

Improved Text Language Identification for the South African Languages

Bernardt Duvenhage

bernardt@feersum.io

Feersum Engine

Praekelt Consulting

Johannesburg, South Africa

Mfundu Ntini

mfundo@praekelet.com

Engineering Team

Praekelt Consulting

Johannesburg, South Africa

Phala Ramonyai

phala@praekelet.com

Engineering Team

Praekelt Consulting

Johannesburg, South Africa

Abstract—Virtual assistants and text chatbots have recently been gaining popularity. Given the short message nature of text-based chat interactions, the language identification systems of these bots might only have 15 or 20 characters to make a prediction. However, accurate text language identification is important, especially in the early stages of many multilingual natural language processing pipelines.

This paper investigates the use of a naive Bayes classifier, to accurately predict the language family that a piece of text belongs to, combined with a lexicon based classifier to distinguish the specific South African language that the text is written in. This approach leads to a 31% reduction in the language detection error.

In the spirit of reproducible research the training and testing datasets as well as the code are published on github. Hopefully it will be useful to create a text language identification shared task for South African languages.

Index Terms—Naive Bayesian text classification, lexicon based text classification, text language identification

I. INTRODUCTION

Virtual assistants and text chatbots seem to be gaining much popularity, but to be accessible to South Africans these software agents need to understand our local languages. South Africa has 11 official languages belonging to a couple different language families. Afrikaans (afr) and English (eng) are Germanic languages. isiNdebele (nbl), isiXhosa (xho), isiZulu (zul) and siSwati (ssw) belong to the Nguni family of languages. Sepedi (nso), Sesotho (sot) and Setswana (tsn) belong to the Sotho-Tswana family of languages. Finally, Xitsonga (tso) belong to the Tswa-Ronga family and Tshivenda (ven) belong to the Venda family. Many of these languages are under-resourced and further work is required to build software agents that are fluent in the country's rich vernacular.

Text language identification (LID) is an important early step in many multi-lingual natural language processing (NLP) pipelines because many of the later steps are still language dependent. Given the short message nature of text based chat interactions and the possibility of code switching the language identification system might only have 15 or 20 characters to make a prediction. However, lower LID accuracies may be expected for short text due to fewer text features being available during classification. Any errors that occur early in an NLP pipeline are also potentially compounded by later processing steps.

This paper gives an overview of the related LID literature in Section II, a discussion of the chosen baseline classifier in Section III followed by the discussion of the paper's contribution to the improvement of LID on short pieces of text in Section IV. Comparative results are presented in Section V followed by some concluding remarks and suggested future work in Section VI.

A further contribution of this work is that the training and testing datasets as well as the code are published on github. Hopefully it will be useful to create a text language identification shared task for South African languages.

II. RELATED WORKS

An LID system for long texts based on normalised histograms of character n-grams is presented in [1]. A similar system that also successfully used character n-grams for doing LID of long texts is presented in [2]

A frequency based n-gram difference based classifier and a support vector machine (SVM) that uses the n-gram frequencies as features are discussed in [3]. Error rates of approximately 0.3% are achieved over large text window sizes. It is also found that the SVM's performance is better than the n-gram based estimator's, but at a much greater computational cost.

In [4] a spell-checker from the South African Centre for Text Technology (CTeXT) is applied to do LID. A sentence level accuracy of 97.9% is achieved on texts of approximately 400 characters in length.

A naive Bayes classifier with various character n-gram text features, called *langid*, is discussed in [5]. In the current paper *langid* is also trained on the South African languages and used as an LID reference in Section V.

A difference in n-gram frequencies classifier, a naive Bayes text classifier with n-gram features and an SVM are evaluated for LID in [6]. The Bayesian classifier is reported to be the most accurate in practise at 17% error on texts of 15 characters.

An SVM and a naive Bayes classifier for language identification of individual words are compared in [7]. The system was trained to identify afr, eng, sot and zul which, except for afr and eng, are all from different South African language

Avrg. accuracy = 1.0, F-Score = 1.0

	Germanic		Nguni				Sotho-Tswana			Tswa-Ronga	Venda
240 characters	afr	eng	zul	xho	ssw	nbl	nso	sot	tsn	tso	ven
afr	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
eng	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
zul	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
xho	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ssw	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
nbl	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000
nso	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000
sot	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000
tsn	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000
tso	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000
ven	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000

0.1 0.3 0.5 0.7 0.8 0.9

Fig. 1. Confusion matrix of the baseline classifier and a test set with strings of length 200-300 characters.

families. Accuracies of around 85% - 95% on single 10 - 15 character words are reported.

In [8] joint sequence models are used to further improve the accuracy of LID of single words 10 - 15 characters long. Accuracies of around 97.2% are reported when labelling text as afr, eng, sot or zul. The training data used is from the National Centre for Human language Technology's language dictionary word lists.

In [9] Char2Vec and an LSTM are used to do end-to-end trained LID. Char2Vec is used to get word embeddings which are then combined via an LSTM. Once trained the LSTM is able to predict a language for each word in the sentence. The significant text features are automatically learned and text pre-processing and cleanup is not required. Near state-of-the-art performance is reported on code switching LID shared tasks.

Recently a lexicon based LID [10] was applied to under-resourced languages. It is not clear what the character length of the training and testing samples were, but the reported LID accuracies are in the low 90's.

III. LID BASELINE FOR SOUTH AFRICAN LANGUAGES USING CHARACTER N-GRAMS

The use of a naive Bayes classifier with character n-gram text features has become the standard for sentence level text LID [6]. The baseline used in this paper is sklearn's multinomial naive Bayes classifier with character 5-grams.

The data used in this paper is the NCHLT Text Corpora [11] [12]. The NCHLT Text Corpora data was cleaned up a bit by replacing numbers and all punctuation except ' ' with spaces. All other characters such as š were left unmodified.

3000 training samples and 1000 test samples per language were randomly chosen from the subset of full sentences in

the CText data that are 200-300 characters long. Using more training data doesn't significantly improve the LID accuracy for long sentences as shown later in Figure 3 in the results section. Binary text features were used as opposed to integer feature counts. Later in the paper a classifier trained on all 4000 of these long sentences are reused for classification of short sentences.

Initially the trained classifier had an accuracy and F-score of 99.5%. However, some of the mis-predicted sentences were spotted to be mislabelled in the NCHLT data. The mis-predicted sentences were few enough to all be checked manually and were indeed found to be mislabelled in the CText data. Approximately 0.468% of the data was correctly relabeled in this manner. The updated datasets are hosted with the LID code on github at <https://github.com/praeckel/feersum-lid-shared-task>.

After cleaning up the data the trained classifier had an accuracy and F-score of 99.9909% ≈ 100.0%. This baseline classifier already outperforms previous work [6] on long sentences.¹ Figure 1 shows the confusion matrix for the test set.

The Google Translate API was also used to verify the results of the n-gram classifier for the languages it understands (i.e. afr, eng, sot, xho and zul). The Google results correlated with all predictions except for some differences between isiXhosa and isiZulu - which belong to the same language family. Approximately 0.09% of the Google results differed from the baseline results, but again all of these could be checked manually and were found to have been incorrectly labelled by

¹The authors of the earlier work didn't attribute their data to CTEXT, but it is curious that they achieved the same accuracy as we did before the data cleanup.

Avrg. accuracy = 0.930, F-Score = 0.929

	Germanic		Nguni				Sotho-Tswana			Tswa-Ronga	Venda
15 characters	afr	eng	zul	xho	ssw	nbl	nso	sot	tsn	tso	ven
afr	0.981	0.008	0.000	0.002	0.000	0.000	0.000	0.002	0.007	0.000	0.000
eng	0.004	0.987	0.003	0.000	0.000	0.000	0.001	0.002	0.001	0.002	0.000
zul	0.000	0.008	0.808	0.052	0.044	0.083	0.001	0.001	0.001	0.001	0.001
xho	0.000	0.003	0.057	0.894	0.011	0.033	0.000	0.000	0.000	0.000	0.002
ssw	0.000	0.002	0.014	0.001	0.973	0.010	0.000	0.000	0.000	0.000	0.000
nbl	0.000	0.002	0.052	0.016	0.022	0.907	0.000	0.000	0.000	0.001	0.000
nso	0.001	0.005	0.002	0.000	0.000	0.001	0.873	0.046	0.071	0.001	0.000
sot	0.000	0.003	0.000	0.000	0.000	0.000	0.019	0.924	0.052	0.001	0.001
tsn	0.004	0.003	0.000	0.000	0.000	0.000	0.064	0.030	0.898	0.000	0.001
tso	0.000	0.004	0.000	0.000	0.001	0.001	0.000	0.000	0.000	0.991	0.003
ven	0.000	0.002	0.002	0.001	0.000	0.000	0.001	0.000	0.000	0.002	0.992

0.1 | 0.3 | 0.5 | 0.7 | 0.8 | 0.9

Fig. 2. Confusion matrix of the baseline classifier and a test set with strings of length 15 characters.

Google Translate. A side effect of this validation is that one can be assured that Google’s API is relatively accurate for the South African languages it supports.

The baseline model trains and tests in 90 minutes on a single core of a 3.30 GHz i5 CPU, uses below 2GB of RAM during training and the trained model is approximately 50MB in size. Long sentence language detection therefore seems to be a solved problem for at least the 11 official South African languages and given that the training data is from a similar domain as one’s production environment. Others [7] have also noted that one easily achieves 100% LID accuracy given 300 characters of text.

IV. LID OF SHORT STRINGS

Short sentence data was derived from the cleaned dataset by selecting from the start of the long sentences 200, 100, 50, 30 or 15 characters plus any characters required to not split a word. A potential problem with the alternative sliding window approach that others have used is that the fragmented start and end words affect the classifier performance for short sentences and such an approach would also prohibit the use of a lexicon based classification algorithm.

The classifier’s F-score on short pieces of text are shown in Figure 3 for training set sizes from 1000 to 4000 samples. The datasets size prevents using more samples to train the baseline classifier, but from the graph it seems that the short sentence performance could benefit from using more than 4000 training samples. Although not the focus of this paper note that our baseline already outperforms earlier reported results [6] of 1.5% on 100 char strings and 17% for 15 chars. The current baseline achieves 0.1% error on 100 char strings and 7.0% error for 15 chars.

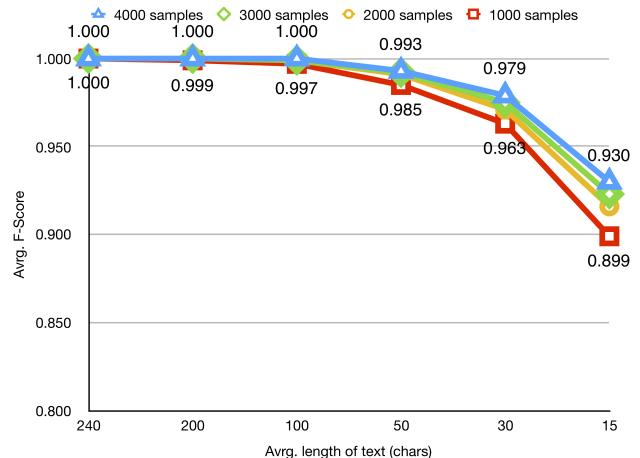


Fig. 3. The baseline classifier’s F-score for shorter text fragments. The different graphs show how the number of full sentence training samples influence the short sentence LID results.

Figure 2 shows the confusion matrix for short 15 char strings. As others have noted, there is some confusion between languages of the same family. This can be clearly seen from the widening of the diagonal into the family blocks. Note also the limited confusion between language families.

Figure 4 shows the family confusion matrix for short sentences of 15 characters. The limited confusion between language families should be clear. Also note that the average accuracy and F-score of classifying a short string into language families is 99.2% while the average accuracy of classifying a short string all the way into a language is only 93.0%

Avg. accuracy = 0.992, F-Score = 0.992					
15 characters	Germanic	Nguni	Sotho-Tswana	Tswa-Ronga	Venda
Germanic	0.990	0.003	0.006	0.001	0.000
Nguni	0.004	0.994	0.001	0.001	0.001
Sotho-Tswana	0.005	0.001	0.992	0.001	0.001
Tswa-Ronga	0.004	0.002	0.000	0.991	0.003
Venda	0.002	0.003	0.001	0.002	0.992

0.1 0.3 0.5 0.7 0.8 0.9

Fig. 4. Confusion matrix of language families of a test set with strings of length 15 characters.

The baseline classifier therefore performs very well (99.2% accurate) at classifying even short 15 character sentences into their language families. Such accurate classification of the language family is possibly good enough to enable a software agent to interpret and act on short sentences. However, in some cases one might want to identify the specific language that a piece of text is written in.

A key realisation is that although Sesotho, Setswana and Sepedi are strongly related, certain words might appear in one language, but not in the others within the family. The same is true of isiZulu, siSwati, isiXhosa and isiNdebele. Therefore, a second lexicon based classifier may be useful to distinguish languages within the same language family.

To test this idea, a lexicon is created from all the sentences in the cleaned language corpuses (4.1k - 25k samples per language). During language identification the naive Bayes classifier result is used to classify the text as belonging to a language family after which the language lexicons are used to count how many words of each language in the family is present in the input. If one language in the family dominates then it is chosen as the language label otherwise the naive Bayes result is taken as the most informative and used as the language label.

V. MORE RESULTS AND ANALYSIS

Figure 5 shows the comparative accuracies of langid_97, langid_za, Google Translate’s language detection API and the naive Bayes baseline classifiers. langid_97 is the langid model included with the langid package that was trained on 97 languages. langid_97 and Google translate’s detector are pretrained and were tested on the full 4000 (training + testing) cleaned samples per language. The South African languages that langid_97 supports are afr, eng, xho and zul. Google’s detector additionally supports sot. For the pre-trained models only the supported languages were included in the accuracy and F-score estimates. The other langid model, langid_za, we trained on the cleaned-up long (200-300 character) full sentences used in this paper. An ideal accuracy for language identification is above 99% so that less than one in a hundred predictions fail.

The proposed lexicon classifier on its own achieves an accuracy of only 89.8% and an F-score of 89.7%. However,

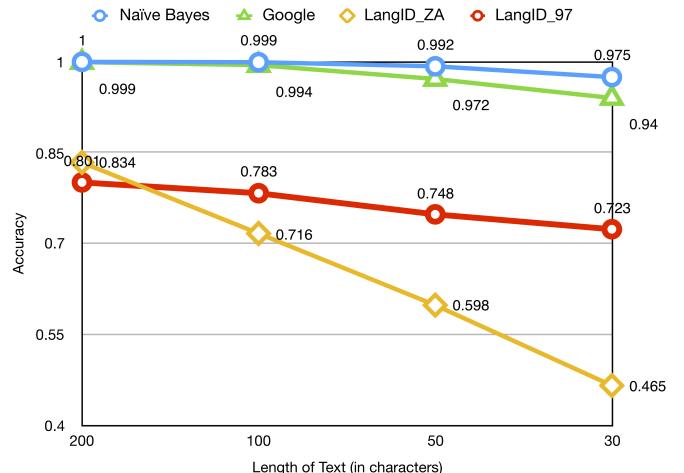


Fig. 5. Accuracy of the baseline NB, Google Translate, langid_ZA (trained on cleaned data) and langid_97 (built-in langid model trained on 97 languages).

when used in combination with the baseline classifier the lexicon classifier’s result is only used when it responds with a high confidence which results in an overall reduction in error. Figure 6 shows the updated confusion matrix for short 15 char strings classified using the simple two stage classifier. The noise level in the results using the 1000 testing samples seem to be in the order of 0.001.

The resulting short sentence LID accuracy is 95.2% which is 31% reduction in LID error over the baseline classifier. The family LID accuracy stays unchanged at 99.2% accuracy. As mentioned previously, earlier works by Botha and Barnard [6] did a full language classification, but only achieved a 17% error (83% accuracy) for short sentences of 15 characters. Giwa and Davel [8] achieved what is essentially a language family LID accuracy of just over 97% for single words of 10 - 15 characters long.

VI. CONCLUSION

A. Summary

The baseline naive Bayes classifier is shown to be more accurate than Google Translate’s pre-trained language identification API, the pre-trained langid_97 and a langid model trained on the cleaned data used in this paper. The baseline also outperforms earlier reported results on South African languages as discussed in Section IV.

Adding a lexicon to the baseline classifier reduced LID error by 31%. The resulting short sentence LID accuracy is 95.2%. When compared to the previous reported result [6] of 83% the current model reduced the error from 17% to 4.8% which is a 3x reduction in error.

The improved dataset and code are hosted at <https://github.com/praekekt/feersum-lid-shared-task>. It is possible that the process of shortening a sentence changes the certainty of its language label when distinguishing words and other features are lost. This would result in a performance ceiling for short sentence LID.

Avrg. accuracy = 0.952, F-Score = 0.952

	Germanic			Nguni				Sotho-Tswana			Tswa-Ronga	Venda
15 characters	afr	eng	zul	xho	ssw	nbl	nso	sot	tsn	tso	ven	
afr	0.981	0.008	0.000	0.002	0.000	0.000	0.000	0.002	0.007	0.000	0.000	
eng	0.004	0.987	0.003	0.000	0.000	0.000	0.001	0.002	0.001	0.002	0.000	
zul	0.000	0.008	0.893	0.026	0.019	0.049	0.001	0.001	0.001	0.001	0.001	
xho	0.000	0.003	0.035	0.939	0.002	0.019	0.000	0.000	0.000	0.000	0.002	
ssw	0.000	0.002	0.012	0.000	0.982	0.004	0.000	0.000	0.000	0.000	0.000	
nbl	0.000	0.002	0.034	0.010	0.006	0.947	0.000	0.000	0.000	0.001	0.000	
nso	0.001	0.005	0.001	0.001	0.000	0.001	0.911	0.034	0.045	0.001	0.000	
sot	0.000	0.003	0.000	0.000	0.000	0.000	0.017	0.937	0.041	0.001	0.001	
tsn	0.004	0.003	0.000	0.000	0.000	0.000	0.057	0.026	0.909	0.000	0.001	
tso	0.000	0.004	0.000	0.000	0.001	0.001	0.000	0.000	0.000	0.991	0.003	
ven	0.000	0.002	0.002	0.001	0.000	0.000	0.001	0.000	0.000	0.002	0.992	

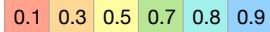


Fig. 6. Confusion matrix of the lexicon improved classifier and a test set with strings of length 15 characters.

B. Future work

The Multinomial NB classifier was used with binary n-gram features. It should be interesting to compare the results of using the normalised feature counts as well.

The short sentence language labels need to be verified and it is important to also gather data from other domains and on modern usage of the various languages. The effect of the lexicon size on the performance of the classifier could also be investigated. It would be interesting to estimate the performance ceiling on LID of short sentences.

Stemming of the lexicon could possibly ensure that the LID generalises better to unseen words. However, stemming in many of the South African languages hasn't been addressed yet.

It should also be interesting to train an end-to-end Deep RNN or CNN to do language ident in the South African context as opposed to manually engineering the two stage classifier.

REFERENCES

- [1] H. Combrinck and E. Botha, "Text-based automatic language identification," in *In Proceedings of the 6th Annual Symposium of the Pattern Recognition Association of South Africa*, 1994.
- [2] W. Cavnar and J. Trenkle, "N-gram-based text categorization," in *In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, 1997.
- [3] G. Botha, V. Zimu, and E. Barnard, "Text-based language identification for the south african languages," in *SAIEE Africa Research Journal*, 2006.
- [4] W. Pienaar and D. Snyman, "Spelling checker-based language identification for the eleven official south african languages," in *In Proceedings of the Twenty-First Annual Symposium of the Pattern Recognition Association of South Africa*, 2010.
- [5] M. Lui and T. Baldwin, "Langid.py: An off-the-shelf language identification tool," in *Proceedings of the ACL 2012 System Demonstrations*, 2012.
- [6] G. Botha and E. Barnard, "Factors that affect the accuracy of text-based language identification," *Comput. Speech Lang.*, vol. 26, no. 5, pp. 307–320, Oct. 2012. [Online]. Available: <http://dx.doi.org/10.1016/j.csl.2012.01.004>
- [7] O. Giwa and M. Davel, "N-gram based language identification of individual words," in *Proceedings of the Annual Symposium of the Pattern Recognition Association of South Africa (PRASA)*, 2013.
- [8] O. Giwa and M. H. Davel, "Language identification of individual words with joint sequence models," in *Proceedings of the Annual Conference of the International Speech Communication Association INTER-SPEECH*, 2014.
- [9] A. Jaech, G. Mulcaire, S. Hathi, M. Ostendorf, and N. Smith, "A neural model for language identification in code-switched tweets," in *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, 2016.
- [10] A. Selamat and N. Akosu, "Word-length algorithm for language identification of under-resourced languages," *Journal of King Saud University - Computer and Information Sciences*, vol. 28, no. 4, pp. 457 – 469, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1319157815000609>
- [11] S. A. D. of Arts and S. A. Culture & Centre for Text Technology (CText), North-West University, "Nchlt text corpora," 2014, available from <http://www.nwu.ac.za/ctext>.
- [12] "Developing text resources for ten south african languages," in *In Proceedings of the 9th International Conference on Language Resources and Evaluation*, 2014.