Journal of King Saud University – Computer and Information Sciences (2016) **28**, 457–469

King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

Word-length algorithm for language identification of under-resourced languages

Ali Selamat [a,*], **Nicholas Akosu** [b]

[a] *UTM-IRDA Digital Media Center and Faculty of Computing, Universiti Teknologi Malaysia, 81310 UTM, Johor Bahru, Johor, Malaysia*
[b] *Software Engineering Research Group (SERG), Faculty of Computing, Universiti Teknologi Malaysia, Malaysia*

Received 22 July 2014; revised 27 October 2014; accepted 22 December 2014
Available online 28 November 2015

KEYWORDS

Language identification;
Under-resourced languages;
Resource-scarce;
Digital divide;
Spellchecker model

Abstract Language identification is widely used in machine learning, text mining, information retrieval, and speech processing. Available techniques for solving the problem of language identification do require large amount of training text that are not available for under-resourced languages which form the bulk of the World's languages. The primary objective of this study is to propose a lexicon based algorithm which is able to perform language identification using minimal training data. Because language identification is often the first step in many natural language processing tasks, it is necessary to explore techniques that will perform language identification in the shortest possible time. Hence, the second objective of this research is to study the effect of the proposed algorithm on the run-time performance of language identification. Precision, recall, and F_1 measures were used to determine the effectiveness of the proposed word length algorithm using datasets drawn from the Universal Declaration of Human Rights Act in 15 languages. The experimental results show good accuracy on language identification at the document level and at the sentence level based on the available dataset. The improved algorithm also showed significant improvement in run time performance compared with the spelling checker approach.

1. Introduction

Language identification (LID) refers to the process of determining the natural language in which a given text is written. Pienaar and Snyman (2010) observed that the language of a document can often not be determined on the basis of the file name alone. Moreover, documents on the Internet are not easily deciphered by computers with respect to language identification, because Web documents are traditionally created with the human reader in mind. Beesley (1988) noted that computers cannot use HTML code to determine the language of a web document even though XML and semantic mark-up with entries such as "xml: Lang attribute" and the <meta Lang = "fr"/> constructs have been introduced to tackle these challenges. Many documents still do not make use of metadata tags, or where such tags are used they may

* Corresponding author.
E-mail address: aselamat@gmail.com (A. Selamat).
Peer review under responsibility of King Saud University.

ELSEVIER Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jksuci.2014.12.004

not be used correctly, thereby giving misleading information. According to Beesley (1988) as far as language identification is concerned the best effort is to try and deduce the information from the text itself, knowing that even when metadata are provided they may contain errors. Language identification is often the first step in many text processing systems. Whether it is a machine translation, semantic understanding, categorisation, storage, or information retrieval, text manipulation used online with mobile devices, or email interception, language identification would need to be done first. Therefore, there are serious implications and consequences for not embarking on research in language identification of under-resourced languages. We define under-resourced languages as those languages that do not have (or not enough) digital resources that can be employed for extensive research. The native speakers of such languages either do not use computers or if they do it is usually via a foreign language. This research is focused on languages with little or no digital resources, hence the name 'under-resourced languages'. These are mainly minority languages i.e., languages spoken by a few, but which are gaining importance due to an increasing and widespread use of the Internet and the possibility of such languages being used for communication over the Internet. So far, not much research has been done on identification of these languages probably because they were previously perceived as being less important than the popular languages. In this research we have taken advantage of the fact that the UDHR corpus is a multilingual corpus covering several languages (including some under-resourced languages) thereby making it possible to get a kind of kick-off resource base for this class of languages. Most resource-scarce languages cannot be identified automatically because no research has been done in this area, which means that criminals can use these languages for purposes of information hiding. There are several other consequences. For example, accessibility to Web documents is often hindered due to linguistic diversity on the Internet. Easy worldwide information exchange is one of the core advantages of the Web.

According to Kralisch and Mandl (2006), the language-related link following behaviour reveals important insight into the role of language when accessing information on the Web. Such insight into the role of language helps realise the goal of expanding language participation in Internet communication, thereby reducing the language "digital divide." To bring any language into the fold of natural language processing, some measure of research into its nature needs to be carried out. For many minority languages, however, such a study has yet to be done (Pienaar and Snyman, 2010). Such research would necessarily include or even begin with language identification of the languages in question. In addition, the study of any language on the digital stage needs a significant amount of digital resources. Where such resources are not available, research into these languages becomes difficult. Since language identification is often the first step in many natural language processing tasks (Newman, 1987), it is considered the place to begin. For example, it is only after language identification has been done that an appropriate translator can be selected for a meaningful translation wherever this is required.

Initially the digital divide was perceived as an issue of inadequate access to Information and Communication Technology (ICT) facilities. However, as the accessibility problem was being tackled it was soon realised that language would pose an even bigger problem with respect to information sharing among the peoples and strata of society. Erard (2003) emphasised the need for encoding of languages that are to be used on the Internet, noting that very few languages have so far been encoded which means that all the other languages are left out of the digital information bracket. On the other hand, Martindale, 2002 points out the special difficulties of digital communication in South Africa, a country with 11 official languages which necessitates the creation of websites in each separate language. The author concludes that the problem needs to be addressed by creating automatic translation programmes (Al-Salman, 2008; Bajwa et al., 2012) to facilitate information exchange. We have already noted that for any meaningful translation to happen, language identification must be performed first. It is clear that the relevance and gravity of effect of the various aspects of the language digital divide vary from country to country and from society to society. The implication of inability to identify any language automatically is that such languages become 'invisible' in any multilingual environment like the Internet. Even if documents in these languages are available, other participants do not know what to do with them. The language digital divide really means a division between those languages that are recognisable and those that are not recognisable by computers. By recognisable we mean ability to identify it automatically so that documents written in the language can be treated appropriately as far as natural language processing is concerned.

Language identification of resource-scarce languages using the spelling checker technique was proposed by Pienaar and Snyman (2010). Their experiments demonstrated substantial benefits in the identification of the South African languages using second-generation spelling checkers. In this research we propose an algorithm that improves the algorithm used by Pienaar and Snyman (2010). The proposed method involves pre-processing of input documents, tokenization, and generation of wordlist models using word-length aggregation, aimed at improving computational time gains and efficiency. The proposed models are targeted at solving the current problems of computational complexity, and time-consuming and multilingual identification. The techniques proposed hold the potential of applicability to any other languages as long as they are written in orthographical forms that permit tokenization. Using the lexicon-based approach for language identification as proposed in this research could pave way for further research and generate more digital resources for under-resourced languages. For example, the resulting word list models derived from training data in standard corpora can be further developed into pronouncing dictionaries (Carnegie Mellon University, 2008), thereby enabling applications and research in speech technology. In this research we undertake to find out how this technique will perform with respect to other languages, including languages of the same family. The languages featured in the study include four Nigerian languages (Hausa, Igbo, Tiv, and Yoruba), two South African languages (Ndebele and Zulu), Swahili in East Africa, two Ghanaian Languages (Akuapem and Asante), two South East Asian languages (Bahasa Melayu and Bahasa Indonesia), Croatian, Serbian, and Slovakian. This selection was deliberate in including two Asian languages which are strictly not under-resourced but are closely related languages. The same can be said of Serbian and Croatian which were only included in order to test the performance of our system on closely related languages. The English language is possibly the

most resourced language but is included here to test the viability of the proposed approaches to the richly resourced languages of the world. Our focus is to investigate the performance of an improved lexicon-based approach for language identification of under-resourced languages using an even smaller corpus. The proposed algorithm improves the time performance of language identification by combining the effects of type-token and word-length features in lexicon-based language identification.

The rest of the paper is structured as follows. Section 2 presents related work in this area. Section 3 considers the relevance of language identification of resource-poor languages. In Section 4 we discuss our proposed approach, and in Section 5 we present our results. Section 6 concludes the paper.

2. Related work

Although language identification is often portrayed as a solved problem (McNamee, 2005), much research is still going on in this area because there are yet outstanding issues, including the identification of minority languages, open-class language identification, sparse or impoverished training data, language identification of multilingual documents, standard corpora, and the effects of pre-processing and encoding standards (Da-Silva and Lopes, 2006; Hughes et al., 2006). The dominant approach in the literature is the character-based *n*-gram model. Cavnar and Trenkle (1994) used the *n*-gram profile, based on the most frequent character *n*-grams in a text. They used the ad hoc "out-of-place" ranking distance measure to classify specific texts into one of the existing profiles, with a precision of 90–100% using a 300 *n*-gram profile, in detecting a text of 300 characters. However, the issues of under-resourced languages were not addressed due to lack of corpora. McNamee (2005) applied character *n*-gram tokenization as the basis for language identification in cross language text retrieval contexts. The focus of the research was not on resource-scarce languages.

In one study, Lodhi et al. (2002) proposed a method using the character sequence as opposed to words as the nexus for kernel creation, and showed promising results for discrimination between texts of different languages and for clustering based on string kernels; however, issues of resource-poor languages were not discussed. Kruengkrai et al. (2005) revisited the language identification task and showed state-of the-art results using string kernels, but did not consider performance with respect to under-resourced languages. In another research, Ramisch (2008) investigated the application of *n*-gram language models using a training set of 150,000 sentences and a test set of 11,000 sentences. Such size of data is often hard to find in under-resourced languages. Chew et al. (2009) presented an *n*-gram-based algorithm using a Boolean method to determine the output of matching target *n*-grams to training *n*-grams. They used the algorithm to evaluate how *n*-gram orders and a mixed *n*-gram model affect the relative performance and accuracy of language identification. The experimental results showed a 99.59% correct identification rate on selected languages. Similar results were obtained by Selamat (2011).

Vatanen et al. (2011) used a naïve Bayes classifier based on character *n*-gram models and the ranking method developed by Cavnar and Trenkle (1994). They tested several standard smoothing techniques, including the modified Kneser–Ney interpolation using test samples of between 5 and 21 characters. Under-resourced languages were not considered in their research. Chew et al. (2011) presented two new heuristics to improve an *n*-gram-based language identification algorithm for Asian languages, showing that extension of the training corpus produced improved accuracy. The performance of the algorithm was evaluated based on a written text corpus of 1660 webpages, spanning 182 languages from Asia, Africa, the Americas, Europe, and Oceania. Researchers have demonstrated that the language of electronic documents can also be identified using machine learning techniques or by simply referring to the encoding standards. Machine learning methods that have been used include SVM, neural networks, *n*-gram, decision tree, and ARTMAP (Selamat, 2011; Selamat et al., 2009; Xi and Wenxin, 2010). However, resource-poor languages lack the large digital resources needed for training by machine learning methods to attain optimum results.

Brown (2012) used the *n*-gram approach for language identification of more than 900 languages with impressive results. Brown's method, being applicable to non-textual strings, requires training data of up to 500 k bytes for each language. This makes it unsuitable for under-resourced languages. The research by Brown (2012) is of high impact in tackling such a large number of languages (over 900), a number well ahead of the number done by LexTek International, the most prominent commercial language identification service provider. LexTek-International (2012) currently claims to identify 260 language/encoding pairs on its LexTek language identifier SDK (LexTek International, November, 2012).

A study by Tromp and Pechenizkiy (2011) presented a Graph-based approach for language identification of twitter messages. A similar research was done by Carter et al. (2011), they used semi-supervised priors on twitter messages for language identification based on the assumption that a particular user will only post in one language. Both Tromp and Carter report over 90 per cent accuracy in identification of short text, but these methods are not suitable for multilingual identification i.e. ability to determine that a document is written using more than one language and to indicate which languages are involved. However, both researchers did not focus on the issue of computational time costs. In the research by Hammarstrom (2007), Chew et al. (2011), Brown (2012), Nguyen and Dogruoz (2013), and many others, large amounts of text are required for training, but such amounts are unattainable for under-resourced languages. Also, there is a need to investigate the performance of existing techniques on under-resourced languages, especially languages that have not been investigated in the past (Botha and Barnard, 2012). A good number of methods have been used for language identification over the years, and many researchers have adapted these methods to their research in various circumstances (Chew et al., 2011; Choong et al., 2011; Fiol-roig et al., 2011; Jothilakshmi et al., 2012; Kockmann et al., 2011; Ng and Selamat, 2011; Selamat and Ng, 2011; Sun et al., 2011; Yang et al., 2012; Zampieri and Gebre, 2012). It would be interesting to investigate how these methods would perform given limited training data.

Research shows that even in comparative studies only accuracy is used as yard stick for comparison. The works of Grothe

et al. (2008) and Gottron and Lipka (2010) are good examples. In both studies the authors used only accuracy as a standard to measure and evaluate their research. As a rather rare exception, Amine et al., 2010 used a hybrid technique for language identification. The authors carried out a run time analysis on *F*-measures across three methods, Cosine distance, Euclidean distance, distance of Manhattan, and demonstrated that distance of Manhattan outperformed the other two measures based on computational speed performance. In their research, Winkelmolen and Mascardi (2011) proposed investigation of dictionary-based language identification by running text through a spell checker in different target languages and using the number of errors in each language (or the Hamming distance) to determine the language of the text. They noted that such approach could give very accurate results, but would be very inefficient. However, only an empirical study of the performance of this approach can confirm or disprove such opinions.

The first research to address the issues of under-resourced languages was done in 2006 by Botha et al. (2006) who used a likelihood classifier and SVMs to investigate the accuracy achievable for all 11 official languages of South Africa using *n*-gram statistics. They concluded that the computational complexity, for training the SVM for a large number of features, is prohibitive for higher values of *n* (*n*-gram). Pienaar and Snyman (2010) applied second-generation spelling checkers (i.e., spelling checkers that include a morphological analyser/-generator) to perform language identification on the 11 official languages of South Africa. Their choice of technique was predicated on the fact that African languages are resource-poor. They obtained over 95% accuracy with respect to identification of closely related languages (some of the South African official languages are of the same family) and in multilingual identification.

From the foregoing we infer that the spelling checker technique used by Pienaar and Snyman (2010) was successful in the identification of some under-resourced languages and should be tested on more languages in the same category.

3. Relevance of language identification of under-resourced languages

Research in the area of language identification has grown steadily over the years. However, most researchers have concentrated attention on English and the other European languages for obvious reasons. Since English was the original language of most computer designers and users, it became like the official language of computer usage. Naturally, the spread of computer use again flowed first among the European languages, and the most pressing issues then were how information exchange among these languages could be facilitated. Thus, for many years research on language identification and other areas of natural language processing concentrated on the areas of European and later also on the Asian languages. Only recently has there been some interest in expanding the coverage in terms of other languages. Africa has been particularly neglected in the area of language identification research. Indeed, only in 2006 the first African language was featured in any language identification research. In general the coverage of language identification research on the languages of the world has also been low. According to Gordon (2005), more than 7000 languages are listed in the Ethnologue as living languages spoken on earth. However, most of the published research on language identification focuses on languages that are spoken by large numbers of speakers and are also well resourced in terms of written language resources or both (Hughes et al., 2006). The most important reason for the omission of resource-poor languages lies mainly in the fact that the most popular identification techniques are statistical in nature, and these require large amounts of data to build the necessary evaluation models. This situation is bound to change with the development of techniques like the spelling checker method used by Pienaar and Snyman (2010) which is suitable for the identification of under-resourced languages. Such a development will contribute greatly in reducing the negative effects of the language digital divide.

4. The proposed method

The lexicon-based technique is a simple method that identifies the language of a target text by comparing the words in the document with the list of words that exist in the vocabulary of any set of available languages. If a particular language emerges as having, in its lexicon, the largest number of words in the target text, the system concludes that the target document must have been written in that language. We describe the details of the algorithm in the following sections.

4.1. Lexicon-based language identification

Fig. 1 shows the flow chart of the language identification process. The process starts with the construction of the language models, which are generated by tokenizing the training sets in the various languages and eliminating duplicate words after pre-processing. The resulting language models are word lists comprising unique occurrences of words in each language. The resulting word lists thus provide spellchecker models that serve as functional definitions of each language. The testing profile is constructed in the same way by tokenization of the testing set into a word list. The system then computes a binary matrix of the test profile by searching for each word (of the test profile) in all the training profiles.

We adapt standard notation in set theory to explain the working of the lexicon-based technique for language identification. The goal is to determine the status of each word in a document (test set) with respect to the vocabulary of a particular language. Any word w can only be a member of the vocabulary of a language if such a word is a proper word in the language. We define this as property D, such that $D(w)$ is true if and only if w is in a given document, D. Thus, if w is also a member of the vocabulary of any language, this condition increases the chance that the document being tested is in the language with vocabulary, V.

We can express the search as follows:

$$\{w|w \in V \& D(w)\} \quad (1)$$

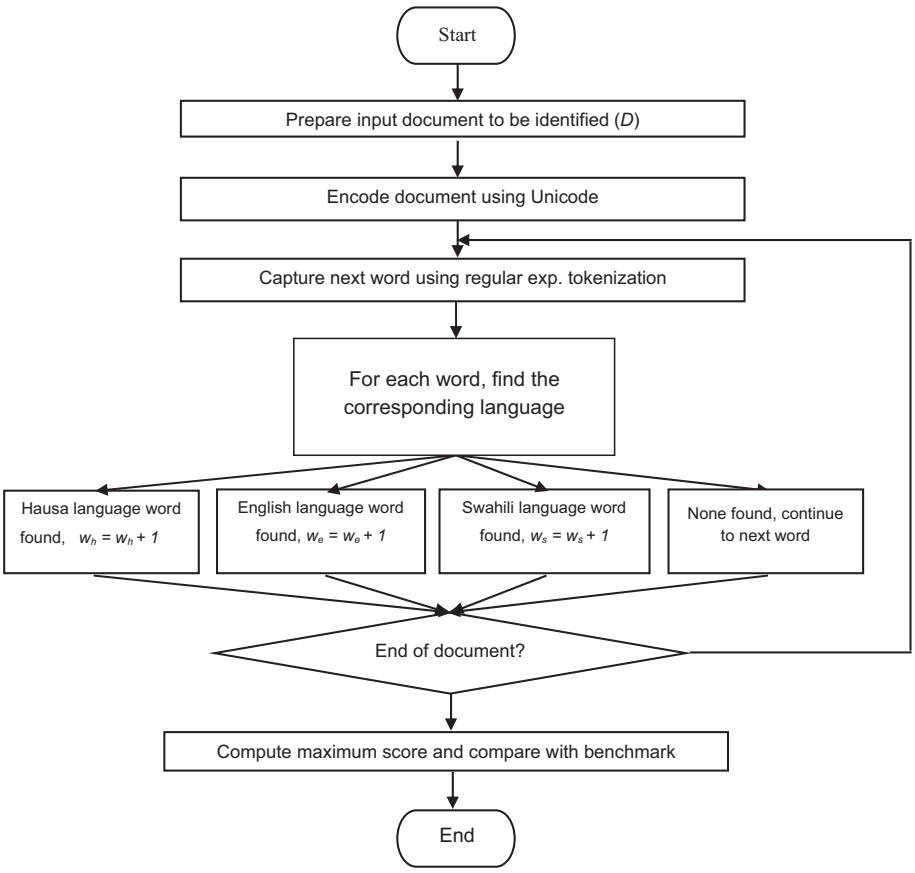

Figure 1 Flow chart of the language identification process.

This means "the set of all w such that w is an element of V (the vocabulary of a particular language) and w has property D." With this we are able to build the binary matrix that is subsequently analysed to determine the language of the document, D (or even a sentence).

4.2. Computing the binary matrix

The binary matrix is computed using a Boolean method, which returns a '1' if the word in the test profile is found in a particular training profile. Otherwise a '0' is returned. After all the words in the target profile have been processed, the system computes the score for the target profile by adding all the matrix values for each training profile.

Hence we describe the processing rule for the score of the various languages for document (D) as follows:

$$\text{Score}_D = \begin{cases} wh = wh + 1 \text{ if selected Hausa word is found} \\ we = we + 1 \text{ if selected English word is found} \\ ws = ws + 1 \text{ if selected Swahili word is found} \\ \text{Continue if none is found} \end{cases} \tag{2}$$

where wh accumulates the score for the Hausa language, we accumulates the score for the English language, ws accumulates the score for the Swahili language, and the desired output, SPLID, is computed as follows:

$$\text{SPLID} = \max\left(\sum_{i=1}^{n} wh_i, \sum_{i=1}^{n} we_i, \sum_{i=1}^{n} ws_i\right). \tag{3}$$

After determining the maximum score using Eq. (3), the system converts SPLID into percentage and compares the result with the benchmark set by the user to confirm language identification.

The process of the methodology can be broken into five steps, as follows:

Step 1: Input training texts and test texts.
Step 2: Generate the training profiles and test profiles.
Step 3: Compute binary matrix for test profile using all training profiles.
Step 4: With the binary matrix as input, determine the highest score using the training profiles.
Step 5: If the highest training profile's score is greater or equal to the benchmark set by the user, then that determines the language of the test profile. Otherwise the language of the test profile is unknown.

The lexicon-based algorithm for language identification is given below (Akosu and Selamat, 2014):

Input: Set of Spellcheckers, L_i; unknown document, D, Benchmark Specification, BM
Output: Language of the input document or document is declared unknown
Begin
  Preprocess unknown document and tokenize into words
  Remove all numeric words and all special characters
  Convert all words into lowercase
  Index word list into set such that each word is searched only once
  for each word $w \in D$
    for each $L_i (i = 1, \ldots, i = n)$ ## i.e. all the language models
      if w in L_i
        $L_i(w) = 1$
      else $L_i(w) = 0$
    end for
  end for
  Compute matrix totals (and %) for all Spellcheckers using the equation:
  $\text{Score} = \frac{1}{n}\left(\sum_{i=1}^{n} a_i\right) \times 100 \quad \text{where } a = \begin{cases} 1 \\ 0 \end{cases}$
  for language in L_i
    if percentage (%) of the highest scoring Spellchecker $\geqslant$ BM
      The language of the document, D is identified as language, L_i
    end if
  print "Document language is unknown"
End

4.3. Lexicon-based model with word length statistics

The use of word-length statistics is targeted at speeding the search process, which is the most important and potentially the most time-consuming activity in the whole process, since all the words in the test profile need to be checked against all the words in the lexica for all the languages under consideration. The idea is that if we search for 3-letter words among 3-letter words our search will be much faster since there is no point in searching for a 3-letter word among 5-letter words where it will never be found. Thus, we propose to speed up the process by organising the vocabulary (word list) by word length. This will be a one-time process, such that once the vocabulary is indexed in this way it is only updated as and when necessary.

The proposed algorithm considers language identification as a problem of analysing the distribution over some set W of variables $W_1 \ldots W_n$, (i.e., words), each of which takes values in the domain $Val\ (W_i)$, the vocabulary of a language. In this case the variables contain words and the input is a data set, $D = \{x_1, \ldots, x_m\}$, where each $w(m)$ is a complete assignment to the variables $W_1 \ldots W_n$ in $Val\ (W_1 \ldots W_n)$ (Akosu and Selamat, 2014).

Our target is to compute an '$N \times K$' binary matrix that can be used to predict the data.

For this purpose we define a scoring function, Score (L:D), which generates the '$N \times K$' matrix relative to the data set, D (Akosu and Selamat, 2014).

We reduce score to summary statistics associated with individual language models, using the generated binary matrix, M.

$M[x_i, u]$ for each $x_i \in Val(X_i) \| u \in \{L\}$, set of language models

$$\text{Score} = \begin{cases} 1 & \text{if } (w \in V \text{ and } D(w)) \\ 0 & \text{otherwise} \end{cases}, \tag{4}$$

Then we compute the sum of scores associated with individual language models using the function

$$\text{Score } (L : D) = \sum_{i=1}^{n} \text{score}(X_i) \tag{5}$$

We then determine the language based on some threshold value set by the user. Suppose the user decides that a document must possess at least 80% of words in a particular language to secure confidence that the document is in the stated language, then we convert the score to percentage as follows:

$$\text{Score } (\%) = 100 * \sum_{i=1}^{n} \text{score}(x_i)/n \tag{6}$$

where n is the number of words in dataset, D (Akosu and Selamat, 2014).

By comparing the highest score to the threshold value we determine the language of the document. If score is less than the required threshold, the document language is unknown. According to Metha and Sahni (2005) the time taken by an algorithm grows linearly with the size of input. Thus, it is traditional to describe the running time of a programme as a function of the size of its input. The implication for this algorithm is that the more words we include in our search space, the more time the algorithm will take to preform language identification. The search space in this algorithm has two dimensions, one over the length of the test set and the other over the vocabulary of all the languages. Metha and Sahni (2005) further observed that in searching a database for a particular piece of information, the searching algorithm's worst case will often occur when the information is not in the database.

Given that the worst-case running time in any search algorithm will occur when looking for non-existent items we expect that reducing such cases will definitely result in a considerable cost saving in running time. We consider it necessary to explore ways to improve the time performance of the proposed algorithm. Consequently, we pose the following question: Is it possible to reduce the running time of language identification algorithms by taking advantage of the structure of natural language? We used two heuristics to investigate this possibility.

First, we investigate the running time of the algorithm by using the type/token heuristic to reduce the search space, thereby reducing the time taken for language identification. In natural language, it has been confirmed that the highest-frequency words take up a large percentage of any document (Manning and Schute, 2002). Thus by searching for word types instead of tokens we expect to reduce significantly the time taken to do language identification, since the number of words,

n, would have been significantly reduced both in the test profiles and the training profiles.

Suppose the number of words in dataset $D = n$. If D contains k word-types ($n \geqslant k$) then the search space will be reduced by $\frac{n-k}{n} * 100\%$ for data set D. As can be observed from Table 1, the type/token frequency distribution for the Hausa language shows that the highest-frequency words occurring 25 or more times number 495. However, these comprise only 8 word types! Further, we observe that these 8 words also constitute 27.1% of the entire document. The figure for Akuapem (last row of Table 1) is even more significant. Here we observe that the highest-frequency words occurring 25 or more times amount to 1167 words comprising only 19 types. However, these 19 types account for 58.9% of the document. This means that we are bound to achieve significant time gains by processing using types instead of tokens.

Table 1 shows the ratio of tokens to types for the 15 languages studied over the frequency range of the most frequent words. We observe from this table that in some of the languages the words occurring 25 times or more account for about 50% of the document. It appears that taking advantage of this statistical composition of the document could yield considerable improvements in time functionality gains. This happens to be the case because for all such high frequency words a lot of time is saved using 'types' for processing because if a particular word (i.e. type) occurred 25 times in the document, such a word would be searched only once and not 25 times, which would be necessary if processing was done using tokens. This is very significant because these high frequency words are usually few and yet make up a high percentage of any documents (see Table 1).

4.3.1. Language identification using word length statistics

Word-length information is the second heuristic that we considered for speeding up the search in lexicon-based language identification. This of course requires a new algorithm designed to deliver the anticipated gains. The algorithm reduces the worst case scenarios (Metha and Sahni, 2005) (i.e., searching for what does not exist) by searching the dictionaries using word-length information, since there is no need to search for a 5-letter word among 4-letter words or 10-letter words. This heuristic will help to reduce the search space and prevent the algorithm from searching for what does not exist, thereby avoiding many worst case situations.

For example, in a typical passage, D of n words, there may be j seven-letter words ($j < n$). In an ordinary search the search space for a seven-letter word would be n. However, using the word-length strategy will reduce the search space in document D by $\frac{n-j}{n} * 100\%$.

Such a reduction in search space is expected to contribute to reduction in running time of the algorithm. Tables 2 and 3 show the distribution of words by word length for two languages in this study.

Word length language identification algorithm:

Input: Set of Spellcheckers, L_i; unknown document, D, Benchmark Specification, BM
Output: Language of the input document or document is declared unknown
Begin:
Pre-process unknown document and tokenize into words
  Remove all numeric words and all special characters
  Convert all words into lowercase and sort words by word length
  Index word list into set such that each word is searched only once
  for each word $w \in D$
    if length (w) = n
      for language in L_i
        if w in lang-word-length (n)
          lang-word-count = lang-word-count + 1 ## increment word count
          end if
        end for
    end if
  end for
  Compute matrix totals (and%) for all Spellcheckers using the equation:
$Score = \frac{1}{n}\left(\sum_{i=1}^{n} a_i\right) \times 100 \quad \text{where } a = \begin{cases} 1 \\ 0 \end{cases}$

  for language in L_i
    if percentage (%) of the highest scoring Spellchecker $\geqslant$ BM
      The language of the document, D is identified
    else document language is unknown
  end for
End

Table 1 Frequency distribution of types/tokens for the 15 languages studied.

Language	Type frequency	No. of tokens	No. of types	% of doc
Hausa	$\geqslant$25	495	8	27.1
Tiv	$\geqslant$25	703	15	39.0
English	$\geqslant$25	606	12	38.5
Malay	$\geqslant$25	331	10	25.3
Zulu	$\geqslant$5	214	14	21.2
Swahili	$\geqslant$25	629	12	37.6
Ndebele	$\geqslant$5	266	23	27.7
Indonesian	$\geqslant$25	338	9	25.1
Croatian	$\geqslant$15	320	12	23.4
Serbian	$\geqslant$15	342	13	23.9
Slovak	$\geqslant$10	343	16	25.7
Igbo	$\geqslant$25	827	16	43.1
Yoruba	$\geqslant$25	1145	19	72.5
Asante	$\geqslant$25	891	17	46.3
Akuapem	$\geqslant$25	1167	19	58.9

4.3.2. The experimental set-up

In this research we studied the performance of the lexicon-based language identification and conducted experiments focused on identification of 15 languages, as listed in Section 1. We investigated the performance of an improved lexicon-based approach for under-resourced languages using an available (small) corpus. However, we included English among the 15 languages studied in order to demonstrate that this approach can apply to other languages as well. For this purpose, we used the Universal Declaration of Human Rights (UDHR) translations in 15 languages. The data were pre-processed by removing punctuation marks and special characters. We then tokenized the text and split it into training sets and testing sets

Table 2 Word length and type/token distribution for the Slovak language.

Slovak language training set (1199 tokens, 640 types)			
Word length	No. of tokens	No. of types	Difference
2	236	41	195
3	96	50	46
4	122	65	57
5	200	80	120
6	137	77	60
7	128	91	37
8	91	81	10
9	77	61	16
10	40	36	4
11	32	26	6
12	19	15	4
13	13	10	3
14	4	4	0
>14	4	3	1
Total	1199	640	559

for each language. The resulting word lists were used to build the language models after converting all characters to lower case. Our approach is based on the concept of using lexicon-based models for language identification in which we built word list-based models from a training set obtained from 9/10 of the UDHR translation for each language. Thus, for each language, we built a lexicon-based model that consisted of the vocabulary derived from the unique set of words in the available data set (9/10 of the dataset). The other 1/10 of the data set was reserved for testing. The words were organised into a sorted set such that each word featured once in the model. The words were further indexed by word length to speed up searching, as explained in Section 4.3.1. Models were labelled by the name of the language from which the text to build the model was derived.

4.3.3. Identification

To determine the language of a dataset (test set), we tokenized it into a word list after pre-processing. Each word in the list was then checked by all available language models and classified according to its acceptance or rejection by the respective language models. We represent the classification in a matrix of binary vectors, as shown in Table 4. We assign scores to each language based on the presence (1) or absence (0) of the current word in the language (row). The words represent the columns, as shown in Table 4.

Table 3 Word length and type/token distribution for the Akuapem language.

Akuapem language training set (811 tokens, 193 types)			
Word length	No. of tokens	No. of types	Difference
2	474	48	426
3	159	60	99
4	81	36	45
5	76	35	41
6	16	9	7
7	3	3	0
8	2	2	0
9	0	0	0
10	0	0	0
11	0	0	0
12	0	0	0
13	0	0	0
14	0	0	0
>14	0	0	0
Total	811	193	618

4.3.4. How the matrix of Table 4 is generated

In this case the sentence under consideration is in row 1. The leftmost column contains the languages tested, while each of the following columns contains the score for the words as listed in the top row (Table 4). The matrix is built progressively by assigning '1' to the language model cell if the 'word' is a valid word in the language, or '0' if there is no such word in the language model. For all words, $w_1 \ldots w_n$ in the unknown document, if w is found in the language model L_i then

$$L_i(w) = 1, \quad L_i(w) = 0 \tag{7}$$

The word w is searched in all the language models $L_1 \ldots L_m$. Thus, in Table 4 we compute the binary matrix for one word as follows:For the word '*The*' we have

Hausa (*The*) = 0, because the word '*The*' does not exist in the Hausa language.
Igbo (*The*) = 0 because the word '*The*' does not exist in the Igbo language.
Tiv (*The*) = 0, because the word '*The*' does not exist in the Tiv language
Yoruba (*The*) = 0, because the word '*The*' does not exist in Yoruba language.

After all the words have been checked, we calculate the score based on the number of words (n) and the binary value for each word (a). Eq. (6) is used to compute the percentage score. The language with the highest score wins. The score is then compared with the benchmarks selected by the user. For example, the user could decide that at least 80% of the words in the test document must be valid words in a particular language for the document to be identified as belonging to that language. Thus, the benchmark can be seen as a confidence measure.

4.4. Evaluation measurements

Standard evaluation measurements were used to evaluate the accuracy of the results:

(i) Cross validation and accuracy;
(ii) Precision, recall, and F_1 measurements.

Table 4 Example of binary matrix, adapted from (Pienaar and Snyman, 2010).

Language	The main difference between man and animal is brain development										%
Hausa	0	0	0	0	0	0	0	0	0	0	0
Igbo	0	0	0	0	0	0	0	0	0	0	0
Tiv	0	0	0	0	1	0	0	0	0	0	10
Yoruba	0	0	0	0	0	0	0	0	0	0	0
Malay	0	1	0	0	0	0	0	0	0	0	10
...	...	...	...	...	...	...	...	...			...
English	1	1	1	1	1	1	1	1	1	1	100

4.4.1. Cross validation and accuracy

We used the 10-fold cross validation to validate the experiments. We applied this to the dataset (UDHR) by splitting it into 10 mutually exclusive subsets of approximately equal size for each language. Ten iterations were used to conduct the experiments. For each iteration, we isolated one part of the dataset for testing while retaining the remaining nine parts as the training set. Then we obtained the accuracy estimation for this first iteration, ae_1. We repeated the steps for the 2nd to the 10th iterations resulting in accuracy estimations, $ae_2 - - ae_{10}$. For each step, accuracy estimation was done using Eq. (8). After all the 10 steps were done and the accuracy for each step computed, we computed the overall accuracy using Eq. (9).

Accuracy estimation is given by

$$ae = \frac{co}{pa} * 100 \tag{8}$$

where, co is the number of correct identifications, pa is the total number of patterns in the dataset, and ae is the accuracy estimation.

Overall accuracy is then computed using

$$Ac = \frac{\sum_{i=1}^{n} ae_i}{n} \tag{9}$$

where, Ac is the overall accuracy estimation.

4.4.2. Precision, recall, and F_1 measurements

The standard information retrieval measures of precision (P), recall (R), and F_1 measures were used to evaluate the effectiveness of our proposed method. They are defined as follows:

$$P = \frac{p}{p+q} \tag{10}$$

$$R = p/(p+r) \tag{11}$$

and

$$F_1 = \frac{2PR}{P+R} \tag{12}$$

Table 5 explains the values of p, q, and r. The relationship between the classifier and the expert is specified using four values, i.e., p, q, r, and s as shown in Table 6. While p measures the ratio of tested documents that are labelled correctly divided by the number of documents identified correctly based on the label given by the user and the system, r gives the probability that a given document is correctly identified as being in a certain language. The F_1 measure is the harmonic mean of precision and recall.

Table 5 Definition of the parameters p, q, and r as used in precision, recall, and F_1 measures.

Value	Meaning
p	True positive
q	False positive
r	False negative
s	True negative

Table 6 Explanation of classification measures.

Expert system	Yes	No
Yes	p	q
No	r	s

5. Experimental results and discussion

In this research we performed three experiments to examine the performance of the improved lexicon-based approach on the 15 languages listed in Table 1. The first two experiments were performed to evaluate the effectiveness of language identification based on precision, recall, and F_1 measurements. We also used the same experiments to measure the time performance of language identification using the proposed algorithm on *type* versus *token* processing and word-length statistics. The third experiment was undertaken to assess the performance of the proposed technique on language identification at the sentence level in a multilingual setting. The results of the various experiments are presented below.

5.1. Effectiveness of lexicon–based language identification

The performance of lexicon based language identification was evaluated using precision, recall, and F_1 measurements. Fifteen languages were used in the experiments, namely, Hausa, Igbo, Yoruba, Tiv, Ndebele, Zulu, Swahili, Akuapem, Asante, Bahasa Melayu, Bahasa Indonesia, Croatian, Serbian, Slovak and English. The average accuracy was 93% with precision of 0.920, recall of 0.925 and F_1 of 0.923. Even in the cases of closely related language pairs like Bahasa Melayu and Bahasa Indonesia, Ndebele and Zulu, Asante and Akuapem, and Serbian and Croatian, the languages were correctly identified. Table 7 shows the confusion reflecting the high rate of shared words among closely related languages.

These results are in line with those of experiments conducted by Pienaar and Snyman (2010). Even the relatively high level of confusion observed in the case of closely related languages can be attributed to the small size of the corpus used in our experiments. This yielded a small word list to serve as yardstick for identification by the algorithm. We present the results of our experiments in two sets, one showing results of identification using the unique tokens (or types) where each word is searched only once in all the lexicon-based models, and the other in which all the tokens are searched.

5.2. Time performance of improved lexicon-based language identification

To speed up the identification process we implemented the language identification algorithm in which the lexicon-based models (word lists) were structured using word length such that the search could proceed by searching the words in the document by word length, i.e., the algorithm proceeds by searching two-letter words only among two-letter words and five-letter words only among five-letter words. The idea is that by using this technique along with searching for each word only once we should be able to cut down drastically on the time-consuming search activity.

To test these theoretical statements, two algorithms were implemented with a view to measuring the time consumption of the lexicon-based language identification. The results are shown in Table 8, in which we randomly selected 50% of the readings for each experiment averaged for all the 15 languages. From the results we observed that implementing *type* as a feature for identification yields a time gain of 22% in the original lexicon-based algorithm. However, in the proposed lexicon-based algorithm implementation we observed a time gain of 32.5% by using *type* instead of *tokens*. Fig. 2 shows the results of average performance of language identification for the 15 languages featured in this research.

However, a further improvement in time performance is observable by comparing the time taken to identify a given document by varying implementation of the two algorithms using tokens and types. In the first instance, we observed that by using tokens, the word length algorithm gives a time gain of 73% over the original spelling checker algorithm. In the second instance we observed an astonishing 89% of time gain by using types in the word length algorithm over the original spelling checker algorithm. Figs. 3 and 4 show the results graphically. This suggests that the word length algorithm is much faster than the original spelling checker algorithm and that this speed advantage can be further increased by taking advantage of the type/token statistics in natural language as illustrated in Table 1.

Table 8 Time performance (secs) – 'word length' and non-word length implementation.

	Without word length		Using word length	
	Using types	Using tokens	Using types	Using tokens
Step 1	0.022	0.026	0.0083	0.011
Step 3	0.021	0.027	0.0093	0.011
Step 5	0.021	0.027	0.0082	0.012
Step 8	0.023	0.026	0.0091	0.011
Step 10	0.022	0.027	0.0083	0.012
Average	0.0218	0.0266	0.0086	0.0114

5.3. Language identification at the sentence level

Our third experiment was on sentence-level language identification. We extracted six sentences each from the 15 languages in our dataset and tested the system. The results revealed an average of 97% correct identification of all the sentences in the closely related language pairs like Bahasa Melayu and Bahasa Indonesia, Ndebele and Zulu, Asante and Akuapem, and Serbian and Croatian. We observed that among the closely related languages a few of the sentences were not identified correctly because the system was not able to decide for either of the closely related languages. We also found one case in which one Malay sentence was identified as Indonesian, which was the most extreme case. In the case of closely related languages the situation arises in which many words are common to both languages. The implication is that the same words keep showing up as belonging to both languages, thereby making it harder for the system to reach a definite decision as to which language the sentence belongs to. The significance of this situation was heightened by the fact that the sentences being considered had a small number of words as is characteristic of sentences in most languages. Given a small corpus such as the UDHR this was rather limiting especially in cases where not all the words in a sentence were found in the spellchecker models generated. However, a larger corpus should present a larger word list that might improve the situation with sentence level identification; though it may also be the case that a larger corpus shows even more overlap between the word lists for the two languages. We shall investigate this in future research. However, experimental results showed that sentence level identification was 100% accurate with respect to all the languages of different language families.

Our python routine even went as far as to give the percentage composition of each language in each sentence. For example, a typical output of the programme was "The input sentence is in Igbo language: 81% Igbo, 18% Yoruba, 18% Tiv, 15% Hausa", 0% English, 0% Malay, etc. Another positive feature of this approach is its ability to decide that the sentence (or document) under consideration is of unknown language. This is usually the case when the percentage composition obtained by the highest scoring spellchecker model is less than the benchmark set by the user. For example, if the user sets 60% as his benchmark, then it means that at least 60%

Table 7 Confusion matrix of LID using type.

	Tiv	Ibo	Has	Yba	Eng	Mal	Zul	Swa	Nde	Akp	Asa	Ind	Cro	Ser	Slo
Tiv	**94.8**	2.1	1.4	1.5	1.4	0	1.2	2.7	1.2	1.4	1.5	1.2	1.5	1.5	1.5
Ibo	2.6	**92.2**	2.4	2.4	0	1.5	0	1.9	0	1.2	2.1	2.5	1.9	2.1	1.7
Has	2.6	1.2	**95.1**	1.6	1.7	1.4	0.9	1.2	0	1.3	2.0	1.5	0	2.0	1.4
Yba	2.8	2.1	1.5	**96.4**	2.3	0	0	2.5	0	2.5	1.6	0	0	1.3	0
Eng	1.1	3.4	0	0	**92.9**	0	0	0	0	0	0	0	0	2.3	0
Mal	0	0	3.9	0	0	**91.2**	1.3	0	0	0	0	23.6	0	0	1.3
Zul	0	1.2	0	0	0	0	**93.3**	0	19.5	0	0	0	0	0	0
Swa	2.5	2.6	2.1	1.3	0	0	0	**91.8**	0	1.4	1.5	0	1.3	1.5	2.3
Nde	1.5	0	0	1.2	0	0	17.3	1.2	**90.7**	0	0	0	0	0	0
Akp	1.6	1.6	2.4	2.0	2.4	0	0	1.2	0	**88.0**	28.6	1.2	2.0	2.2	3.4
Asa	2.6	3.4	2.5	2.3	1.7	1.9	0	2.8	0	18.0	**95.3**	0.9	2.6	2.5	1.6
Ind	0	0	3.3	0	0	25.2	0	0	0	0	0	**89.3**	0	0	0
Cro	1.7	1.8	2.7	1.9	0	0	0	1.8	0	1.8	1.7	0	**88.9**	24.3	2.5
Ser	2.1	1.5	2.3	1.8	0	0	0	2.5	0	2.1	2.1	0	25.1	**89.9**	10.5
Slo	1.8	1.2	2.1	0	1.9	0	0	0	0	0	1.8	1.9	1.8	2.7	**89.1**

Bold values indicate the high rate of shared words among closely related languages.

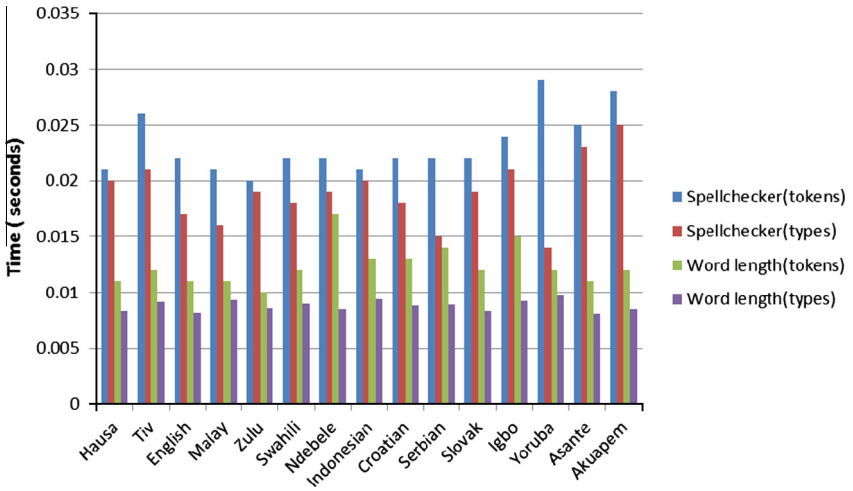

Figure 2 LID time performance for 15 languages.

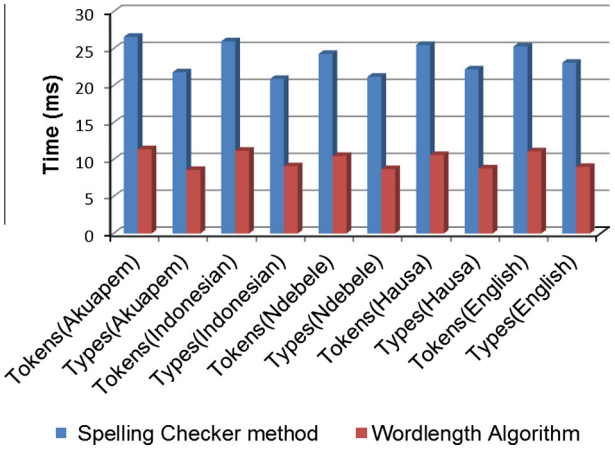

Figure 3 Time performance of identification using *tokens* and *types*.

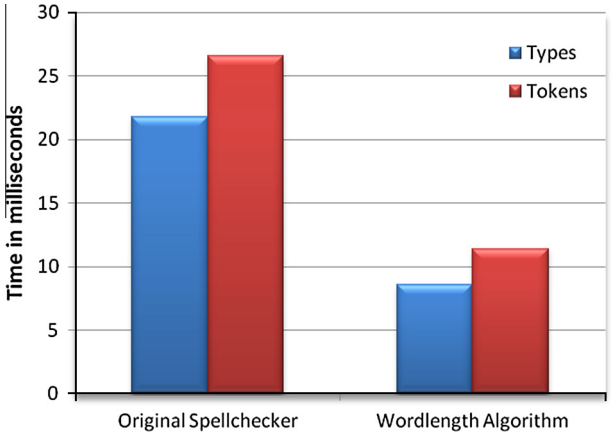

Figure 4 Average time performance of original spellchecker technique and word length algorithm.

of the words in a sentence or document must be confirmed valid in a particular language to establish confidence that the text in question is written in the said language. Thus, if all the available lexicon-based models fail to score up to 60% then the system must report that the text is in unknown language.

5.4. Comparing language identification performance of the lexicon and n-gram based methods.

Using Google's freely available language identification package – *LangDetect* and the UDHR translations for the selected languages we trained the statistical language profiles on 90% of the data set and evaluated the system using 10% of the data set. All the testing sets were identified with precision of 99% except for Zulu language which obtained precision of 86%. However, using a different genre data set (downloaded from the South African Government Services web site) produced some very disturbing results. While the testing sets for Zulu language and English language were identified with precision of 99%, the Ndebele testing set was identified as a Zulu language document with precision of 85%. To our surprise this result did not change even when we increased the size of the testing document 9 times.

Our next experiment involved same genre-multilingual identification in which we combined the testing sets for Akuapem language and Croatian language into one document and submitted it to the system for identification. The document was identified as Asante language with a precision of 42%, Akuapem language also with precision of 42% and Croatian language with precision of 14%. This result is rather disappointing as Asante language was not even part of the composition of the multilingual document! However, the result confirms the observation by Hammarstrom (2007) on statistical multilingual identification. In our next series of tests we reduced the Croatian language component by half and mixed it into the Akuapem language document in 4 variations and obtained the following results:

- By placing the Croatian language portion in front of the Akuapem language portion the document was identified as Akuapem language with precision of 85% and Asante language with precision of 14%.
- Placing the Croatian language portion in the middle gave the result that the document is Akuapem language with precision of 99%.

Table 9 Multilingual identification using lexicon based technique for LID.

Language	Zulu text (%)	Ndebele text (%)	Zulu + Ndebele (%)
Afrikaans	0	0	0
English	3	1	2
siswati	15	15	1
isiXhosa	18	18	1
Zulu	53	26	3
Ndebele	25	80	5
Sesotho	2	1	1
Sepedi	2	1	1
Setswana	2	2	2
Tshivenda	2	1	2
Xitsonga	5	1	3

- By placing the Croatian language portion at the rear of the document, it was identified as Akuapem language with 71% precision, Asante language – 14% and Slovak language – 14%.
- Finally we scattered the Croatian language portion in several parts of the Akuapem language document and the result? Akuapem language – 99% precision. This result is certainly unacceptable Natural Language Processing application that requires efficient multilingual identification.

Next we tested multilingual identification on different genre documents in which we combined Zulu language and Ndebele language texts. We report the following results:

- Equal portions of Zulu language text concatenated with Ndebele language text were identified as Zulu language with 99% precision.
- By placing a small portion of Ndebele language in front of the Zulu language portion the document was identified as Zulu language with precision of 99%.
- Placing the small portion of Ndebele language in the middle gave the result that the document is Zulu language with precision of 99%.
- By placing the small portion of Ndebele language at the rear of the document, it was again identified as Zulu language with −99% precision.
- Finally, scattering the Ndebele language portion in several parts of the Zulu language document did not change the result of 99% precision for Zulu language.

In order to have a fair comparison of the results of the 2 methods (*n*-gram method and the lexicon based method) we used the same data set from the South African Government services website to investigate multilingual identification using the lexicon based technique for language identification. We tested multilingual identification by combining the Zulu language text and the Ndebele language text and evaluated using models trained on 90% of the text in the respective languages. The result is exhibited in Table 9.

From this result it is easy to see that the lexicon based algorithm gives better results for multilingual identification since it consistently shows the presence of any particular language as is present in the multilingual document.

6. Conclusion and future work

Language identification is a core technology in many multilingual applications. Therefore, research on suitable techniques for language identification of under-resourced languages, which make up the majority of languages in the world, is of definite interest and holds the potential for development of digital resources for further research on this category of languages. In this paper, we presented an improved lexicon based algorithm for language identification and experiments carried out to evaluate its accuracy using datasets on 15 languages drawn from the UDHR corpus. Our major objective in this research was to investigate the suitability of the lexicon based approach to language identification of under-resourced languages. The second objective was to study improvements in the run-time performance of the new lexicon based technique through the implementation of optimization algorithms. The proposed improved lexicon based approach was able to maintain acceptable accuracy using experimental datasets and showed outstanding improvements on run-time performance. By including the English language in the list of languages tested we further demonstrated the applicability of the improved lexicon based approach to other languages not belonging to the under-resourced category. In future research we intend to investigate the possibility of incorporating vocabulary extension into language identification using the improved lexicon based algorithm. Also to be considered in future is the investigation of the performance of these approaches using larger corpora.

Acknowledgements

The Universiti Teknologi Malaysia (UTM) and Ministry of Higher Education (MOHE) Malaysia, under research grant FRGS 4F550 and GUP 02G31, are hereby acknowledged for some of the facilities utilised during the course of this research work.

References

Akosu, N., Selamat, A., 2014. A dynamic model selection algorithm for language identification of under-resourced languages. Int. J. Digital Content Technol. Appl. (JDCTA) 8.

Al-Salman, A.S., 2008. A bi-directional bi-lingual translation Braille-text system. J. King Saud Univ. Comput. Inf. Sci. 20, 13–29.

Amine, A., Elberrichi, Z., Simonet, M., 2010. Automatic language identification: an alternative unsupervised approach using a new hybrid algorithm. Int. J. Comput. Sci. Appl. Technol. Math. Res. Found. 7, 94–107.

Bajwa, I.S., Lee, M., Bordbar, B., 2012. Translating natural language constraints to OCL. J. King Saud Univ. Comput. Inf. Sci. 24, 117–128.

Beesley, K., 1988. Language identifier: a computer program for automatic natural language identification of on-line text. In: Proceedings of the 29th Annual Conference of the American Translators Association, 47–54.

Botha, G., Zimu, V., Barnard, E., 2006. Text-based language identification for the South African languages. In: Proceedings of the 17th Annual Symposium of the Pattern Recognition Association of South Africa, Parys, South Africa, 7–13.

Botha, G.R., Barnard, E., 2012. Factors that affect the accuracy of text-based language identification. In: Computer Speech and Language (in press uncorrected proof available on: 16/1/2012).

Brown, R.D., 2012. Finding and identifying text in 900+ languages. Digital Invest. 9, 34–43.

Carnegie Mellon University, 2008. CMU Pronouncing Dictionary.

Carter, S., Manos, T., Wouter, W., 2011. Semi-supervised priors for microblog language identification. Dutch–Belgian Information Retrieval Workshop (DIR-2011), Amsterdam.

Cavnar, W.B., Trenkle, J.M., 1994. N-gram-based text categorization. In: Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval. Las Vegas, Nevada, USA, 161–175.

Chew, C.Y., Mikami, Y., Marasinghe, C.A., Nandasara, S.T., 2009. Optimizing *n*-gram order of an n-gram based language identification algorithm for 68 written languages. Int. Adv. ICT Emerging Reg. 02, 21–28.

Chew, C.Y., Mikami, Y., Nagano, R.L., 2011. Language identification of web pages based on improved *n*-gram algorithm. Int. J. Comput. Sci. Issues 8, 1694–1814.

Choong, Chew Y., Robin Lee Nagano, Y.M., 2011. Language identification of web pages based on improved *n*-gram algorithm. Int. J. Comput. Sci. Issues 8, 1694–1814.

Da-Silva, J.F., Lopes, G.P., 2006. Identification of document language is not yet a completely solved problem. In: Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce, IEEE Computer Society, p. 212.

Erard, M., 2003. Computers learn new ABC'S. Technol. Rev. Info. Trac., 28–30

Fiol-roig, G., Miró-julià, M., Herraiz, E., 2011. Data mining techniques for web page classification. In: Highlights in Practical Applications of Agents and Multiagent Systems, 89. Springer-Verlag, Berlin, pp. 61–68.

Gordon, R.G., 2005. Ethnologue: Languages of the world. SIL International, Dallas, TX.

Gottron, T., Lipka, N., 2010. A Comparison of language identification approaches on short, query-style texts. In: Gurrin, C.H.Y., Kazai, G., Kruschwitz, U., Little, S., Roelleke, T., Ruger, S., Rijsbergen, K. (Eds.), Advances in Information Retrieval, Lecture Notes in Computer Science. Springer, Berlin/Heidelberg, pp. 611–614.

Grothe, L., Luca, E.W.D., Nürnberger, A., 2008. A comparative study on language identification methods. In: Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08).

Hammarstrom, H., 2007. A fine-grained model for language identification. In: Workshop of Improving Non English Web Searching. The Netherlands, Amsterdam, pp. 14–20.

Hughes, B., Baldwin, T., Bird, S., Nicholson, J., MacKinlay, S., 2006. Reconsidering language identification for written language resources. In: Proceedings of the 5th International Conference on Language Resources and Evaluation, Genoa, Italy, 485–488.

Jothilakshmi, S., Ramahungam, V., Palanivel., S., 2012. A hierarchical language identification system for Indian Languages. In: Digital Signal Processing, in press. Corrected proof available on: 27/1/2012.

Kockmann, M., Burget, L., Ernock, J.H., 2011. Application of speaker- and language identification state-of-the-art techniques for emotion recognition. Speech Commun. 53, 1172–1185.

Kralisch, A., Mandl, T., 2006. Barriers to information access across languages on the internet: network and language effects, system sciences, 2006. HICSS '06. In: Proceedings of the 39th Annual Hawaii International Conference on, pp. 54b–54b.

Kruengkrai, C., Srichaivattana, P., Sornlertlamvanich, V., Isahara, H., 2005. Language identification based on string kernels, communications and information technology, 2005. ISCIT 2005. In: IEEE International Symposium on, pp. 926–929.

LexTek-International, 2012. LexTek Language Identifier SDK.

Lodhi, H., Saunders, C., Shawe-Tailo, J., Cristiani, N., Watkins, C., 2002. Text classification using string kernels. J. Mach. Learn. Res. 2, 419–444.

Manning, C.D., Schute, H., 2002. Foundations of Statistical Natural Language Processing. The MIT Press, Cambridge, Massachusetts.

Martindale, L., 2002. Bridging the Digital Divide in South Africa. Linux Journal. <http://www.linuxjournal.com/article.php?sid=5966>.

McNamee, P., 2005. Language identification: a solved problem suitable for undergraduate instruction. J. Comput. Small Coll. 20, 94–101.

Metha, D.P., Sahni, S., 2005. Analysis of algorithms, in: Metha, D.P. (ed.), Handbook of Data Structures and Applications. Chapman & Hall/CRC Computer and information Sc series, Boca Raton, Florida.

Newman, P., 1987. Foreign language identification – a first step in translation. In: Proceedings of the 28th Annual Conference of the American Translators Association, 509–516.

Ng, C.-C., Selamat, A., 2011. Improving language identification of web page using optimum profile 180, 157–166.

Nguyen, D., Dogruoz, S., 2013. Word level language identification in online multilingual communication. EMNLP2013, Seatle, USA.

Pienaar, W., Snyman, D.P., 2010. Spelling checker-based language identification for the eleven official south african languages. In: Proceedings of the Twenty-First Annual Symposium of the Pattern Recognition Association of South Africa, 22–23 November 2010, Stellenbosch, South Africa, 213–216.

Ramisch, C., 2008. N-gram models for language detection.

Selamat, A., 2011. Improved N-grams approach for web page language identification. in: Nguyen, N., (ed.), Transactions on Computational Collective Intelligence V, Springer, Berlin/Heidelberg, pp. 1–26.

Selamat, A., Ng, C.C., 2011. Arabic script web page language identifications using decision tree neural networks. Pattern Recogn. 44, 133–144.

Selamat, A., Subroto, I.M.I., Ng, C.-C., 2009. Arabic script web page language identification using hybrid-KNN method. Int. J. Comput. Intel. Appl. 18, 315–343.

Sun, A., Liu, Y., Lim, E.-P., 2011. Web classification of conceptual entities using co-training. Expert Syst. Appl. 38, 14367–14375.

Tromp, E., Pechenizkiy, M., 2011. Graph-based N-gram language identification on short texts. In: The 20th Annual Belgian–Dutch Conference on Machine Learning (BENELEARN-2011), The Netherlands, pp. 27–34.

Vatanen, T., Vayrynen, J.J., Virpioja, S., 2011. Language identification of short text segments with *n*-gram models.

Winkelmolen, F., Mascardi, V., 2011. Statistical language identification of short texts. In: Proc. ICAART, pp. 498–503.

Xi, Y., Wenxin, L., 2010. An N-gram-and-wikipedia joint approach to natural language identification. In: Universal Communication Symposium (IUCS), 2010 4th International, pp. 332–339.

Yang, J., Zhang, X., Suo, H., Lu, L., Zhang, J., Yan, Y., 2012. Maximum a posteriori linear regression for language recognition. Expert Syst. Appl. 39, 4287–4291.

Zampieri, M., Gebre, B., 2012. Automatic identification of language varieties: the case of Portuguese. In: Jancsary, J. (ed.), Proceedings of KONVENS, ÖGAI, pp. 233–237.