Automatic Identification of Close Languages - Case study: Malay and Indonesian

Bali Ranaivo-Malancon, Non-member

ABSTRACT

Identifying the language of an unknown text is not a new problem but what is new is the task of identifying close languages. Malay and Indonesian as many other languages are very similar, and therefore it is a real difficulty to search, retrieve, classify, and above all translate texts written in one of the two languages. We have built a language identifier to determine whether the text is written in Malay or Indonesian which could be used in any similar situation. It uses the frequency and rank of trigrams of characters, the lists of exclusive words, and the format of numbers. The trigrams are derived from the most frequent words in each language. The current program contains as language models: Malay/Indonesian (661 trigrams), Dutch (826 trigrams), English (652 trigrams), French (579 trigrams), and German (482 trigrams). The trigrams of an unknown text are searched in each language model. The language of the input text is the language having the highest ratio in "number of shared trigrams / total number of trigrams" and "number of winner trigrams / number of shared trigrams". If the language found at trigram search level is 'Malay or Indonesian', the text is then scanned by searching the format of numbers and of some exclusive words.

Keywords: Language identifier, Trigram, Close languages, Malay, Indonesian, Exclusive words

1. INTRODUCTION

As long as human communicate by language, in written or spoken form, it is natural to identify the language used. The constant increase of the number of electronic documents combined with the perpetual lack of satisfaction of users - they request accurate and understandable documents - lead to the elaboration of a variety of tools making possible the automatic classification of these documents by language.

Language identification is not a new problem and many methods have been tried to resolve it. A quick glance to the Web offers a glimpse of the great number of published papers related to the topic and also the number of free or commercial tools that can perform the task of language identification. Identifying the main language of a document is very important for many applications in addition to the fact that not every one can speak and understand more than one language.

A language identifier finds its application in any task involving multilingual electronic documents. Most of the current Internet search engines allow the users to restrict the search to documents written in a specified language. A language identifier is not only used for searching specific documents in the Web. It allows also the evaluation or establishment of the languages used in the Web, "in order to determine the needs for language specific processing it is helpful to know the distribution and relative importance of languages on the web" [1]. In 1997, the Babel team [2] used SILC language identifier [3] in order to establish Web languages hit parade. Another application of language identifier is found in Microsoft Word that integrates a tool to set up the language of the current document and if the text within this document is not the one that has been defined, the text is identified as incorrect. Other applications are e-mail management tools, information retrieval, and speech synthesis [4]. For Natural Language Processing applications, a language identifier can be used as a filter. For example, before submitting a document to a bidirectional machine translation system, the document is processed by the language identifier. Once the language is identified, the translation can start automatically without human interference.

Different types of identification can be obtained by a language identifier. One common result is the identification of the main language of a document. In this case, the output of the language identifier is the name or family language name of this identified language. As we know, many documents contain more than one language, one main language and one or more than one language generally used to cite original texts in their own language. In this situation, the language identifier should identify the main language of the document and the languages of each foreign citation.

In this paper, we present a language identifier that aims to discriminate very close languages like Malay (henceforth BM) and Indonesian (henceforth BI). Besides the two tasks that are identifying the main language and the languages of foreign citations (isolated by quotation marks), the language identifier is able to distinguish close languages and very short texts like titles or snippets.

Manuscript received on June 12, 2006.

The author is with Computer Aided Translation Unit (UTMK) School of Computer Sciences Universiti Sains Malaysia 11800 Minden, Pulau Pinang, Malaysia; E-mail: ranaivo@cs.usm.my

The identification of the language is done in two steps. During the first step, close languages are discarded from other languages present in the language model. At this level, the language identifier uses the frequency and rank of trigrams of characters. During the second step, close languages are discarded between them. For BM and BI, the language identifier uses the list of exclusive words and the format of numbers.

This paper is organised as follows. Section 2 provides an overview of similarities and differences that may exist between two close languages like BM and BI. Section 3 presents some techniques that have been used to identify languages. Section 4 explains the method that we have used in building our language identifier. Section 5 concerns the evaluation of the accuracy of our language identifier by comparing manually its results with the results of two other language identifiers.

Table 1: *priority constant of design patterns components*

Pair of languages	Lexical similarity
Spanish – Portuguese French – Italian	89 %
Spanish – Catalan	85 %
Spanish – Italian	82 %
French – German	29 %
French – English	27 %

2. CLOSE LANGUAGES: SIMILARITIES AND DIFFERENCES

Most of existing language identifiers has been built to recognise the language of any document without taking into account the problem of close languages. Our first motivation was to find an accurate, simple and reusable method that can identify without ambiguity close languages.

Two languages (or a group of languages) are said to be close if they share relatively important similarities at word level. Close languages share many lexical and morphological units with the same spelling, pronunciation and meaning. To be close imply differences since they are not identical. These differences can be used to discriminate close languages.

Two examples of close languages are BM (official language of Malaysia) and BI (official language of Indonesia). The two languages belong to the family of Austronesian languages and both derive from Bahasa Melayu, which was the lingua franca of Southeast Asia since at least the 7^{th} century until 13^{th} century [5-6]. We use BM and BI to illustrate the similarities and differences that may exist between close languages, and also because our language identifier has been built in order to differentiate the two languages.

2.1 Lexical similarities

Two languages are said close or similar if they have an important volume of shared vocabulary. The following table (Table 1), built from the information given by Ethnologue (a catalogue of known living languages, www.ethnologue.com), shows the percentage of vocabulary shared by a pair of close languages.

There is no real and statistical evaluation of the similarity or difference between BM and BI. Asmah [5] did some small tests with her students in 1998. The result showed that 30% of the two Indonesian texts, submitted to 81 Malaysian students, were "odd, unintelligible, and unusual". Asmah divided these unintelligibilities into five categories, and the category "Presence of words and phrases totally unfamiliar to Malaysians" is less than 10%. From this small but very informative test, we may start to state that BM and BI have more 90% lexical similarity.

Table 2: *Examples of borrowings in BM and BI*

	BM (words from English)	**BI (words from Dutch)**
August	Ogos	Agustus
deposit	deposit	deposito
March	Mac	Maret
ticket	tiket	karcis

2.2 Different spellings

The Malay used in Malaysia is written whether with a variety of Arabic script (*tulisan Jawi*) or Roman script (*tulisan Rumi*). A unified spelling of native and borrowed words in BM and BI was introduced in 1972, known in Indonesia as "the Perfect Spelling" (*Ejaan Yang Disempurnakan*), and in Malaysia as "the New Spelling of Malaysian Language" (*Ejaan Baharu Bahasa Malaysia*). Table 3 shows some examples of spelling changes adopted by both countries, Malaysia and Indonesia, in 1972. Besides these changes, the apostrophe for glottal stop (*hamzah*) and the inverted comma were omitted, and

were replaced by the letter 'k'. New letters were included to write borrowed terms: 'o', 'v', and 'x'.

Table 3: *Examples of spelling agreement (1972)*

BM	BI	Reformed Spelling
u	oe	u
ch	tj	c
jj	dj	j
kh	ch	kh
ny	nj	ny
sy	sj	sy
y	j	j

In 1975, the Language Council of Indonesia and Malaysia (MBIM) proposed a set of rules for the coining of technical terms. The introduction of Brunei Darussalam in the Council has changed the name of the Council which became the Language Council of Brunei Darussalam, Indonesia, and Malaysia (MABBIM). Whatever the efforts done by the Council, we must say that all these reforms did not achieve their aims. We still have different spellings for some words in both BI and BM as it is shown in Table 4. The symbol '#índicates word boundary.

Table 4: *Spelling differences between BM and BI*

	BM	BI	MABBIM's rules
adjective	adjektif	adjektiva	ve# ⇒ if
activity	aktiviti	aktivitas	ty# ⇒ ti
phonemics	fonemik	fonemis	ics# ⇒ ik
academic	akademik	akademis	ic# ⇒ ik
sabotage	sabotaj	sabotase	age# ⇒ aj
animal	haiwan	hewan	
newspaper	surat khabar	surat kabar	
zone	zon	zona	

Another difference between BM and BI can be noticed in writing numbers (Table 5). The difference is related to the use of full stop and comma.

In BI, the spelling of monetary value in trillion is not standardised. Sometimes the three digits are separated with a comma (e.g. *Rp 6,565 triliun, Rp1,536 triliun*), sometimes with a full stop (e.g. *Rp 1.000 triliun, Rp 24.000 triliun*).

2.3 Different vocabularies

To express the same concept or meaning, BM and BI may use two different words (Table 6). The two words are totally synonymous.

Table 5: *Format of numbers in BM and BI*

	Full stop	Comma
BM	Decimals are denoted with a full stop 8.9 juta penduduk negara RM11.2 juta RM8.6 bilion 0.6 peratus 26.8 peratus; 0.5 sentimeter 1.57m $3.50 per helai	Thousands are separated by a comma 1,488 kes kencing manis 7,000 pelajar 200,000 bekas tentera 46,000 hektar 341,684 tan RM1,000 RM10,000 RM100,000
	RM8,219,904.79 M$5,000.00 12,438.420 kilogram 1,329.8 km	
BI	Thousands are denoted with a full stop ada 7.000 kasus 500.000 orang Rp 1.000 2.000rupiah waktu 1.062 dtk	Decimals are separated by a comma 3,25 juta pelanggan 1,80 persen 21,0 persen Rp 3,8 triliun Rp 13,09 triliun
	Rp5.000,00 Rp500.000,00	

Table 6: *Different words for the same meaning in BM and BI*

Meaning	Malay	Indonesian
aide-de-camp	adikung	pembantu pribadi
approximately	agak-agak	kira-kira
contagious disease	ambah-ambahan	penyakit menular
economist	ahli ekonomi	ekonom
Equator	Ekuator	Garis Lintang
football	bola sepak	sepakbola
national archives	Arkib Negara	Arsip Nasional
purse; wallet	bekas duit	dompet
recently; of late	akhir-akhir ini	belakangan ini
service sector	bahagian perkhidmatan	sektor jasa

3. LANGUAGE IDENTIFICATION TECHNIQUES

Many parameters can be used to identify the language - or at least to guess the family language - of a document. The identification can be performed at different levels of the document with for each level its good or bad reliability.

3.1 Language encoding information

Markup metalanguages, like SGML, HTML, XML, etc., include attributes that allow the users to specify both language and script used in a document. In Latex, the author can indicate with a specific tag, the language used for any portion of the text which is not the same as the main language. Unfortunately, the use of these attributes is simply recommended. Most of the time, the author of a document omits to indicate the language of the document. However, if this language information is mentioned, there is no standardised format of the way how to represent the name of the language according to "Language Identifiers in the Markup Context" (http://xml.coverpages.org/languageIdentifiers.html # oss). For these reasons, a language identifier cannot rely fully on this unpredictable information. Therefore most language identifiers handle only ASCII texts and do not rely fully on the language and character encoding information.

3.2 Word or sub-word identification

The easiest and simplest solution in the identification of the language of a given text is to perform lexicon look-up for each language available in the library. This kind of approach will create a language identifier that is undoubtedly one-hundred percent accurate. Adopting this method means that all or almost all existing and recognised languages have their electronic dictionaries ready in the language identifier. Unfortunately not all living languages have a dictionary and even less an electronic dictionary. It means also that a good and fast search string must be applied allowing the comparison of a given document with thousand dictionaries.

Giguet [8] declared that "it was possible to categorize long sentences and texts using only linguistic knowledge". Following this idea, the size of the lexicon can be reduced by looking for specific subset of the lexicon: common words for [9][10], grammatical words to discriminate the language of sentences for [11], the most frequent words for [12], and short strings of characters which are unique to each language for [13][9]. The common limitation of this kind of approach is the size of the unknown text. A short input text may not contain those particular words.

Other language identifiers use linguistic segments. Giguet [11] realised that accented letters can be only used to discard languages but not to identify the right language of unknown texts. The reason is that in a short text, accented letters are not so frequent. To improve his approach, he explored other natural language properties: "using knowledge upon word morphology via syllabation: the idea is to check the good syllabation of words in a language, distinguishing the first, middles, and last syllables; in the same way, using word endings and using sequences of vowels or consonants" [11].

3.3 N-gram identification

The most popular statistical language models for language identification are n-gram characters. The idea is to accumulate the frequency or probability of each sequence of characters and detect the sequences that are specific and recurrent to each language. The value of n varies from 1 to 5: 2 and 3 for [12], 1 to 5 for [16] but usually n equals to 3 is often used [9]-[14][15]-[16]-[12][17].

Trigrams work well with short texts, 40-80 characters thought Liberman in his message in Linguist List 2.530 answering Kulikowski on how to identify the language of a text line. Liberman's thought is confirmed by Poutsma's tests [10]. Poutsma realised that "the n-gram method performed best with small amounts of input (i.e. less than 100 characters input)". Prager [18] found during his tests that combining words with unrestricted length and 4-grams gave the best performance.

In 1994, Cavnar and Trenkle [16] have developed a language identifier very similar to our program in a sense that it uses n-grams and computes the similarity between an unknown text and any of the language models by calculating "out-of-place" measure. This measure corresponds to the distance that exists between the rank of an n-gram in the unknown text and in a language model. The language of the unknown text is the language of the language model that has a minimum distance. The performance of Textcat and our language identifier is shown in section 5.

4. THE PRESENT WORK

Our motivation in building a new language identifier is due to the fact that the performance of available language identifiers is not wholly satisfactory when guessing close languages like BM and BI. We started our research by looking for a method that could satisfy the criteria for the best automatic language identification. The tool must require a very small size as language models. It has to be fast since it will be used as a pre-processing tool for other applications. It must be able to handle multilingual documents and short texts. And more importantly, it has to be accurate.

We have built a language identifier for close languages written with Roman alphabet by using trigrams and some linguistic information like the

spelling of numbers and exclusive words. Close languages mean that some words are used only in one language and never in the other. We call 'exclusive words' those words that appear only in one language.

Currently five languages are made available in our language identifier that is BI, BM, English, French, German, and Dutch (Table 7). The language model for each language is very small: between 231 (for German) and 270 words (for English). Those words are considered as being the most frequent words in each respective language. The most frequent words in BI and BM have been mixed as they show two similar lists. For each language, all words are transformed into lower case and the trigrams are extracted, sorted, and reduced to one occurrence with its frequency. The list of trigrams for each language is saved in a simple text file with one line per each trigram and its number of occurrences in the language.

Table 7: *Language models*

Languages	Words	Characters	Trigrams
BI and BM	267	1589	661
Dutch	257	1919	826
English	270	1142	652
French	239	1242	579
German	231	1083	482

The language of an unknown text is given after processing the text through four main modules as it is shown in figure 1.

In the pre-processing module, the unknown text is divided by sentences and all letters are changed into lower case. All sequences of white spaces are reduced to one white space which in turn is changed into one underscore. After the pre-processing module, the unknown text is sent to the 'Trigram segmenter'. This module transforms a stream of characters (corresponding to a sentence or a quoted text) into a list of trigrams. Once a list of trigrams is obtained, it is sent to the module called 'Language Identifier'. At this level, each trigram in the unknown text is searched in each list of trigrams representing each available language. A binary search is an appropriate algorithm for being a fast search in a sorted list. The Language Identifier module calculates the language of the unknown text based on two values: (1) the ratio of the total number of 'shared trigrams' divided by the total number of trigrams in the unknown text, and (2) the ratio of the total number of 'winner trigrams' divided by the total number of shared trigrams. We called 'shared trigrams' trigrams that appear in the unknown text and the current language model. 'Winner trigrams' are trigrams that appear in the unknown text, in at least one language model, and have the highest rank among all language models.

Table 8 provides an illustration of shared and winner trigrams. The BM input sentence '*Saya suka makan nasi goreng.*' means 'I like eating fried rice.' It generates 27 trigrams. The five trigrams 'a _ m', 'a _ s', 'gor', 'i _ g', and 'n _ n' do not appear in any of the language models. The winner trigrams are highlighted in bold.

Table 8: *Language models*

	Dutch	French	German	English	BM and BI
_go	**18**	0	0	8	0
_ma	11	**12**	**12**	3	11
_na	15	0	14	12	**17**
_sa	**17**	12	13	9	12
_su	0	12	0	0	**18**
a_m	0	0	0	0	0
a_s	0	0	0	0	0
aka	0	0	0	0	12
an_	**18**	13	0	8	1
asi	0	0	0	0	**17**
aya	0	0	0	0	**16**
eng	0	0	0	0	**14**
gor	0	0	0	0	0
i_g	0	0	0	0	0
ka_	0	0	0	0	**15**
kan	0	0	**14**	0	8
mak	18	0	0	10	**19**
n_n	0	0	0	4	0
nas	0	0	0	10	**19**
ng_	6	13	**14**	12	11
ore	**18**	13	0	11	0
ren	10	0	9	0	**18**
say	0	0	0	0	**19**
si_	0	11	0	0	0
suk	0	0	0	0	**18**
uka	0	0	0	0	**16**
ya_	0	0	0	0	**14**
Shared	9	7	6	10	19
Ratio-1	0.3333	0.2592	0.2222	0.3703	0.7037
Winner	4	1	3	0	13
Ratio-2	0.4444	0.1428	0.5	0	0.6842

The language of the unknown text is the one that has its ratio-1 and 2 equal or bigger to 0.45. This value has been defined after running many tests on the language identifier. In our example, the text is either BM and BI. To specify to right language of the unknown text, the original text is sent to the module called 'Malay-Indonesian Language Identifier'. This module performs two actions to discard BM and BI texts. First, it checks the format of each number (if there is any) in the unknown text and start counting the differences based on Table 5. Then it looks up in the exclusive lists of words for BM and BI sian. The correct language of the unknown text is the one with the highest number of markers: format of numbers

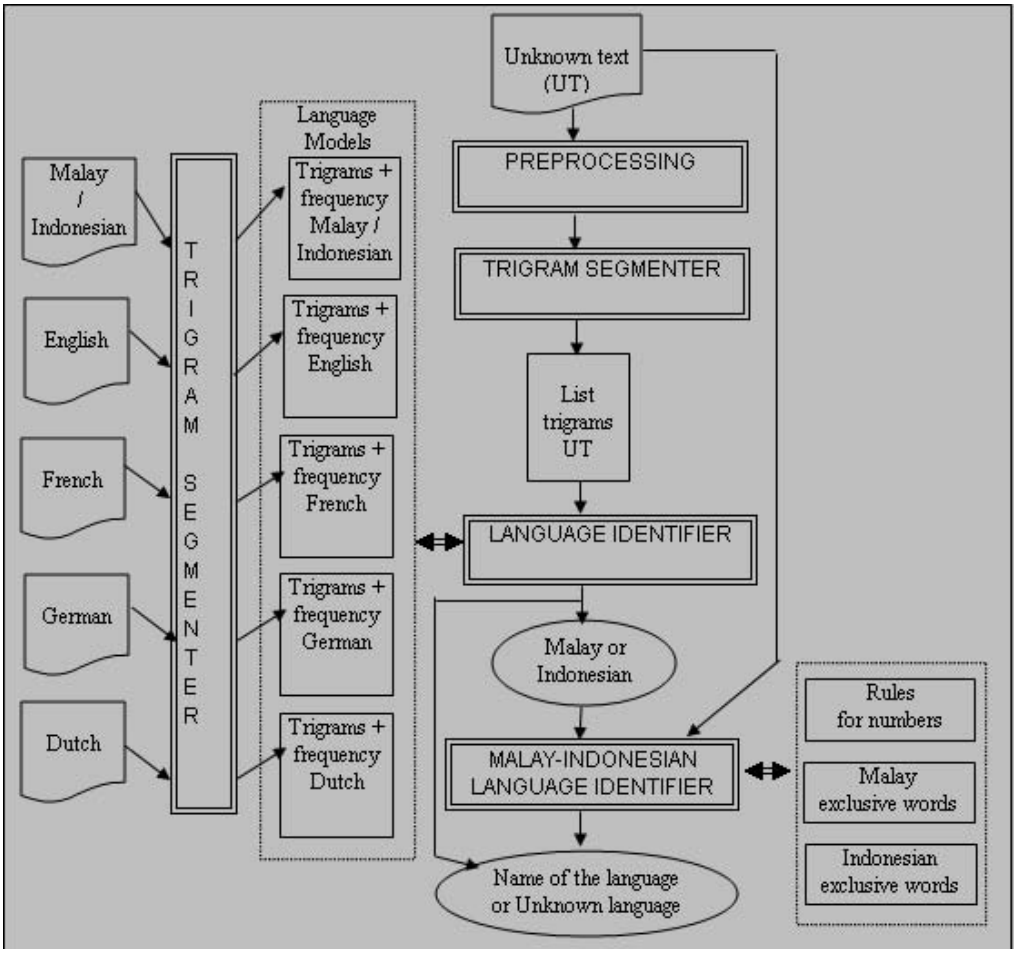

Fig.1: *Architecture of the Malay-Indonesian language identifier.*

and exclusive words.

5. EVALUATION AND RESULTS

A method for evaluating the accuracy of language identifiers is not straightforward as the objective, the discriminators, and the number of languages to be identified is often different.

To evaluate the accuracy of our language identifier, we need to compare its results with some existing tools that contain BM and BI. We have found only four language identifiers that include BM and BI languages: Xerox Language Guesser, Textcat, Rosette Language Identifier, and Lextek Language Identifier SDK. The other available language identifiers consider BI and BM as the same language. We have conducted two tests and evaluate manually the results.

The first test has been done based on the number of words and characters in five BM texts and five BI texts. The ten texts have been chosen at random from our BM and BI corpora. We compared the results given by Textcat [19], Lextek [20], and our program. Textcat is a free online language identifier having more than 75 languages (including BM and BI). Textcat provides most of the time more than one language leaving the user to guess the correct one. Lextek is also a free language identification program that offers over 260 language (including BM and BI) and encoding modules. Lextek is more precise by providing only one possible language. Table 9 (LI stands for language identifier) gives the results of this first test in which our LI performs very well - zero error - compared with Textcat and Lextek. As one of our reviewer highlighted, the good performance of our LI is not a real surprise since it uses lists of exclusive words and rules for writing numbers, two data that Textcat and Lextek do not have. The remark could be true if Lextek does not try to identify clearly BI texts from BM texts.

The second test was conducted by comparing Textcat and our language identifier by looking at the accuracy of each tool at sentence length. The length of a sentence corresponds to the number of words that it contains. We compared 180 sentences (90 BM sentences and 90 BI sentences) chosen at random based on their length: only sentences with less or equal to 10 words. Each sentence length has a set of 10 sentences. For example, there are 10 sentences of length two, 10 sentences of length three, and so on. Table

Table 9: *Identification errors by number of words and characters.*

Input texts	Nb. of words	Nb of char.	Textcat	Lextek	Our LI
BI	25	194	BM or BI	BM	BI
BI	27	180	BM or BI	BI	BI
BI	29	173	BM or BI	BI	BI
BI	48	288	BM or BI	BM	BI
BI	63	444	BI or BM	BI	BI
BM	26	180	BM or BI	BI	BM
BM	28	186	BM	BM	BM
BM	33	199	BM or BI	BM	BM
BM	39	202	BM	BI	BM
BM	41	255	BM or BI	BM	BM
Errors			8	4	0

12 and Table 11 show the results of the comparison when we consider as an error any output with more than one language.

Table 10: *Identification errors by BM sentence length (more than one language in the output).*

Length of sentences	Number of BM sentences	Textcat	Our LI
2	10	10	6
3	10	10	9
4	10	10	10
5	10	10	7
6	10	10	9
7	10	10	8
8	10	10	9
9	10	10	7
10	10	10	8
	90	90	73

The two language identifiers, Textcat and our LI, show very bad results with short BM texts. The results are slightly better with short BI texts. Two reasons may explain these results. Firstly, BI short texts contain more specific words (21 sentences over 90 sentences) than short BM texts (12 sentences over 90 sentences). Secondly, many short texts cannot be clearly identified as BI or BM as all words that form the texts belong to both languages. 147 sentences over the 180 sentences tested are in this situation. If we take in account this second reason, the results of the two language identifiers are reviewed in Table 12.

It appears that Textcat performs slightly better than our language identifier. But since the expected task of a language identifier is to provide "the language" and not "the possible languages" of a given document, we are still comforting in our approach. Table 10 and Table 11 show that our LI is more precise than Textcat.

Table 11: *Identification errors by BI sentence length (more than one language in the output).*

Length of sentences	Number of BI sentences	Textcat	Our LI
2	10	10	4
3	10	9	4
4	10	10	5
5	10	8	4
6	10	9	6
7	10	8	6
8	10	8	7
9	10	9	4
10	10	10	3
	90	81	43

Table 12: *Identification errors by sentence length (21 BI sentences, 12 BM sentences, 147 BI/BM sentences).*

Length of sentences	Textcat		Our LI	
	BM	BI	BM	BI
2	1	4	4	2
3	2	4	2	1
4	3	4	8	5
5	7	0	7	4
6	5	1	6	6
7	7	1	7	5
8	8	4	8	5
9	9	2	5	4
10	9	0	2	3
Total of errors	51	20	49	35

6. CONCLUSION

We have described in this paper the first language identifier that aims to guess close languages written in Roman alphabet. The method is simple and fast. The language identifier has been built to distinguish BM and BI texts but it can be applied for other close languages like American English and British English. The identification task is done in two steps. First, the BM or BI text is identified among other texts written in other languages. The use of trigrams provides acceptable results as it is shown in our results. Then, in the second step, the language of the unknown input text is identified clearly by applying two criteria: the presence of exclusive words and the format

of numbers. For the moment the two exclusive lists are not sufficient to cover all BM and BI texts. We have built manually the lists of exclusive words. We are currently looking for a method that can extract automatically these exclusive words from two aligned texts of close languages. There is no doubt that getting a perfect language identifier for close languages must contain two successive filters: a simple statistical method that is n-gram of characters, and a lookup to an exclusive list of words combined with some specific spelling rules (e.g. format of numbers, presence of diacritic characters, high frequency of some morphological endings, etc.).

ACKNOWLEDGMENTS

The author wishes to thank the anonymous reviewers for their valuable remarks and suggestions. The lists of sentences and outputs provided by the tested language identifiers presented in this paper are available on request. The author gratefully acknowledges Universiti Sains Malaysia through Research Creativity and Management Office for their financial support (USM Short Grant, 304/PKOMP/634121). The author thanks Ng Pek Kuan for implementing the language identifier in Java and making it available online (http://utmk.cs.usm.my:8080/identifier/Identifier.jsp).

References

[1] S. Langer. Natural languages and the World Wide Web, *Bulag, Revue annuelle*, Presses Universitaires Franc-Comtoises, S. pp. 89-100, 2001.

[2] *Web Languages Hit Parade*, http://alis.isoc.org/palmares.en.html. (07/11/2006)

[3] *SILC* (*Systeme dIdentification de la Langue et du Codage*), http://rali.iro.umontreal.ca/. (07/11/2006)

[4] S. Lewis, K. McGrath, J. Reuppel, Language Identification and Language Specific Letter-to-Sound Rules, *Colorado Research in Linguistics*, Vol. 17, Issue 1, 2004.

[5] O. Asmah, The Malay Language In Malaysia And Indonesia: From Lingua Franca To National Language,. *The Asianists ASIA*, vol. II, 2001.

[6] D. A. Aziz, The Development of the Malay Language: Contemporary Challenges, 2003, http://english.gdufs.edu.cn/ArticleShow.asp?ArticleID = 182.(07/11/2006)

[7] R. G. Jr. Gordon (ed.), *Ethnologue: Languages of the World*, Fifteenth edition, Dallas, Texas: SIL International, 2005, Online version: http://www.ethnologue.com/. (07/11/2006)

[8] E. Giguet, Categorization according to Language: A step toward combining Linguistic Knowledge and Statistic Learning, *Proceedings of the International Workshop of Parsing Technologies*, Sept. 1995, Prague - Karlovy Vary, Czech Republic.

[9] G. Grefenstette, Comparing two language identification schemes, *JADT 1995, 3rd International Conference on Statistical Analysis of Textual Data*, Dec. 1995, Rome, pp. 11-13.

[10] A. Poutsma, Applying Monte Carlo Techniques to Language Identification, *Language and Computer, Computational Linguistics in the Netherlands 2001, Selected Papers from the Twelfth CLIN Meeting*, Edited by M. Theune, A. Nijholt and H. Hondord, pp. 179-189, 2001.

[11] E. Giguet, Multilingual Sentence Categorization according to Language, *Proceedings of the European Chapter of the Association for Computational Linguistics SIGDAT Workshop From text to tags: Issues in multilingual Language Analysis*, March 1995, Dublin, Ireland, pp. 73-76.

[12] C. Souter, G. Churcher, J. Hayes, J. Hughes S. Johnson, Natural Language Identification using Corpus-Based Models, *Hermes Journal of Linguistics*, no. 13, pp. 183-203, 1994.

[13] S. Kulikowski, Stan, Using short words: a language identification algorithm, Unpublished Technical Report, 1991.

[14] M. Damashek, Gauging Similarity via N-Grams: Language-Independent Sorting, Categorization, and Retrieval of Text, *Science*, Vol. 267, pp. 843-848,1995.

[15] K. R. Beesley, Language Identifier: A computer program for automatic natural-language identification of on-line text, *Language at Crossroads: Proceedings of the 29th Annual Conference of the American Translators Association*, Oct. 1988, pp. 47-54.

[16] W. B. Cavnar, J. M. Trenkle, N-Gram-Based Text Categorization, Proceedings of SDAIR-94, *3rd Annual Symposium on Document Analysis and Information Retrieval*, April 1994, Las Vegas, pp. 161-175.

[17] M. Padr?, L. Padro, Comparing methods for language identification, *Proceedings of the XX Congress de la Sociedad Espanola para el Procesamiento del Lenguaje Natural (SEPLN)*, July 2004, Barcelona, Spain, pp. 155-161.

[18] J. M. Prager, Linguini: Language Identification for Multilingual Documents, *Proceedings of the 32nd Hawaii International Conference on System Sciences,* 1999, Hawaii, pp. 1-11.

[19] *Textcat Language Guesser*, http://odur.let.rug.nl/ vannoord/TextCat/Demo/. (07/11/2006)

[20] *Lextek Language Identifier SDK*, http://www.languageidentifier.com/. (07/11/2006)

Bali Ranaivo-Malancon was born in Madagascar. She received a Ph.D. in Natural Language Processing (NLP) from the National Institute for Oriental Languages and Civilisations (INALCO), Paris, France, in 2001. Since 2002, she has been with the School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia, where she is currently a Lecturer. Her research interests include the development of a generic language identifier for close languages and the creation of NLP tools for the analysis and annotation, of Malay and Indonesian texts.