

TEXT-BASED LANGUAGE IDENTIFICATION FOR SOUTH AFRICAN LANGUAGES

Gerrit Botha*, Victor Zimu† and Etienne Barnard†

* *Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria, South Africa*

† *Human Language Technologies Research Group, Meraka Institute, Pretoria, South Africa*

Abstract: We investigate the performance of text-based language identification systems on the 11 official languages of South Africa, when n -gram statistics are used as features for classification. In particular, we compare support vector machines, likelihood and frequency difference-based classifiers on different amounts of input text and for various values of n . With as few as 15 words of input text, reliable language identification is possible. Although the support vector machine is generally more accurate as classifier, the additional computational complexity of training this classifier may not be justified in light of the importance of using a large value for n .

Key Words: language identification, n -gram statistics, likelihood model, frequency difference, support vector machine

1. INTRODUCTION

In a multilingual environment, language processing is often initiated with some form of language identification. For example, a general-purpose telephonic help line may require identification of spoken language before routing calls to an appropriate operator [1]. Similarly, a document-processing system may need to determine the language of textual contents before performing tasks such as topic identification, stemming or search. A text-to-speech system may need to determine the language of short pieces of text to determine appropriate pronunciations rules, prosodic models, and phrasing strategies.

In the current contribution, we investigate text-based language identification for the official languages of South Africa (Afrikaans, English, isiNdebele, isiXhosa, isiZulu, Sepedi, Sesotho, Setswana, siSwati, tshiVenda and xiTsonga). The general topic of text-based language identification has been studied extensively, and a spectrum of approaches have been proposed, with the most important distinguishing factor being the *depth of linguistic processing* that is utilized. At the one extreme are approaches that attempt to do a complete parse of text in order to determine not only the language spoken, but also the syntactic structure of the textual fragment. Whenever a successful parse is found, these techniques are (by definition) perfectly accurate. However, they require substantial resources for their development, and can be computationally expensive if a large set of languages has to be considered.

The opposite extreme of complexity attempts to identify language by using simple statistical measures of the text under consideration – for example, letter frequencies, the presence of certain keywords, etc. Conventional algorithms from pattern recognition are then used to perform language identification based on these statistics. Such techniques are the focus of the current research.

In particular, we are interested in methods that employ the frequencies of adjacent groups of n letters as input features. These n -grams are extremely simple to compute for any given text, allow for a straightforward trade-off between accuracy and complexity (through the adjustment of n), and have been shown to perform well in language identification and related tasks in several languages.

The central aim of the current study is to investigate the accuracy that can be achieved for the South African languages using n -gram statistics. This accuracy depends on a number of factors, including the following:

- *The size of the textual fragment used for language identification* - more text will certainly lead to higher accuracy, but in many applications it is desirable to identify the language spoken from a limited amount of text. For example, when code switching is likely to occur, one would like to identify languages based on only a few words; for Short Messages (SMSs), ten to thirty words are typically available.

- *The amount and variety of training data available* - in the major world languages, corpora measured in millions of words are not uncommon, but in most of the languages of the world, algorithm development has to proceed from much less text. The domain from which the training text is extracted is potentially important in that context, since limited training data will lack the full variety of a language, and is therefore expected to generalize less successfully to new domains.
- *The classification algorithm employed* - the number of occurrences of each n -gram in a text to be identified can be thought of as the components of a vector, and numerous algorithms are potentially applicable to the classification of such vectors. However, the number of possible n -grams grows exponentially with n , which restricts the set of applicable algorithms to those that can handle very large feature vectors.
- *The similarity of the languages to be distinguished* - the official South African languages can be grouped into a number of language families; specifically, Nguni (consisting of, isiZulu, isiNdebele, isiXhosa and siSwati), Sotho (consisting of, Sesotho, Setswana and Sepedi), Germanic (English and Afrikaans) and a pair that falls outside these families (xiTsonga and tshiVenda). It is predictably much harder to distinguish between languages that fall within a single family.

We investigate these factors by constructing a number of classifiers based on n -gram statistics for the official languages of South Africa. In Section 3 we give a more detailed description of the data that was used for training and evaluation of our system. Section 4 describes the details of our training and evaluation processes, and Section 5 contains our experimental results.

2. STATISTICAL APPROACHES TO TEXT-BASED LANGUAGE IDENTIFICATION

In this section, we briefly review a number of statistically oriented approaches that have been applied to the task of language identification.

2.1. Methods using N -gram statistics

The n -gram method used for language identification has the advantage that no linguistic knowledge needs to be gathered to build a classifier. The number of possible n -gram combinations depends on (a) the value of n and (b) the number of distinct “characters” contained in the orthography employed. A training model, containing the frequency of individual n -grams, is built from a large corpus of sample text. Identification can be implemented using several techniques.

The n -gram rank ordering [2][3] orders the n -gram combinations in the training and testing data from most to least frequent. The rank difference of the n -gram in the testing and that of the training model is calculated. The sum of all the rank distances is then calculated and the testing language with the smallest distance is identified as the most probable language.

The likelihood-based method calculates the probability of a sequence of n -grams [4], based on some probabilistic model of their relationship. The most common model is the naive Bayesian model, which assumes that successive n -grams are independent of one another, so that the various log likelihoods can be added together. This classifier was implemented in our research, as described in Section 4.

The distance-measure-based classifier was also used in our experiments. This classifier calculates a difference between the n -gram likelihoods observed in the training and test data [5]. The language with the smallest difference measure is chosen as the most probable language.

A support vector machine (SVM) can also be trained using the frequency characteristics of the n -grams [5]. Our third classifier is implemented this way.

2.2. Other statistical methods

Besides n -gram statistics, a number of features can be used for language classification. These include the following:

- Certain languages contain unique or highly distinctive letters, which are therefore useful features for language classification [6]; of course, such letters also produce equally distinctive n -grams, so that distinctive letters can be seen as a sub-class of n -gram based methods.
- Frequent words in a language can be used to build models in the same way as n -gram frequencies [6]. The different classifiers in Section 2.1 can then be applied using the word statistics.

Statistical models other than n -gram models can be built from a sequence of letters. A decision tree based classifier [7] asks a series of questions about the context of the current letter, as defined by the corresponding nodes in a tree. The leaf node represents a language. The language that is most common over a window is chosen.

A discrete Hidden Markov Model (DHMM) can also be used to model a letter sequence [8]. Using the *viterbi* algorithm the most probable language to produce the sequence can be found.

3. DATA

As discussed above, we restrict our attention to the South African languages – in particular, the eleven official languages of South Africa. Text from various domains in all 11 languages were obtained from a variety of sources by D.J. Prinsloo of the University of Pretoria. Various sources were used (such as newspapers, periodicals, books, the bible and government documents), and the corpus therefore spans several domains. The size of the text per language varied from 5 MB to 6 MB, except for Afrikaans (900 kB). Due to the diversity of sources employed, text was not homogeneous and needed some processing in order to be used for building models; for example, consecutive white spaces were replaced with a single space character; numbers and abbreviations were removed, as were links to internet websites and punctuation marks. After this processing the size of the text was significantly reduced: Afrikaans was now 500 kB and the other languages varied from 2 MB to 3 MB. Language specific characters that are found in Afrikaans (è, é, ê, ë, ï, ò, ó, ô, ö, ú, û, and ü) and some other languages (š) improved classification and therefore UTF-8 character encoding was used in building our models.

4. IMPLEMENTATION

4.1. Constructing features

For either a sample or a large amount of text, the frequency count of the n -grams was calculated. The characters that can be included in n -gram combinations were a space, the 26 letters of the alphabet and the other 13 special characters found in Afrikaans and Sepedi. No distinction was made between upper and lower case and no punctuation was included.

Increasing the size of n can increase the accuracy of the classifier (since a larger window of characters is considered), but beyond a certain level the large number of possible n -grams is too sparsely represented in any given corpus, and accuracy decreases thereafter. In addition, the burden on computation and memory usage grows exponentially with n ; we have therefore restricted our attention to the cases $n=3$ and $n=6$.

4.2. Difference-of-frequencies classifier

The difference-of-frequencies classifier creates a model consisting of the frequencies of all unique n -grams found in training text. To classify the test text, a similar model is computed. The difference in n -gram frequencies between test and training models is calculated, with an appropriate norm (we have used the L_1 norm). The test language with the smallest difference is chosen as the most likely language.

$$D_l = \sum_{i=1}^{n-\alpha+1} |l_j(c_i) - x(c_i)|, \quad (1)$$

where α is the character window size, $l_j(c_i)$ is the probability of the n -gram c_i in the language model l_j and $x(c_i)$ is the probability of the n -gram c_i in the test vector \mathbf{x} .

For each language the above metric is computed and this gives an indication of how similar the test string is to the language model. The language profile with the smallest difference is chosen as the most likely language for the string.

4.3. Likelihood-based classifier

For each language, a model is created on the training data in the same way as for the distance-based classifier. The probabilities of all n -grams in the test text, according to this trained model, are multiplied together. The most likely language is selected as the model with the highest probability. (To simplify calculations, probabilities are calculated in the logarithmic domain; these log-probabilities are therefore added together.) For each language a vector of n -gram probabilities is computed by

$$\mathbf{l}_j = \frac{\mathbf{f}_j}{|\mathbf{f}_j|}, \quad (2)$$

where \mathbf{f}_j is a vector of n -gram frequencies calculated from a language document of class j .

In the next equation the log likelihood simplifies calculations by adding logarithmic probabilities and can be expressed as

$$P(L|D) = \sum_{i=1}^{n-\alpha+1} \ln l_j(c_i), \quad (3)$$

where $l_j(c_i)$ is the probability of the n -gram c_i in the language model l_j .

For unseen n -grams a penalty value was assigned. We performed tests using various penalty values and chose the best value based on optimum classification accuracy.

4.4. Support Vector Machine

The support vector machine is a non-linear discriminant function that is able to generalize well, even in high-dimensional spaces [9]. Each n -gram frequency is used as a feature for the classifier. Each sample presented to the classifier is therefore a vector that contains the frequency count of each n -gram in a window of words. The size of the feature space grows exponentially with n , which leads to long training times and extensive resource usage as n becomes large; we therefore limited our experiments to $n=3$. LibSVM [10]

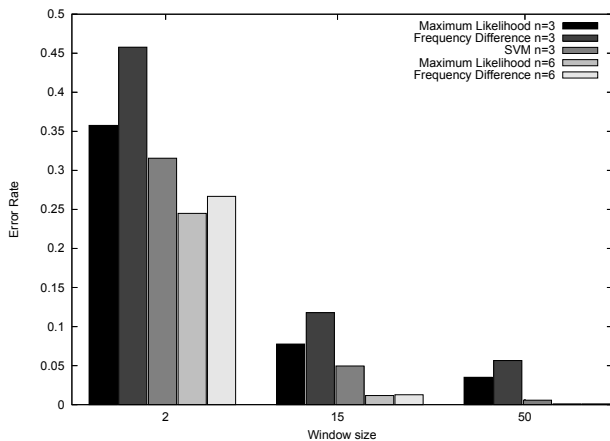


Figure 1: Error rates for different window sizes.

Classifiers were trained with 800 kB of text. For a window size of 50 words, the error rates of the two classifiers with $n=6$ are too small to be visible on this scale.

was used as the classifier in the experiments. A Gaussian kernel was employed, and sensible values for the two free parameters (kernel width and margin-overlap trade-off) were found on a small set of data. These “reasonable” parameters were employed throughout our experiments.

5. RESULTS

5.1. Experimental setup

The collected text was divided into training and test sets. The test text contained 100 000 (100 k) characters per language, whereas the training text contained 400 k characters for Afrikaans and 2 M for all other languages. It would be ideal to have the size of the training text for all languages be the same, but this was the most balanced distribution that we could obtain. Various subsets of this training set were used to examine the relationship between the amount of training text and classification accuracy.

For the likelihood-based and distance-of-frequencies classifiers, configurations with $n=3$ and $n=6$ were compared. For the SVM only $n=3$ was used. The number of words per sample (that is, the number of words used to count the n -grams that are used for classification) was also varied to investigate how this variable interacts with training-set size in determining classifier accuracy. An issue of practical importance is that the SVM classifier needs to be trained with the same window size as the window used during testing, whereas the other two models do not require the same correspondence between training and test conditions. Thus, if we train the SVM with an n of 3 and 15 words, the test set also needs to be constructed in the same man-

ner.

5.2. Tests and evaluation

The average error rates across all languages obtained with various window sizes and n -gram sizes are summarized in Figure 1 for the likelihood-based, distance-of-frequencies and SVM classifiers. For this test the classifiers were built with 800k characters of training data (with the exception of Afrikaans, as described above). Increasing the window size significantly decreases the error rate. For $n=3$ the SVM performs the best for all window sizes, followed by the likelihood and then the frequency-based classifier. Only the likelihood-based and frequency-based classifiers were tested for $n=6$. The $n=6$ likelihood-based classifiers outperformed even the $n=3$ SVM for window sizes of 2 and 15 words. For a window size of 50 words the error rates for these two classifiers were 0.1%. At the same window size, the SVM comes close with an error rate of 0.5%. (We will see below that the SVM will eventually outperform these two classifiers with $n=6$, as more training data is added.)

For the smallest window size of 2 words, classification is much better than chance (which would give an error rate of around 91% on this task). In fact, for some applications such a classifier would be quite usable, especially since the errors tend to fall within the correct language family. This phenomenon is demonstrated in Table I, which shows the confusion matrix obtained with the likelihood-based classifier. Table II contains the language abbreviations. The rows in the confusion matrix correspond with the true class of the input text, whereas the columns reflect the choice made by the classifier. We see that the two groups with the most confusable languages are (Sesotho, Sepedi and Setswana) and (isiXhosa, isiZulu, isiNdebele and siSwati). Since these groups contain closely related languages, it may not be harmful for further linguistic processing if one is confused with the others on a short sample.

The next graphs compare the error rates of various classifiers for the different sizes of training text. The error rates of all classifiers generally decrease as the number of training samples increases, the only exception being an unexpected increase in error rate when 400k training characters are employed. (The consistency of this phenomenon across window sizes and classifiers suggests that it is caused by peculiarities in that block of training data.) For all classifiers, similar classification accuracies are found for 1.6M and 2M training characters, suggesting that asymptotic accuracy is approached with about 1.5M training data.

The experiments show that, for $n=3$, the SVM con-

Table I: Confusion matrix obtained of SVM classifier trained with 2 M characters, for $n=3$ and a window size of two words

SS	SP	ST	XH	ZU	ND	SW	TV	XT	AF	EN	
618	142	204	0	0	1	0	21	12	0	2	SS
187	628	157	1	0	1	0	11	13	0	2	SP
241	224	525	2	0	0	0	4	3	0	1	ST
8	5	3	703	140	85	22	15	16	2	1	XH
4	2	2	163	601	140	53	11	10	1	13	ZU
7	15	5	158	197	576	27	4	9	0	2	ND
14	6	10	54	97	40	740	16	23	0	0	SW
33	24	22	17	2	8	0	858	31	1	4	TV
10	18	11	3	1	7	2	38	910	0	0	XT
7	4	2	2	0	2	2	6	0	934	41	AF
5	7	6	1	1	2	1	8	1	10	958	EN

Table II: Language Abbreviations

SS	Sesotho
SP	Sepedi
ST	Setswana
XH	isiXhosa
ZU	isiZulu
ND	isiNdebele
SW	siSwati
TV	tshiVenda
XT	xiTsonga
AF	Afrikaans
EN	English

sistently outperforms the likelihood and frequency difference based classifier. For $n=6$ the likelihood classifier outperforms the frequency difference based classifier and all classifiers with $n=3$, but it is interesting to note that for a window size of 50 words and 2Mb of training data the SVM with $n=3$ performs best. (The error rates are too small to be visible in Figure 4: the error rate for the SVM was 0.034% and for both the likelihood and frequency difference based classifiers the error rates were 0.062%.)

6. CONCLUSION

We have found that acceptable language-identification accuracy can be obtained with as few as 15 words of input text (in fact, even with 2 words somewhat useful results are obtained). In our tests, the SVM performs with better accuracy than the likelihood and frequency difference-based classifiers when employed under the same circumstances. However, the computational complexity of training the SVM with large numbers of features is substantial – we were therefore not able to train the SVM with $n=6$ (which gave our best result for the likelihood classifier). The likelihood-based classifier may therefore be preferable in practice, because of its overall simplicity.

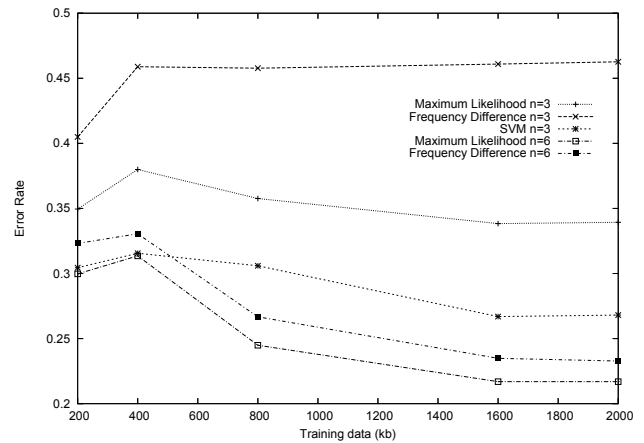


Figure 2: Error rate for a window size of 2 words using various numbers of training samples

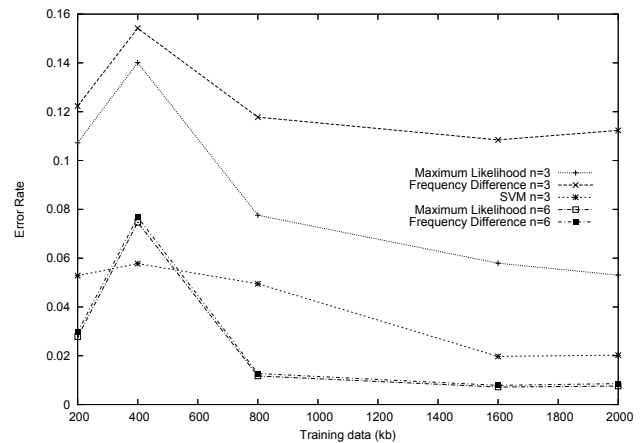


Figure 3: Error rate for a window size of 15 words using various numbers of training samples

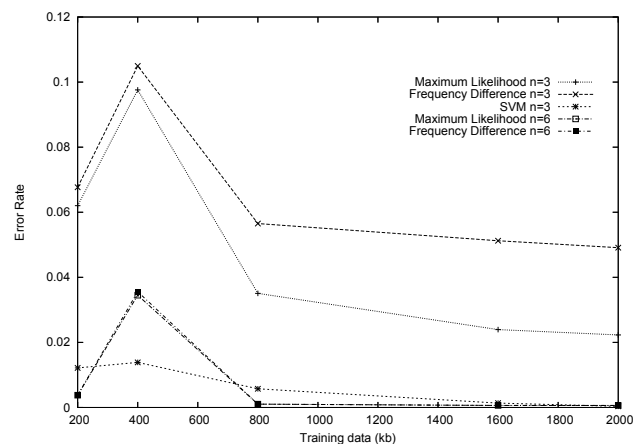


Figure 4: Error rate for a window size of 50 words using various numbers of training samples

Our results clearly demonstrate that larger values of n are preferable and in tests not reported here we have found $n=6$ to be the best value overall. The optimal value of this parameter will depend on the amount of training data available.

In [2] the average error rates for Danish, Dutch, English, French, Italian and, Spanish text was calculated by comparing the n -gram frequencies for different sizes of n (2-7). For a string of 100 characters an error rate of 1.03 % was obtained. This can roughly be compared with a window size of 15 words, where our error rate is for the vir SVM with $n=3$ is 2.02%, and for the likelihood based with $n=6$ is 0.75%. To test if this assumption is reasonable we implemented SVM to classify text based on a 100 character window and 1.92% error rate was found. In the only other research (to our knowledge) including the South African languages in a text-based language identification task, Combrink and Botha [12] reported substantially lower performance for the South African languages than for a set of European languages. Unfortunately, they do not report error rates, so it is not possible to compare results directly.

Our experiments with different training-set sizes suggest that the addition of more training data is the most straightforward route to improving the accuracy of our system. Further improvements can possibly result by maintaining the distinction between lower- and upper-case characters and with algorithmic improvements (e.g. by combining the likelihoods in more sophisticated ways) may also produce additional benefits.

7. FUTURE WORK

During the course of this research, a high-accuracy system for language identification was developed. To further increase the accuracy of the system, a detailed analysis of the errors committed by the system (especially for smaller window sizes) should be undertaken. It will also be interesting to see whether the training data can be filtered using this system so as to improve its accuracy in a bootstrapping fashion. Other possibilities for improved accuracy include the use of other tokens such as frequent words or morphemes, and the use of variable-length strings that are chosen to optimize language discriminability.

We intend to use this system in a number of applications. For example, it can be combined with a Web crawler [10] to collect corpora of monolingual or parallel text. A practical text-to-speech system in a multilingual environment can also benefit from our system to select the correct lexicon and pronunciation dictionaries when code switching occurs within text which is to be synthesized.

ACKNOWLEDGEMENTS

The authors wish to thank Prof. D. J. Prinsloo of the University of Pretoria for sharing his text corpora with us.

8. REFERENCES

- [1] Y.K. Muthusamy, E. Barnard and R.A. Cole, "Automatic language identification: a review/tutorial", *IEEE Signal Processing Magazine*, vol. 11, pp. 33-41, October, 1994.
- [2] B. Ahmed, S. Cha and C. Tappert, "Language Identification from Text Using N-Gram Cumulative Frequency Addition", *Proceedings of CSIS Research Day*, New York, pp. 12.1-12.8, 2004.
- [3] W.B. Cavnar and J.M. Trenkle, "N-Gram-Based Text Categorization", *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, pp. 161-169, 1994.
- [4] G. Grefenstette, "Comparing two Language Identification Schemes", *Third International Conference on Statistical Analysis of Textual Data*, Rome, December, 1995
- [5] C. Kruengkrai, P. Srichaivattana, V. Sorlertlamvanich and H. Isahara, "Language Identification Based on String Kernels", *IEEE International Symposium on Communications and Information Technology*, Vol. 2, pp. 926-929, October, 2005.
- [6] T. Olvecky, "N-Gram Based Statistics Aimed at Language Identification", *Proceedings of the Student Research Conference in Informatics and Information Technologies*, Bratislava, pp. 1-7, April, 2005.
- [7] J. Hakkinen and J. Tian, "n-Gram and Decision Tree Based Language Identification For Written Words", *IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 335-339, December, 2001.
- [8] A. Xafopoulos, C. Kotropoulos, G. Almpandis and I. Pitas "Language Identification in web documents using discrete HMMs", *Journal of Pattern Recognition*, vol. 37, pp. 583-594, 2004.
- [9] A.R. Webb, "*Statistical Pattern Recognition*", 2nd ed., John Wiley and Sons Ltd., pp. 134141, 2002.
- [10] C. Chang and C. Lin (2001), LIBSVM: a library for support vector machines. [Online]. Available: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [11] G. Botha and E. Barnard, "Two approaches to gathering text corpora from the World Wide Web", *Proceedings of the 16th Annual Symposium of the Pattern Recognition Association of South Africa*, Langebaan, South-Africa, pp. 194, November, 2005.
- [12] H.P. Combrink and E.C. Botha, "Text-Based Automatic Language Identification", *Proceedings of the 6th Annual Symposium of the Pattern Recognition Association of South Africa*, Gauteng, South-Africa, November, 1995.