre-trained multilingual models, derived from attention mechanisms (Vaswani et al., 2017). Large pre-trained multilingual models are transformer-based architectures simultaneously trained on multiple languages (hence multi-lingual) using various techniques such as token (s) masking training technique, where tokens from a given sentence example are hidden and the objective of the training transformer is to predict the hidden word (s).

In this work, we make use of the recently released Vuk'zenzele crawled corpus (Lastrucci et al., 2023) and the NCHLT dataset (Eiselen and Puttkammer, 2014) to develop and experiment on automatic language identification models on 10 low-resourced South African languages: Northern Sesotho (nso), Setswana (tsn), Sesotho (sot), isiZulu (zul), isiXhosa (xho), isiSwati (ssw), isiNdebele (nbl), Tshivenda (ven), Xitsonga (tso), and Afrikaans (af). Additionally, we included the high-resource South African English (eng) to ensure representation of all 11 official languages in South

Africa. We conduct extensive experiments on N-gram models, large pre-trained multilingual models – XLM-r, mBERT, and Afri-centric multilingual models – AfriBERTa, Afro-XLMr, AfroLM, and Serengeti. We shed light on the limitations and robustness of N-gram-based approaches and the significant improvement boost of pre-trained multilingual models, especially, for those pre-exposed to low-resourced South African languages during pre-training.

2 Related Work

Large pre-trained multilingual models have shown astonishing state-of-the-art results on various Natural Language Processing (NLP) tasks such as Machine Translation, Question Answering, and Sentiment Analyses (Stickland et al., 2021; Yang et al., 2019; Adebara et al., 2023b). A precursor of these tasks is the crawling of large volumes of internet data and categorizing the data into different languages (i.e. language identification) for pre-training. For language identification, many works have used pre-trained multilingual models to expand monolingual datasets using the internet.

Jauhiainen et al. (2021) conducted a comparative study between adaptive Naive Bayes, HeLI2.0, multilingual BERT, and XLM-r models for Dravidian language identification in a code-switched context (i.e. a conventional *modus operandi* for communication on the internet). Caswell et al. (2020) developed a transformer-based LID model aside from basic filtering techniques such as tunable-precision-based filtering using a created wordlist, TF-IDF filtering, and a percent-threshold filtering threshold proposed in their study to filter noisy web-crawled content. Although they were able to collect corpora for over 212 languages, their set-up for their best-performing transformer model was unclear. Similar to our work, Kumar et al. (2023) conducted a comparative study on DistilBERT, ALBERT, and XLM-r and showed that a lightweight version of DistilBERT delivers comparable results to resource-intense models. Adebara et al. (2022), on the other hand, implemented a massive transformer-based LID model with 12 attention layers and heads. They then trained this model on 512 languages with close to 2 million sentences across 14 language families (South African languages included). Their model achieved over 95 % F1 score on a left-out test sets, outperforming available LID tools: CLD version 2, Langid, Fast-

Corpora	No. Sent	Voc	Unq. Voc	Train	Dev	Test
Vuk	33K	690K	132K	3395	-	728
NCHLT + Vuk	74K	16M	258K	6790	-	1454

Table 1: Corpora statistics for Vuk and NCHLT

text, etc. Kargaran et al. (2023) created a language identifier that covers a whopping 1600 low-high-resourced African languages. Due to the unavailability of resources utilized in previous studies, our research concentrated exclusively on 11 South African languages, with only 3 language families - Sotho-Tswana, Nguni, and Creole. Furthermore, we will only consider a comparison of diverse pre-trained multilingual models (E.g mBERT, XLMr, AfriBERTa, Afro-XLMr, Serengeti, e.t.c) and two lightweight BERT-based models – DistilBERT, and `za_BERT_lid` model.

3 Methodology

The methodology employed in this study uses language-identifiable monolingual corpora from reliable sources as training examples for language identification and compares various pre-trained multilingual models for the task of discriminating between languages.

3.1 Corpora

Text corpora for the 11 South African languages were acquired from two sources: Vuk’zenzele (Vuk) (Lastrucci et al., 2023) and National Centre for Human Language Technology (NCHLT) corpora (Eiselen and Puttkammer, 2014). Table 1, describes the number of sentences (No. Sent), vocabulary (Voc) sizes, unique vocabulary sizes (Unq. Voc), and the train size per language, development set size, and test size per language splits for corpora Vuk and NCHLT. We ensure consistent train and test examples across all languages, by ensuring that all train, and test examples for each language are equal. Therefore, we only had varying development sizes. Additionally, we only considered sentences in the range of 3-50 tokens and did not use the rest of the corpus. Figure 1, and 2, describe the sentence length distribution for Vuk, and NCHLT corpora respectively.

3.2 Pre-processing

The dataset is observed to contain links, digits and therefore our pre-processing included the removal of URLs, digits, punctuations, and followed by lower-casing all sentences using Python regular

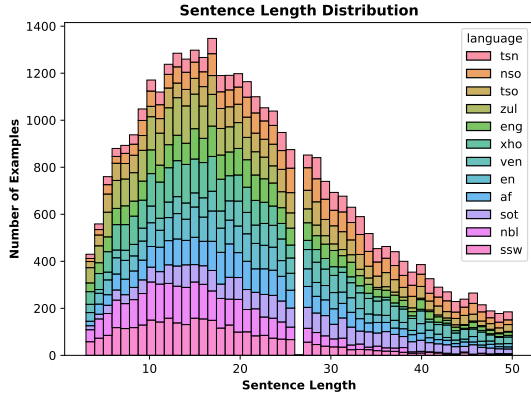


Figure 1: Sentence length distribution of Vuk corpora. The x-axis denotes the number of tokens (words) in the sentences.

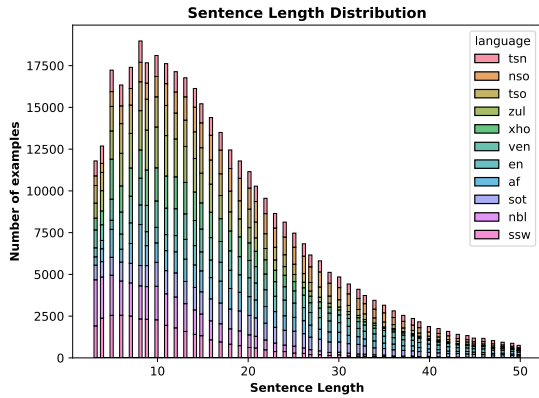


Figure 2: Sentence length distribution of NCHLT + Vuk corpora. The x-axis denotes the number of tokens (words) in the sentences.

expressions. Special characters such as š, found in Northern Sotho were left intact.

3.3 Language detection algorithms

3.3.1 N-grams

An N-gram is a sequence of consecutive characters from text (Dube and Suleman, 2019). This study explored character Bi-grams (2 consecutive characters), Tri-grams (3 consecutive characters), and Quad-grams (4 consecutive characters) models. We build each model for each language from the training dataset (Vuk, NCHLT, and Vuk + NCHLT). Furthermore, we experimented with various data sizes to investigate the impact of the number of training examples on N-gram models and this showed a performance ceiling, where an increase in training examples does not significantly

impact the quality of the models (shown in Figure 3). Each model is made up of a list of tuples of characters-frequency pair ordered in descending order.

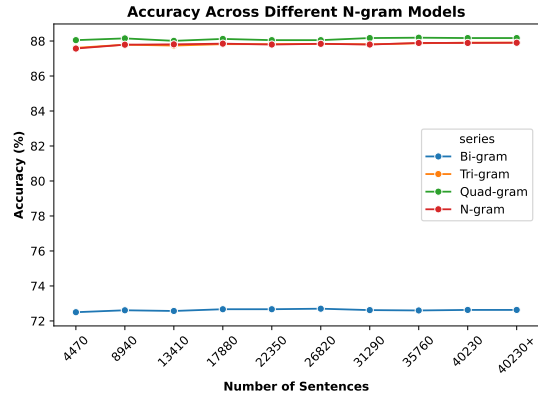


Figure 3: Data size variation performance on Vuk test data.

To discriminate between languages, the models use a ranking function. The ranking function calculates the distance of the frequency distributions of the input examples from the existing N-gram model's frequency distributions (with $k=50$ as the number of ordered N-grams to consider from the trained N-grams). The frequency distribution is calculated as the number of occurrences of each observed N-gram divided by the total number of N-grams from the corpus and taking the log of that ratio. For a given input example (in Northern Sotho) "Ke ya go thopa sefoka" translation - "I am going to win the trophy", the model first extracts the character N-grams (e.g. 2 characters if the observed model is Bi-gram) – Bi-gram Output: ['ke', 'ya', 'a_', 'go', 'th', 'ho', 'op', 'pa', 'se', 'ef', 'fo', 'ok', 'e_', '_y', '_g', 'o_', '_t', '_s'] sorted in reverse, and then the frequency distribution from the existing trained models (looking only at 50 top N-grams per language) for all the languages are compared with the new frequency distribution of the input sentence and the one with the closest similarity is considered the language of the input example. Figure 4, 5, and 6 presents heatmaps depicting the probability scores generated by the ranking function exclusively for all test examples, correctly predicted sentences, and incorrectly predicted examples, during the test phase respectively. The heatmaps reveal that the concentration of scores ranges between 0.04 and 0.06, which could be further used to drive a model's outcome improving the confidence in predictions. This

observation suggests that ranking functions play a crucial role in N-gram-based models, warranting further investigation.

N-grams experimental setup We experimented with Bi (2), Tri (3), and Quad (4) consecutive character sequences to build our models. Additionally, we combined all 3 and called it N-grams combined.

3.3.2 Naive Bayes Classifier

Naive Bayes have been the default standard for various LID tasks such as code-switching detection, dialect discrimination, word-level language detection, and e.t.c. (Dube and Suleman, 2019; Jauhainen et al., 2019). In this study, we experimented with the multinomial Naive Bayes Classifier (NBC) implementation from Python’s scikit-learn. With NBC, we were able to extract discriminating features per language, supporting model prediction (Figure 8), and significantly improved on N-gram models (see confusion matrix in Figure 9). This highlighted important feature correlation, especially for related languages, which explains why it is challenging to discriminate among closely related languages. Moreover, this highlights the importance of lexicon-driven approaches for language filtering mentioned in Caswell et al. (2020) as alternative measures to mitigate these ambiguities.

Naive Bayes Classifier experimental setup We experimented with a TF-IDF vectorizer to generate input features. For this, we used the character bi-gram, tri-gram, quad-gram, and the 3 types combined as consecutive subwords to generate TF-IDF features. We also generated word level input features using CountVectorizer. We used a multinomial version of the Naive Bayes classifier with mostly default parameters from scikit-learn (except the alpha parameter where we tested $\alpha = 0.0001, 1.0$, where $\alpha = 1.0$ performed better). Finally, we trained Support Vector Machine (SVM), K Nearest Neighbor (KNN), and Logistic Regression with the same input features and their scikit-learn default parameters to compare performance outcomes with NBC.

3.3.3 Pre-trained Multilingual Models

This study explored a diverse set of massively pre-trained multilingual models: mBERT, XLM-r, RemBERT, and their Afri-centric counterparts: AfriBERTa, Afro-XLMr, AfroLM, and Serengeti due to their enhanced text processing capabilities and their ability to handle low-resourced languages

with complex linguistic nuances (Devlin et al., 2018; Conneau et al., 2019; Ogueji et al., 2021; Alabi et al., 2022; Dossou et al., 2022; Adebara et al., 2023a).

Large pre-trained multilingual models experimental setup Following setups in (Adelani et al., 2023; Dione et al., 2023), we used a batch size of 16, a learning rate of $2e^{-5}$, 20 epochs, save step of 10000, and sequences cut-off of 200 for all models. We ran our experiments five times with different seeds { 1,.., 5 } and reported the average results.

4 Results

4.1 Baselines

Table 2, shows results for baseline models Bi-gram, Tri-gram, Quad-gram, N-gram combined (N-gram Comb) – which uses bi-, tri-, and quad- -grams combined, and Naive Bayes Classifier (NBC) with the same character N-grams. Naive Bayes with word-level features outperform the rest of the baseline models. Interestingly, for NBC, increasing the character spans improves the performance of the classifier. Figure 10, 11, 12, 13, 14, and 15 depicts the impact of increasing the data size on models NBC, Support Vector Machine (SVM), and Logistic Regression (Log Reg) on various training features – uni-grams, bi-grams, tri-grams, quad-grams, N-grams combined, and word-level features derived using TF-IDF respectively. NBC, SVM, and Log Reg show improved performance with the change in input features while the training size shows gradual improvement in accuracy. KNN was also tried, however, the model showed abysmal performance across all features except for Bi-gram input features and was therefore omitted from the plots.

In the N-gram class, the Quad-gram ranking outperforms the rest of the N-gram-based models. Figure 17, depicts the impact of sentence length on N-gram models performance. This shows that the group of N-gram models struggles to classify shorter sentences, while NBC performs slightly better with them (Figure 18). This may be due to shorter sentences not carrying enough signal information for N-grams to discriminate across all languages as mentioned in Haas and Derczynski (2021). Additionally, N-gram-based models depict inconsistent performance across languages, where improved performance is achieved for select languages and for a specific N-gram type (E.g Bigram – eng, ven, af, e.t.c, Tri-gram – eng, tso, nso, e.t.c),

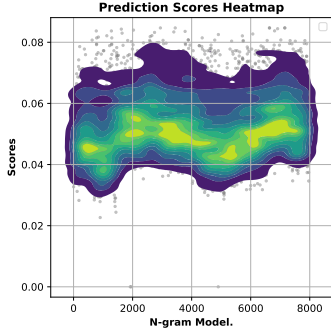


Figure 4: Score heatmap for all predictions using N-gram

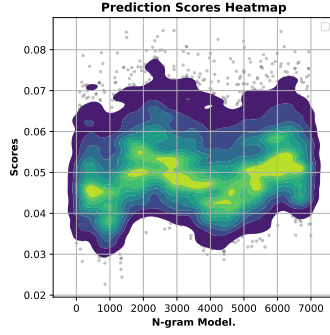


Figure 5: Score heatmap for correctly predicted examples using N-gram

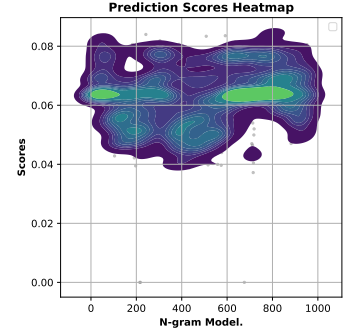


Figure 6: Score heatmap for incorrectly predicted examples using N-gram

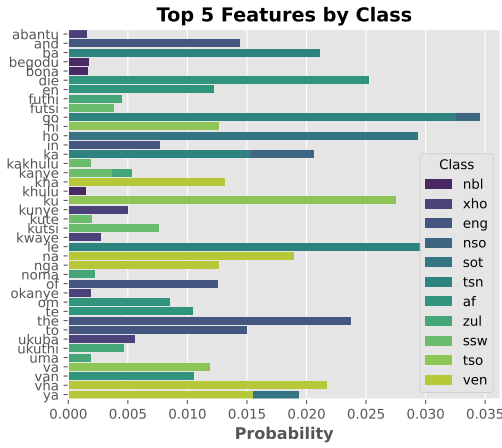


Figure 8: Top 5 important features per class from Naive Bayes

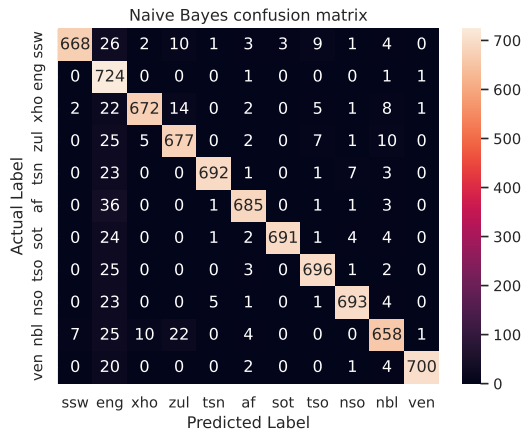


Figure 9: Accuracy of Naive Bayes Classifier.

while other languages underperform (e.g zul, isiNdebele (nbl)) (see Figure 19). Furthermore, the complexity of LID is exacerbated by closely related

Baseline	Acc	Prec	Rec	F1
Vuk				
Bi-gram	72.7	73.5	72.6	72.3
Tri-gram	87.9	88.4	87.9	88.1
Quad-gram	88.4	88.9	88.4	88.5
N-gram (Comb)	87.8	88.3	87.8	88.0
NBC (word-level)	94.5	95.2	94.5	94.6
NBC (2)	90.2	90.7	90.2	90.4
NBC (3)	93.4	93.8	93.4	93.5
NBC (4)	94.4	94.8	94.4	94.5
NBC (Comb)	94.0	94.5	94.0	94.1
K NN (2)	85.0	85.0	85.0	85.0
Log Reg (4)	94.0	95.0	94.0	94.0
SVM (4 & 2-4)	94.0	95.0	94.0	94.0

Table 2: Baseline performance evaluation using Accuracy (Acc), F1 score (F1), Precision (Prec), and Recall (Rec). K Nearest Neighbor (K NN), Logistic Regression (LR), and Support Vector Machine (SVM) are reported with best feature inputs bi-gram (2), quad-grams (4), and combinations (2-4) respectively.

languages (see confusion matrix in Figs. 20–23). While varying dataset size, and character N-gram choices slightly improve performance on distinguishing among closely related languages (Figure 3), it does not add any significant improvement on a per-language basis (see Figure 19), where languages such as isiZulu (zul) are showing no further improvement. For this, we explore large pre-trained multilingual models for automatic LID in the next subsection.

4.2 Pre-trained Multilingual Models

Table 3 reports the accuracy (Acc), precision (Prec), recall (Rec), and F1 score (F1) of pre-trained multilingual models: mBERT, XLM-r, RemBERT;

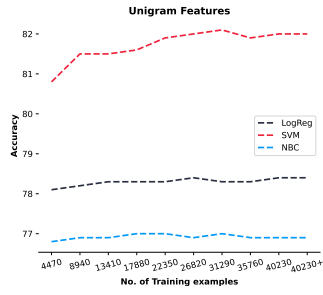


Figure 10: Unigram

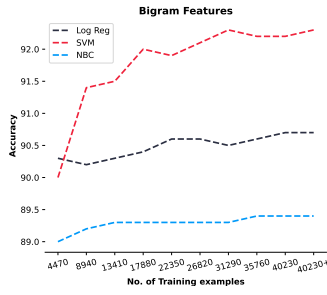


Figure 11: Bi-gram

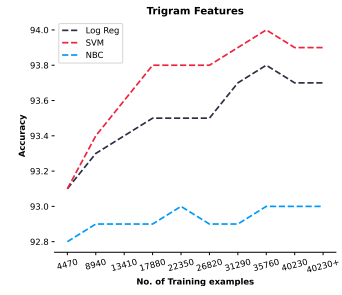


Figure 12: Tri-gram

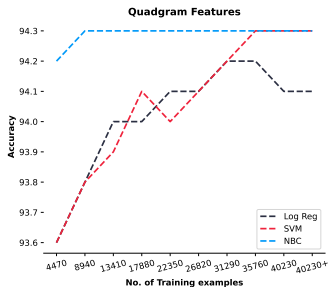


Figure 13: Quad-gram

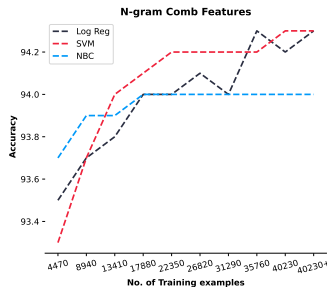


Figure 14: N-grams Comb

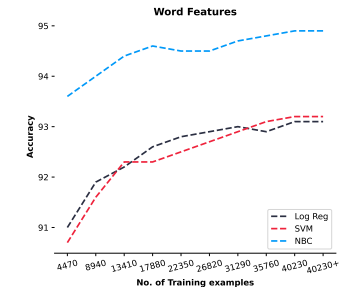


Figure 15: Word-level

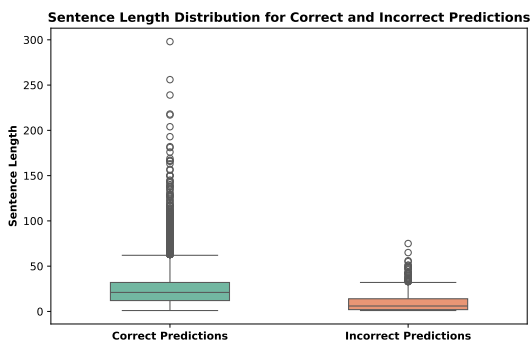


Figure 17: Box diagram depicting sentence length of correctly predicted and incorrectly predicted sentences.

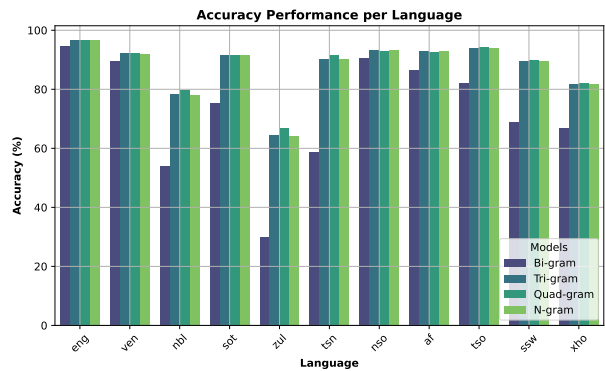


Figure 19: Accuracy score per language using N-grams

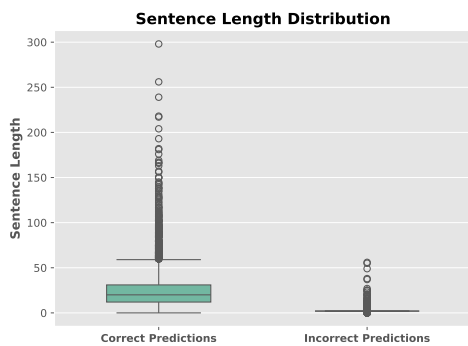


Figure 18: Incorrectly and correctly NBC classified sentence lengths

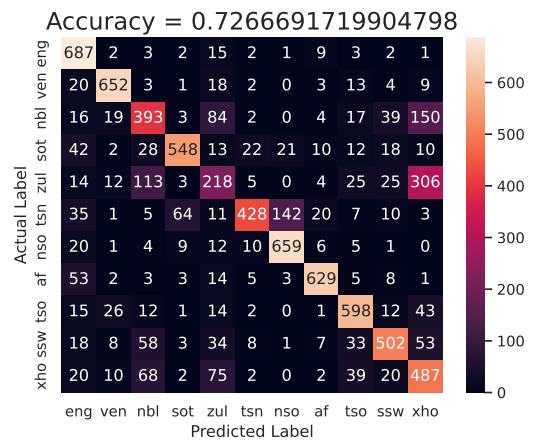


Figure 20: Bi-gram Confusion matrix on Vuk test data

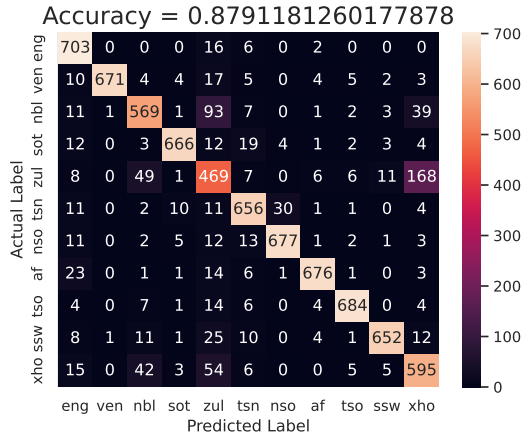


Figure 21: Tri-gram Confusion matrix on Vuk test data

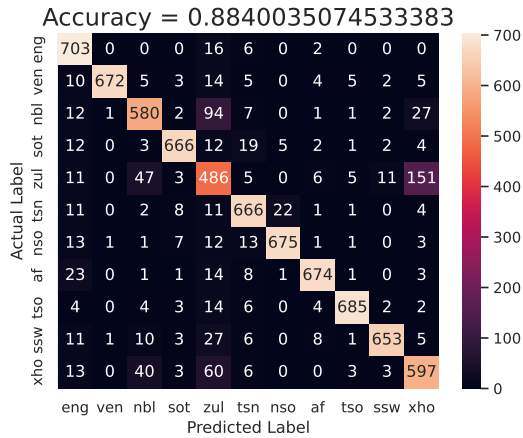


Figure 22: Quad-gram Confusion matrix on Vuk test data

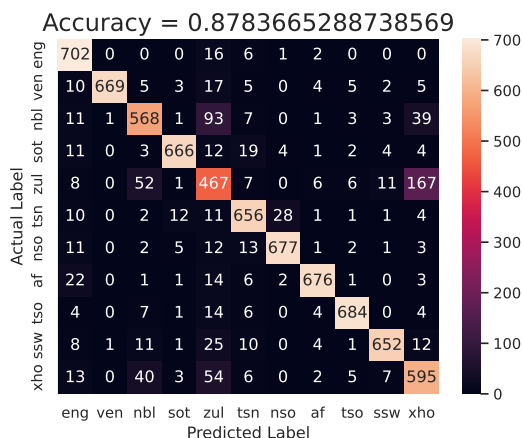


Figure 23: Accuracy of N-gram type (Bi-gram, Tri-gram, Quad-gram) combined

Afri-centric pre-trained models: AfriBERTa, AfroXLMr, AfroLM, and Serengeti; publicly available

LID tools covering South African languages: Compact Language Detector (CLD) version 3 (V3), AfroLID (Adebara et al., 2022), GlotLID (Kargaran et al., 2023), and OpenLID (Burchell et al., 2023); and our proposed lightweight BERT-based architectures: za-BERT-lid, and DistilBERT.

Pre-trained-multilingual models show impressive results for this task, with over 90% average accuracy. Serengeti outperforms the rest of the models with an average accuracy of 98 %, while mBERT is the least-performing model with an average accuracy of 96 % (≈ 2 points difference). Most importantly, the group of Afri-centric models outperforms the largely pre-trained multilingual models with the best model (XLMr-large) in this category performing slightly worse than the lowest performing model (AfroLM) in the Afri-centric group. Moreover, our proposed za-BERT-lid, and DistilBERT perform on par with the best-performing model (≈ 2 points difference) despite them being much smaller in size.

On the other hand, available LID tools show impressive and incremental results. For these models, GlotLID outperforms the rest of the sampled models in this study. This may be due to GlotLID being trained on Vuk data, giving the model an unfair advantage over others. Despite this, analyses of the predictions show that the compared models are not completely wrong, as they often struggle with closely related languages such as Sotho-Tswana language family {nso, sot, tsn}, and Nguni languages {xho, zul, ssw, and nbl}. Perhaps to remedy this, the training of LID models should prioritize precision as a metric of evaluation. Noticeably, but not alarming, the LID tools also predict unrelated languages from their training list, which perhaps highlights the need for a more focused approach rather than including many languages at once. However, we feel this claim needs further justification and we will consider this in future work.

4.3 Cross-domain evaluation

We also wanted to test our model on cross-domain datasets to inspect their generalization capabilities. We simulated this by training with Vuk data and tested it on NCHLT, and vice versa. Table 4 reports the performance of pre-trained models for examining the cross-domain evaluation theory. This table shows that the performance of the multilingual models trained with Vuk and tested with NCHLT dropped by approximately (4%-5%) across all mod-

Model	Acc	Prec	Rec	F1
PLM				
mBERT	96.7	96.7	96.6	96.7
XLMr-base	97.1	97.1	97.1	97.1
XLMr-large	97.3	97.3	97.3	97.3
RemBERT	97.1	97.1	97.1	97.1
Afri-centric				
AfriBERTa	97.6	97.6	97.6	97.6
Afro-XLMr-base	97.7	97.8	97.7	97.7
Afro-XLMr-large	98.0	98.0	98.0	98.0
AfroLM	97.4	97.5	97.4	97.4
Serengeti	98.3	98.3	98.3	98.3
LID Tools				
CLD V3	40.2	33.6	40.2	35.7
AfroLID	66.1	72.1	66.1	64.2
OpenLID	80.8	71.7	80.8	75.0
GlottLID	97.5	98.3	97.5	97.9
Lightweight				
za-BERT-lid	96.8	96.8	96.8	96.8
DistilBERT	96.2	96.2	96.2	96.2

Table 3: Performance evaluation scores of pre-trained multilingual models, available LID tools, and lightweight BERT-based models averaged over 5 runs per metric.

els. In contrast, training with NCHLT and testing with Vuk showed performance improvements. This could be due to NCHLT having more training examples, and a large vocabulary (see Table 1) allowing the model to learn more nuanced representations. Notably, larger models show better performance over smaller models for this task.

5 Discussions

Ensuring the development of robust LID detection systems remains a critical research area with implications on many NLP tasks. Importantly, the availability of reliable LID systems ensures accurate reporting on the state of low-resourced languages (Kreutzer et al., 2022).

On the side of model performance, baseline techniques such as Naive Bayes, Support vector Machines, and Logistic Regression seem to be performing quite well on the task of sentence-level language identification. We recommend these models for further research for high-level LID, compared to large pre-trained multi-lingual models which require specialized computing resources such as GPUs, to accelerate training. However, we deem such trade-offs to require more research, especially

Model	Vuk Test	NCHLT Test
Vuk Trained		
mBERT	-	91.0
XLMr-base	-	91.4
XLMr-large	-	92.2
RemBERT	-	92.3
AfriBERTa	-	92.1
Afro-XLMr-base	-	93.6
Afro-XLMr-large	-	94.1
AfroLM	-	91.8
Serengeti	-	94.9
za-BERT-lid	-	91.3
DistilBERT	-	90.9
NCHLT Trained		
XLMr-base	95.6	93.2
Afro-XLMr-base	96.3	93.6
Serengeti	97.7	94.8

Table 4: Cross-domain evaluation of models trained with Vuk and tested with NCHLT and vice-versa. Reported in F1 score averaged over five runs

in complex LID subtasks such as code-switching, or similar language discrimination.

We also, highlight the importance of evaluation metric selection as we have observed that most of the LID tools explored in this study are not completely wrong, but rather have challenges discriminating among closely related languages. Therefore, we recommend precision as an evaluation metric for LID to be further investigated.

6 Conclusion

Language Identification remains a critical study area for the widespread inclusion of many low-resourced languages into the booming technology space. In this study, we experimented with statistical approaches, traditional machine learning techniques, the recent advanced pre-trained multilingual models, as well as LID tools publicly available (covering a wide range of African languages) on the task of LID for 11 South African language discrimination. We were able to shed light on the approaches showing promising results in the South African language context and made suggestions for future directions. Concretely, we showed that the Naive Bayes algorithm performs surprisingly well for LID and warrants further exploration and exploitation, especially given its cheap-compute advantage. Finally, we compared publicly available pre-trained models and showed that context-

exposed models have an edge over other context-oblivious multilingual models, where context refers to the language. We released our models on [HuggingFace](#) and code with datasets on [GitHub](#).

7 Limitations

In this study, we did not explore any use of word embeddings for language identification. Word embeddings played in crucial role in the development of language technologies, and it would have been interesting to experiment with them. However, such resources are not readily available for many low-resourced languages.

Aside from experimenting and getting results for other traditional models such as Logistic regression, K Nearest Neighbor, and Support Vector Machines, it would have been interesting to develop and experiment with deep neural networks such as multi-layered perceptions, and convolutional neural networks. As universal approximators, these models tend to produce desirable results, with the caveat of requiring time for hyper-parameter tuning.

This study did not extensively explore the impact k (used 50 for this study), which is the count of the N-grams list used to calculate the ranking. However, we aim to explore this extensively in future works.

It is known that LID techniques tend to overfit to domain data, and therefore it would have been interesting to create free-text data created by humans and test the generalization capabilities of the developed models on human-generated text.

Recent studies have focused on resource-conscience alternatives for either compute efficiency, parameter reduction, etc. It would have been interesting if this work would have explored the recently active approaches focusing on smaller models utilizing parameter transfer, and adaptations (Kumar et al., 2023). However, these techniques require intense hyper-parameter selection and tuning, and slightly longer training times, which was not in the scope of this study.

Finally, we aim to incorporate BANTUBERT¹, and zaBANTUBERT² models trained with monolingual South African corpora in our future work.

¹<https://huggingface.co/dsfsi/BantuBERTa>

²<https://huggingface.co/dsfsi/zabantu-xlm-roberta>

Acknowledgements

The authors would like to thank the ABSA chair of Data Science for funding and support. We also like to express our gratitude to Matimba Shingange and Michelle Terblanche for their input and reviews of the paper. Finally, the authors would like to thank the Data Science for Social Impact Research lab for their continued support.

References

- Ife Adebara, AbdelRahim Elmadany, Muhammad Abdul-Mageed, and Alcides Alcoba Inciarte. 2023a. [SERENGETI: Massively multilingual language models for Africa](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1498–1537, Toronto, Canada. Association for Computational Linguistics.
- Ife Adebara, AbdelRahim Elmadany, Muhammad Abdul-Mageed, and Alcides Alcoba Inciarte. 2022. Afrolid: A neural language identification tool for african languages. *arXiv preprint arXiv:2210.11744*.
- Ife Adebara, Abdelrahim Elmadany, Muhammad Abdul-Mageed, and Alcides Alcoba Inciarte. 2023b. [Serengeti: Massively multilingual language models for africa](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1498–1537.
- David Ifeoluwa Adelani, Marek Masiak, Israel Abebe Azime, Jesujoba Oluwadara Alabi, Atnafu Lambebo Tonja, Christine Mwase, Odunayo Ogundepo, Bonaventure FP Dossou, Akintunde Oladipo, Doreen Nixdorf, et al. 2023. [Masakhanews: News topic classification for african languages](#). In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 144–159. Association for Computational Linguistics (ACL).
- David Ifeoluwa Adelani, Dana Ruiters, Jesujoba O Alabi, Damilola Adebajo, Adesina Ayeni, Mofe Adeyemi, Ayodele Awokoya, and Cristina España-Bonet. 2021. [Menyo-20k: A multi-domain english-yorùbá corpus for machine translation and domain adaptation](#). In *AfricaNLP*.
- 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*, pages 4336–4349, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Laurie Burchell, Alexandra Birch, Nikolay Bogoychev, and Kenneth Heafield. 2023. [An open dataset and model for language identification](#). In *Proceedings of the 61st Annual Meeting of the Association for*

- Computational Linguistics (Volume 2: Short Papers)*, pages 865–879.
- Isaac Caswell, Theresa Breiner, Daan van Esch, and Ankur Bapna. 2020. Language id in the wild: Unexpected challenges on the path to a thousand-language web text corpus. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6588–6608.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- 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, et al. 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)*, pages 10883–10900.
- Bonaventure F. P. Dossou, Atnafu Lambebo Tonja, Oreen Yousuf, Salomey Osei, Abigail Oppong, Iyanuoluwa Shode, Oluwabusayo Olufunke Awoyomi, and Chris Emezue. 2022. AfroLM: A self-active learning-based multilingual pretrained language model for 23 African languages. In *Proceedings of The Third Workshop on Simple and Efficient Natural Language Processing (SustainLP)*, pages 52–64, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.
- Meluleki Dube and Hussein Suleman. 2019. Language identification for south african bantu languages using rank order statistics. In *Digital Libraries at the Crossroads of Digital Information for the Future: 21st International Conference on Asia-Pacific Digital Libraries, ICADL 2019, Kuala Lumpur, Malaysia, November 4–7, 2019, Proceedings 21*, pages 283–289. Springer.
- Roald Eiselen and Martin J Puttkammer. 2014. Developing text resources for ten south african languages. In *LREC*, pages 3698–3703. Citeseer.
- René Haas and Leon Derczynski. 2021. Discriminating between similar nordic languages. In *Proceedings of the Eighth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 67–75. Association for Computational Linguistics.
- Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. 2019. Discriminating between mandarin chinese and swiss-german varieties using adaptive language models. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 178–187.
- Tommi Jauhiainen, Tharindu Ranasinghe, and Marcos Zampieri. 2021. Comparing approaches to dravidian language identification. *arXiv preprint arXiv:2103.05552*.
- Amir Hossein Kargaran, Ayyoob Imani, François Yvon, and Hinrich Sch"utze. 2023. Glotlid: Language identification for low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6155–6218.
- Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, et al. 2022. Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.
- R Prasanna Kumar, R Elakkiya, R Venkatakrisnan, Harieni Shankar, Y Sree Harshitha, K Harini, M Nikhil Reddy, et al. 2023. Transformer-based models for language identification: A comparative study. In *2023 International Conference on System, Computation, Automation and Networking (ICSCAN)*, pages 1–6. IEEE.
- Richard Lastrucci, Isheanesu Dzingirai, Jenalea Rajab, Andani Madodonga, Matimba Shingange, Daniel Njini, and Vukosi Marivate. 2023. Preparing the vuk'uzenzele and za-gov-multilingual south african multilingual corpora. In *Proceedings of the Fourth workshop on Resources for African Indigenous Languages (RAIL 2023)*, pages 18–25.
- Vukosi Marivate and Tshephisho Sefara. 2020. Improving short text classification through global augmentation methods. In *Machine Learning and Knowledge Extraction: 4th IFIP TC 5, TC 12, WG 8.4, WG 8.9, WG 12.9 International Cross-Domain Conference, CD-MAKE 2020, Dublin, Ireland, August 25–28, 2020, Proceedings 4*, pages 385–399. Springer.
- Paul McNamee. 2005. Language identification: a solved problem suitable for undergraduate instruction. *Journal of computing sciences in colleges*, 20(3):94–101.
- 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*, pages 116–126, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Asa Cooper Stickland, Xian Li, and Marjan Ghazvininejad. 2021. Recipes for adapting pre-trained monolingual and multilingual models to machine translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3440–3453.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with bertserini. *NAACL HLT 2019*, page 72.