The International Journal on Advances in ICT for Emerging Regions 2009 02 (02) : 21 - 28

Optimizing n-gram Order of an n-gram Based Language Identification Algorithm for 68 Written Languages

Chew Y. Choong, Yoshiki Mikami, C. A. Marasinghe and S. T. Nandasara

***Abstract*—Language identification technology is widely used in the domains of machine learning and text mining. Many researchers have achieved excellent results on a few selected European languages. However, the majority of African and Asian languages remain untested. The primary objective of this research is to evaluate the performance of our new n-gram based language identification algorithm on 68 written languages used in the European, African and Asian regions. The secondary objective is to evaluate how n-gram orders and a mix n-gram model affect the relative performance and accuracy of language identification. The n-gram based algorithm used in this paper does not depend on the n-gram frequency. Instead, the algorithm is based on a Boolean method to determine the output of matching target n-grams to training n-grams. The algorithm is designed to automatically detect the language, script and character encoding scheme of a written text. It is important to identify these three properties due to the reason that a language can be written in different types of scripts and encoded with different types of character encoding schemes. The experimental results show that in one test the algorithm achieved up to 99.59% correct identification rate on selected languages. The results also show that the performance of language identification can be improved by using a mix n-gram model of bigram and trigram. The mix n-gram model consumed less disk space and computing time, compared to a trigram model.**

***Index Terms*—Boolean Method, Character Encoding Scheme, Digital Language Divide, Language Identification, Mix n-gram Model, n-gram, Natural Language Processing, Language, Script.**

Manuscript received April 2, 2009. Accepted October 20th, 2009.

This work was sponsored by the Japan Science Technology Agency (JST) through the Language Observatory Project (LOP) and of the Japanese Ministry of Education, Culture, Sports, Science and Technology (MEXT) through the Asian Language Resource Network project.

Chew Y. Choong is with the Nagaoka University of Technology, Nagaoka, Niigata, Japan, (e-mail: yewchoong@gmail.com).

Yoshiki Mikami and C. A. Marasinghe are also with the Nagaoka University of Technology, Nagaoka, Niigata, Japan. (e-mail: mikami@kjs.nagaokaut.ac.jp, ashu@kjs.nagaokaut.ac.jp).

S. T. Nandasara is with the University of Colombo School of Computing, Colombo, Sri Lanka (e-mail: stn@ucsc.cmb.ac.lk).

I. Introduction

A. Digital Language Divide

Ethnologue [1] claims that there are 6,912 living languages in the world. However, ISO 639-2, the second part of the ISO 639 standard, only adopted 464 codes for the representation of the names of languages [2]. In 1999, worried about half of the world languages facing the risk of dying out, the United Nations Educational, Scientific and Cultural Organization (UNESCO) decided to launch and observe an International Mother Language Day on 21 February every year to honor all mother languages and promoting linguistic diversity [3]. The United Nation's effort in promoting mother languages was recognized by the Guinness World Record when its publication of Universal Declaration of Human Rights (UDHR) was declared the "Most Translated Document" in the world. UDHR is translated into 329 languages as of March 2009. On the Web, Google search engine allows users to refine their search based on one of the 45 languages it supports. As of November 2008, Microsoft's dominant (63.67%) Windows XP operating system was only released in 44 localized language versions. All these facts lead us to conclude that access to the digital world is greatly divided by language.

B. Measure Languages on the Internet

In order to bridge the digital language divide, UNESCO has been emphasizing the concept of multilingualism and participation for all the languages in the Internet. UNESCO, at its 2005 World Summit for the Information Society in Tunis, published a report entitled "Measuring Linguistic Diversity on the Internet", comprising articles on issues of the Language Diversity on the Internet. However, UNESCO admitted that the volume does not present any final answer on how to measure languages on the Internet [4].

The Language Observatory Project (LOP) launched in 2003 is to provide means for assessing the usage level of each language in the Internet. More specifically, the project is expected to produce a periodic statistical profile of language, script and character encoding scheme (LSE) usage in the Internet [5]. The LOP uses a language identifier to

automatically detect the LSE of a web page. The algorithm described in this paper is used to construct the language identifier for LOP.

C. Language Identification

Language identification generally refers to a process that attempts to classify a text in a language to one in a pre-defined set of known languages. It is a vital technique for Natural Language Processing (NLP), especially in manipulating and classifying text according to language. Many researchers [6] [7] [8] [9] [10] [11] [12] have achieved excellent results on language identification based on a few selected European languages. However, majority of African and Asian languages remain untested. This reflects the fact that search engines have very limited support in their language-specific search ability for most of the African and Asian languages.

In this paper, a language is identified by its LSE properties. All LSE properties are important for precise language categorization. For example, the script detection ability allows one to measure the number of web pages that are written in a particular script, for instance, the Sinhala script. Furthermore, LSE detection is critical to determine the correct tool for text processing at a later stage. Table I shows sample texts of Uzbek language written in three different types of scripts and character encoding schemes. A machine translation tool must at first get to know the script and character encoding scheme of the source text, in order to select the proper translator to translate the source text to another language.

TABLE I
EXAMPLE OF UZBEK LANGUAGE USING DIFFERENT SCRIPTS AND CHARACTER ENCODING SCHEMES

Language	Script	Character Encoding Scheme	Sample Text
Uzbek	Arabic	UTF-8	غ ف ق ك گ ڭ ل من ء
Uzbek	Cyrillic	Cyrillic	лмпрстўфх
Uzbek	Latin	ISO 8859-1	abchdefgg

D. N-gram

An n-gram can be viewed as a sub-sequence of N items from a longer sequence. The item mentioned can be refer to a letter, word, syllable or any logical data type that is defined by the application. Due to its simplicity in implementation and high accuracy on predicting the next possible sequence from known sequence, the n-gram probability model is one of the most popular methods in statistical NLP. The principal idea of using n-gram for language identification is that every language contains its own unique n-grams and tends to use certain n-grams more frequently than others, hence providing a clue about the language.

An n-gram order 1 (i.e. n=1) is referred to as a monogram; n-gram order 2 as a bigram and n-gram order 3 as a trigram. The rest is generally referred as "n-gram". Using "No-456" as an example, if we defined that the basic unit of desired n-gram as a "character", the valid lists of character level bigrams and trigrams (each separated by space) will be as below:

Bigram: No o- -4 45 56

Trigram: No- o-4 -45 456

Several researchers [6] [7] [8] [9] [10] reported that using trigram model on selected European languages produced the best language identification result. However, many African and Asian languages are not based on the Latin alphabet that many European languages employ. Thus, this study evaluates the performance of n-gram orders (n=1, 2 …6) and a special mix n-gram model for language identification on selected languages.

The rest of the paper is structured as follows. In the next section the authors briefly discuss related work. The n-gram based language identification algorithm is introduced in Section III. In Section IV, the authors explain about the datasets and experiments. Experimental results are presented and discussed in Section V. Section VI concludes the paper and mentions future work.

II. RELATED WORK

The task of identifying the language of a text had been relatively well studied over the past century. A variety of approaches and methods such as Dictionary method, Closed-class-model [11], Bayesian models [7], SVM [12] and n-gram [6] [7] [8] [9] [10] [13] [14] had been used. Two n-gram based algorithms are selected for detailed description. The Cavnar and Trenkle algorithm deserves special attention as it explains in-depth on how n-gram can be used for language identification. Suzuki algorithm which is implemented in Language Observatory Project is a benchmark to our algorithm.

A. Cavnar and Trenkle Algorithm

In 1994, Cavnar and Trenkle reported very high (92.9–99.8%) correct classification rate on Usenet newsgroup articles written in eight different languages using rank-order statistics on n-gram profiles [8]. They reported that their system was relatively insensitive to the length of the string to be classified. In their experiment, the shortest text they used for classifying was 300 bytes, while their training sets were on the order of 20 Kilobytes to 120 Kilobytes in length. They classified documents by calculating the distances of a test document's n-gram profile from all the training languages' n-gram profiles and then taking the language corresponding to the minimum distance. In order to perform the

distance measurement they had to sort the n-grams in both the training and test profiles.

B. Suzuki Algorithm

In Suzuki algorithm, the method is different from conventional n-gram based methods in the way that its threshold for any category is uniquely predetermined [9]. For every identification task on the target text, the method must be able to respond with either "correct answer" or "unable to detect". The authors used two predetermined values to decide the answer to a language identification task. The two predetermined values are UB (closer to the value 1) and LB (not close to the value 1), with a standard value of 0.95 and 0.92, respectively. The basic unit used in this algorithm is trigram. However, the authors refer to it as a 3-byte shift-codon.

In order to detect the correct language of a target text, the algorithm will generate a list of shift-codons from the target text. The target's shift-codons will then be compared with the list of shift-codons in training texts. If one of the matching rates is greater than UB, while the rest is less than LB, the algorithm will report that a "correct answer" has been found. The language of the training text with matching rate greater than UB is assumed to be the language of the target text. By this method, the algorithm correctly identified all test data of English, German, Portuguese and Romanian languages. However, it failed to correctly identify the Spanish test data.

III. Methodology

The overall system flow of the language identification process is shown in Fig. 1. In this process, a set of training profiles is constructed by converting training texts, in various language, script and character encoding scheme (LSE) into n-grams. The generated training profile contains a set of distinct n-grams, without frequency of occurrence of n-grams. In the same way, the system converted the target text into target profile. The system then measured the matching rates of n-gram between the target profile and the training profiles. The system classifies the target profile belonging to the LSE of the training profile that yields the highest matching rate.

A. The Matching Mechanism

The process in Fig. 1 labeled "Measure matching rates between target profile and all training profiles" is used to calculate the matching rates between a target profile and all training profiles. Unlike many other n-gram based algorithms, our algorithm does not depend on n-gram frequency. Instead, the algorithm uses a Boolean method to decide the output of the matching. The Boolean method returnsvalue of 1 if the n-gram from the target profile is found among the training profiles. The Boolean method returns value of 0 if there is no match. After all n-grams in the target profile have been compared to those in training profile, the system derives the matching rate by dividing the total match values to total number of distinct n-grams in the target profile. (see equiation (1))

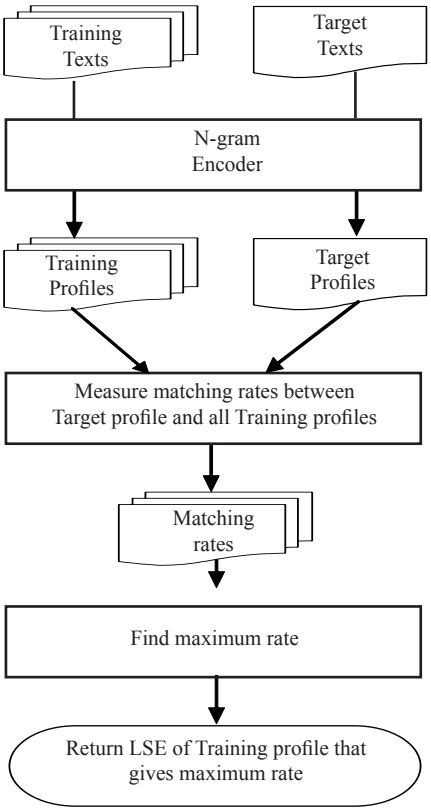

Fig. 1. System flow of the language identification process.

$$R_i = \sum_{i=1}^{n} \frac{m_i}{n} \quad (1)$$

where,

$$m_i = \begin{cases} 0 & \text{if } t_i \text{ did not match with } T_j \\ 1 & \text{if } t_i \text{ matched with } T_j \end{cases}$$

The matching mechanism can be simplified as in the following steps:

- Let us define the target profile as t and the number of distinct n-grams in t as n. Hence, the list of n-grams in t is $t_1, t_2, t_3 \ldots t_n$;
- Similarly, we define the training profile as T and the number of distinct n-grams as k. The list of n-grams in T is $T_1, T_2, T_3 \ldots T_k$;
- R (or R-value) is calculated for every distinct n-gram in the target profile using equation (1), where R_i is the rate at which the ith distinct n-gram in the target profile (t) matches with the jth distinct n-gram in the training profile (T);

B. The Base Unit of the n-gram

In this algorithm, the basic unit of the n-gram is of data type "byte". The reason "byte" is selected instead of character or word is to avoid possible character encoding errors due to unexpected conversion occurring when reading a text file encoded in an abnormal encoding scheme. For example, a text file created with a non-standard legacy font.

IV. Data Sets And Experiments

There are two data sets used in the experiments. The first data set contained all the training texts that are encoded in various language, script and character encoding scheme (LSE). From here onward we refer to this set as the training corpus. The second data set is a collection of text documents that the authors used as target texts in the experiments. From here onward we refer to it as the validation corpus.

The training corpus is mainly based on the Universal Declaration of Human Rights (UDHR) texts collected from the Official United Nation's Universal Declaration of Human Rights web site. At the time of the experiment, the training corpus contained 571 UDHR text documents in various types of LSE. The total size of the training corpus is 10,513,237 bytes. The document sizes ranged from 3,977 to 126,219 bytes.

The validation corpus was mainly based on web pages that the authors collected from online newspapers and media web sites. The six major online newspapers and media service providers used to construct the validation corpus were BBC news in 32 languages, Voice of America news in 45 languages, Wikinews in 25 languages, Google news for 62 countries, Deutsche Welle news in 30 languages and China Radio International in 45 languages. In addition, the authors referred to online news portals such as "ABYZ News Links", "World Newspapers and Magazines" and "Thousands of newspapers on the Net" to locate a wider range of local news in many Asian and African countries. A total of 730 web pages were collected, spanning 68 languages and with a total size of 32,088,064 bytes. The document sizes ranged from 313 to 437,040 bytes. We did not normalize the size of those documents in order to mimic the situation on the Web.

TABLE II
Language In Validation Corpus, Grouped By Region

Africa (9 languages)	Asia (27 languages)	Europe (32 languages)
Afrikaans, Amharic, Hausa, Ndebele, Rundi, Rwanda, Shona, Somali, Swahili	Abkhaz*, Aceh, Arabic, Armenian*, Azerbaijani, Burmese, Chinese, Dari, Farsi, Georgian*, Hebrew, Hindi, Indonesian, Japanese, Korean, Kurdish, Malay, Nepali, Panjabi, Pashto, Russian*, Tamil, Thai, Turkish*, Urdu, Uzbek, Vietnamese	Abkhaz*, Albanian, Armenian*, Bosnian, Bulgarian, Catalan, Czech, Danish, Dutch, English, Estonian, Finnish, French, Georgian*, German, Greek, Hungarian, Italian, Latvian, Lithuanian, Norwegian, Polish, Portuguese, Romanian, Russian*, Serbian, Slovak, Slovenian, Spanish, Swedish, Turkish*, Ukrainian

Table II shows the list of languages used in validation corpus, grouped according to present world's region. The asterisk (*) next to a language name indicates that the language is spread across multiple regions. For example, the language Abkhaz is mainly used in the Caucasus area, which is a geopolitical region located between Europe, Asia, and the Middle East.

Experiment 1: To evaluate the correct identification rate of the algorithm based on different n-gram orders

The first experiment was designed to evaluate the correct identification rate of the algorithm based on different n-gram order. In total, six language identification tests were carried out based on n-gram order 1 to 6. N-gram orders greater than 6 are not considered as they consumes too much processing power and time. For each n-gram order within the range, every text document in training and validation corpus was converted into n-gram profile. After that, the system calculated the matching rate between the target profile and every training profile. The matching rate is determined by the Boolean method described in the "The Matching Mechanism" section. After all matching rates have been determined, the system reported the language, script and character encoding scheme (LSE) of the target profile, derived from the LSE of the training profile that returned the highest matching rate.

Experiment 2: To evaluate the efficiency of the algorithm based on mix n-gram model

The second experiment was designed to evaluate how mix n-gram model affects the language identification result. In this experiment, each training text was trained into training profile using the optimized n-gram order discovered in the first experiment. The authors defined the optimized n-gram order for each LSE as the smallest *N* that gave the most correct answers. In this model, the n-gram order used to convert the target text is dynamically alterd by the system, depending on the n-gram order of the current training profile. If current training profile is trained with *N*=2, the target text will be converted to n-gram using *N*=2.

In the authors' first attempt, 12 languages, namely Armenian, Azerbaijani, Chinese, Czech, Hungarian, Indonesian, Japanese, Korean, Panjabi, Pashto, Rwanda and Slovak were trained using n-gram order 2. The rest of the training texts were trained using n-gram order 3.

In the authors' second attempt, 5 languages, namely Armenian, Chinese, Japanese, Korean and Panjabi were trained using n-gram order 2. The rest of the training texts were trained using n-gram order 3.

V. Results And Discussions

Result for Experiment 1

The objective of the first experiment is to evaluate the accuracy of our algorithm on selected 63 written languages, using n-gram orders 1 to 6.

In Fig. 2, the correct identification rate of language identification (y-axis), along the n-gram order (x-axis) is shown. By using n-gram order 1, the correct identification rate is very low, only 6.99%. When n-gram order increased to 2, the correct identification rate increased to 56.30%. The algorithm achieved its best correct identification rate at 99.59%, when n-gram order is 3. Beyond n-gram order 3, the system gains no improvement on identification result. Instead, the algorithm only achieved correct identification rate of 96.44%, 94.66% and 93.01% for n-gram order 4, 5 and 6, respectively.

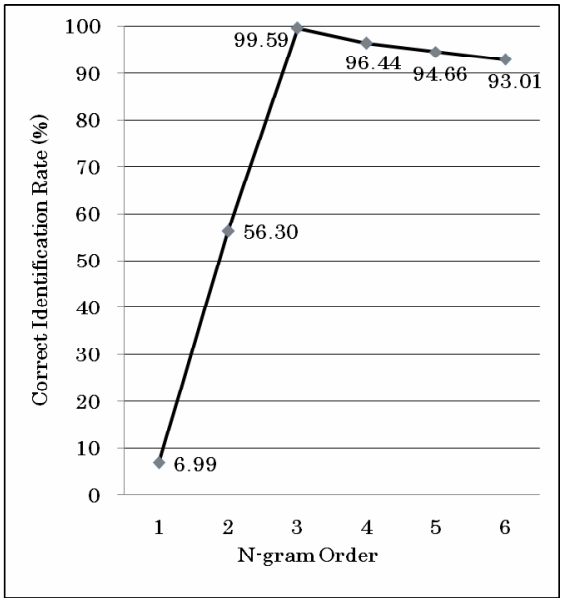

Fig. 2. Correct identification rate based on n gram order 1 to 6.

TABLE III
Language Identification Errors On Trigram Model

Language	Number of Distinct n-gram	Identified As
Danish	66	Norwegian
Dari	829	Farsi
Malay	328	Indonesian

Using trigram model, 3 out of 730 target profiles in validation corpus were incorrectly identified by the algorithm. Table III shows that the three target profiles, namely Danish, Dari and Malay had been identified as another language that is very close to their language family. Danish and Norwegian both belong to the North Germanic languages (also called Scandinavian languages), a sub-group of the Germanic branch of the Indo-European languages. Dari and Farsi are practically the Persian language, where Dari is the local variant of the language spoken in Afghanistan, while Farsi is the local variant of the language used in modern day Iran. In the case of Malay and Indonesian, they share the same language family in Austronesian, a language family that is widely dispersed throughout the islands of Southeast Asia and the Pacific. Besides, it should be noted that the number of distinct n-grams in the Danish target profile is as low as 66.

Result of experiment 1 also showed that for 12 languages we were able to correctly identify all their target profiles using n-gram order 2. Table IV lists the language, script and character encoding scheme (LSE) of the 12 languages.

Result for Experiment 2

In the first test of experiment 2, the authors trained the training texts of the 12 languages using n-gram order 2, while the rest of training texts were trained using n-gram order 3. Unfortunately, the first attempt on using mix n-gram model returned very bad result. The overall correct identification rate for all target profiles was reduced to 46.30%. The authors manually went through every record of the identification results and discovered that, after trained with n-gram order 2, the Azerbaijani, Czech, Hungarian, Indonesian, Rwanda and Slovak's training profiles caused a lot of missed identification errors to other LSE's target profiles.

Hence, the authors learned that languages based on Arabic and Latin scripts are not suitable with n-gram order 2 when they are not tested alone.

In the second test of experiment 2, only Armenian, Chinese, Japanese, Korean and Panjabi were trained with n-gram order 2, while the rest were trained with n-gram order 3. The language identification result of this mix n-gram model was excellent, achieving an overall correct identification rate of 99.59%.

TABLE IV
LSES THAT ACHIEVED HIGHEST CORRECT INDENTIFICATION RATE ON THEIR TARGET PROFILES USING N-GRM ORDER 2

Language	Script	Encoding	Correct Identification Rate (%)
Armenian	Armenian	UTF8	100
Azerbaijani	Latin	UTF8	100
Chinese	Traditional, Simplified	Big5, GB2312, UTF8	100
Czech	Latin	Latin, UTF8	100
Hungarian	Latin	Latin	100
Indonesian	Latin	Latin	100
Japanese	Japanese	EUC, JIS, SJIS, UTF8	100
Korean	Korean	EUC-KR	100
Panjabi	Gurmukhī	UTF8	100
Pashto	Naskh (Arabic)	UTF8	100
Rwanda	Latin	UTF8	100
Slovak	Latin	Latin, UTF8	100

Although the overall correct identification rate is the same as using trigram model, this model achieved better performance in two categories: (1) processing time and (2) disk space.

Using n-gram order 2 resulted in lower training and identification time. Fig. 3 shows the total computing time needed for language identification task based on n-gram order 1 to 6. The time grows enormously when *N* increased. For simplicity, we defined *N*=2.5 to represent the mix n-gram model of bigram and trigram. In this optimized condition, the language identification task needs only 651,078 milliseconds to complete, compared to 817,844 milliseconds on trigram model. The mix n-gram model is able to save up to 20.39% of total computing time.

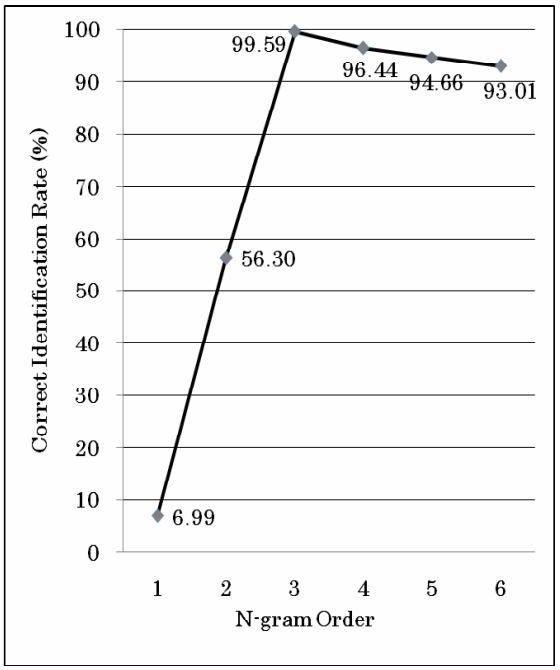

Fig. 3. Exponential growth of computing time on language identification task.

The mix n-gram model also consumes less disk space. Its total size of training profiles is 7,652 Kilobytes. The total size of training profiles using n-gram order 3 is 8,076 Kilobytes. The mix n-gram model requires 5.25% less in disk space.

Table V shows the comparison of our algorithm and several other algorithms in the literature based on language coverage, n-gram order and overall correct identification rate of language identification. Our algorithm stands out in terms of language coverage and the mixture of n-gram order.

TABLE V
COMPARISON BETWEEN N-GRM BASED LANGUAGE IDENTIFICATION ALGORITHMS BASED ON LANGUAGE COVERAGE, N-GRM ORDER AND CORRECT IDENTIFICATION RATE

Algori-thm	Language Coverage	n gram order	Correct Identi-fication Rate (%)
Dunning T. [7]	2 languages (Dutch, Pol-ish)	2 3	92 99.9
Cavnar and Trenkle [8]	8 languages (English, Portuguese, French, Ger-man, Italian, Spanish, Dutch, Polish)	3	92.9–99.8
Suzuki [9]	5 languages (Portuguese, Spanish, Romanian, German, English)	3	No precise fig-ure. Problem on Spanish
Ölvecký [10]	3 languages (Czech, Slo-vak, Polish)	3	95–99.2
This paper's algorithm	68 languages, as listed in Table II	Mixture of 2 and 3	99.59

VI. CONCLUSION AND FUTURE WORKS

In this paper, we reported the n gram based language identification algorithm and the experiments carried out to evaluate its accuracy against 68 languages used in African, Asian and European regions. We show that the algorithm is highly efficient in classifying written text. The algorithm is unique as the matching mechanism does not depend on n gram frequency. The algorithm depends on a Boolean method to determine the output of matching target n grams and training n grams. Like many previous studies done by n gram methods, n gram order 3 generated the best language identification result in the experiments. However, we discovered that the performance of five Asian languages, namely Armenian, Chinese, Japanese, Korean and Panjabi, improved by using n gram order 2. An experiment based on a mix n gram model of bigram and trigram confirmed the effectiveness of mixing n gram order. The total computing time consumed by language identification task in experiment 2 was reduced by one-fifth while maintaining the same correct identification result.

Although the current research has demonstrated good performance, the authors believe there is still room for improvement:

- Currently the validation corpus contains text documents in 68 languages. This number is relatively small if compard to the 571 languages collected in the training corpus. How well the algorithm can scale from the current corpus to a bigger size corpus remains unknown. To confirm the true ability of the algorithm, we need to evaluate it against a larger validation corpus.
- The algorithm made three errors in language identification using the validation corpus. In all cases, the target text was identified as a language that is close to its language family. What are the best strategies to correctly identify languages that are close to each other? The authors need to find a solution for this critical issue.
- Table III showed that the numbers of distinct n-grams for the wrongly identified Danish, Dari and Malay target profiles are quite low. Danish's profile in particular contains only 66 distinct n-grams. This brings up the question of what minimum size of n-grams is needed in order to correctly identify a language. How does the number vary among different languages, scripts and character encoding schemes? Such a study is currently underway.
- A related issue is how the quality of training text in general affects language identification result. Although Universal Declaration of Human Rights is the most frequently translated document, other sources for training text could be considered in order to improve the identification result.

ACKNOWLEDGMENT

The authors are grateful for the sponsorship of the Japan Science Technology Agency (JST) through the Language Observatory Project (LOP) and of the Japanese Ministry of Education, Culture, Sports, Science and Technology (MEXT) through the Asian Language Resource Network project, which provided the entire corpus data used in this research.

REFERENCES

[1] R. G. Gordon , *Ethnologue: Languages of the World* (15th ed.). SIL International, Dallas, 2005, ISBN: 155671159X.
[2] J.W. Group, *Codes for the representation of names of languages: alpha-2 codes*. Technical Report, 1998, ISTC46/SC4 and ISO TC37/SC2.
[3] UNESCO, *Records of the general conference 30th session*. United Nations Educational, Scientific and Cultural Organization, 1999
[4] J. Paolillo, D. Pimienta, D. Prado, *Measuring Linguistic Diversity on the Internet*. United Nations Educational, Scientific and Cultural Organization, 2005.
[5] Y. Mikami, P. Zavarsky, M. Z. Rozan, I. Suzuki, *The Language Observatory Project (LOP)*. In Poster Proceedings of the Fourteenth International World Wide

Web Conference, 2005, (WWW2005). pp. 990-991, May 2005, Chiba, Japan.

[5] P. F. Brown, R.L. Mercer, J. C. Lai, Class-based n-gram models of natural language. *Computational Linguistics*, 1992, Vol.18, pp. 18–4.

[6] T. Dunning, *Statistical Identification of Language*. Technical Report, New Mexico State University, 1994, MCCS 94-273.

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

[8] I. Suzuki, Y. Mikami, A. Ohsato, A Language and Character Set Determination Method Based on N-gram Statistics. *ACM Transaction on Asian Language Information Processing*, 2002 Vol. 1. No. 3, pp. 270–279.

[9] T. Ölvecký, *N-Gram Based Statistics Aimed at Language Identification*. In Mária Bieliková (Ed.), Proceedings of IIT.SRC 2005: Student Research Conference in Informatics and Information Technologies, Bratislava, 2005, pp. 1–7.

[10] E. M. Gold, Language identification in the limit. *Information and Control*, 1967, Vol.10, no.5, pp. 447–474.

[11] D. Chen and H. Bourlard, *Text Identification in Complex Background using SVM*. Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, 2001, pp. 621–626.

[12] Bruno Martins and Mário J. Silva, *Language Identification in Web Pages.* Symposium on Applied Computing, Proceedings of the 2005 ACM Symposium on Applied Computing, 2005, pp. 764-768.

[13] Grefenstette G., *Comparing two language identification schemes*. In 3rd International Conference on Statistical Analysis of Textual Data, Rome, Italy, 1995.