Language Identification of Kannada Language using N-Gram

Deepamala. N
Assistant Professor
Dept. of Computer Science
R.V. College of Engineering
Bangalore, India

Ramakanth Kumar. P
Professor and Head
Dept. of Information Science
R.V. College of Engineering
Bangalore, India

ABSTRACT
Language identification is an important pre-processing step for any Natural Language Processing task. Kannada Language is an Indian Language and lot of research is being carried out on Kannada Language Processing. Major parts of online documents like websites are combination of Kannada and English Sentences. Language Identification is a preprocessing step for NLP tasks like POS tagging, Sentence Boundary Detection or Data mining technique. In this paper, we present an n-gram method of language identification for documents with Kannada, Telugu and English sentences. It has been shown how performance can be improved by n-gram processing only last word of the sentence instead of complete sentence. This method could also be preprocessing step for Sentence Boundary Detection discussed in [1].

General Terms
Language Identification, Kannada Language.

Keywords
n-gram processing, verb suffix, Langauge Identification.

1. INTRODUCTION
Kannada is one of the 40 most spoken Languages in the world with majority of the population in Karnataka, India. Most of the online documents available are combination of Kannada and English Language. Hence, before applying any Natural Language processing technique, the language of the sentence has to be identified. There are different methods of Language Identification techniques applied and tested for different languages. In this paper, we present the Language identification of Kannada Language using N-gram processing and show how performance can be improved by using only last word instead of complete sentence.

2. LITERATURE SURVEY

2.1 Language Identification
Language identification problem has been solved using many techniques for different languages. For Indian Languages, many researchers have followed different techniques to achieve script identification. Most of the research is in parsing the document image to identify the script. Padma and Vijaya [2, 3] have achieved language identification and script identification using OCR technique. Mallikarjun et al. [4] present word level script identification using global and local features. Scripts are grouped into different classes and water reservoir principle, contour tracing, etc are used for identification [5]. Shantanu et al. use Gabor filters combined with pixel distribution around connected components for identification of script [8].

2.2 Language Identification using N-gram
N-gram method of language identification is the popular and reliable method of language identification. Cavnar and Trenkle [6] and Dunning [7] have used n-gram processing of text to identify the language. Lena et al. [8] compares three different algorithms (short words, frequent words and n-gram) for language identification. Yew Choong et al. [9] tried to identify the language of the web pages using n-gram processing method. Muntsa et al. [10] compared 3 method of language identification (Markov Models, Trigram frequency vectors and n-gram based text categorization). Shiho et al. use n-gram statistical analysis to identify person names [11]. Tommi et al. compare 2 distinct methods Naïve Bayes and N-gram models to identify language of short text [12]. [13] explains how language can be identified using cumulative frequency addition method.

3. PRESENT WORK
In this paper, the n-gram technique discussed by Cavnar and Trenkle [6] is followed. The libtextcat [14] software which is Language Identification tool coded in C is used for the experiments and results. The libtextcat code has been modified to use wchat_t data type(wide characters) instead of char and string handling functions like wcslen, wcscat, wcsrchr etc are used. In Kannada Language, characters are multibyte characters and handling them is simple using wide characters. It has been observed that wide characters are assigned 4 bytes in Linux. It is system dependent but since our implementation is completely in Linux, wchar can be used.

3.1 N-gram processing
The N-gram processing of text as proposed by Cavnar and Trenkle [6] involves splitting the word into characters of size n. Where n is bi-gram, tri-gram, quad-gram etc. For example, the word “PAPER” is decomposed as

Bi-grams: _P, PA, AP, PE, ER, R_

Tri-grams: _PA, PAP, APE, PER, ER_, R__

3.1.1 Train Profile generation
N-gram frequency profiles for different languages are generated using the steps as mentioned in [6]. In this paper, we refer to the profiles as Train profile and generation steps are as in Fig 1.

Different languages will have their own Train profiles. The train corpus is sent as parameter to the Train profile generation code. It has following steps:

- Split the text into separate tokens consisting only of letters and apostrophes. Digit and punctuation are discarded. Pad the token with sufficient blanks before and after.

- Scan down each token, generating all possible N-grams, for N=1 to 5. Use positions that span the padding blanks, as well.
- Hash into a table to find the counter for the N-gram, and increment it. The hash table uses a conventional collision handling mechanism to ensure that each N-gram gets its own counter.
- When done, output all N-grams and their counts.
- Sort those counts into reverse order by the number of occurrences. Keep just the N-grams themselves, which are now in reverse order of frequency.

First 300 ranks, which are mostly the common occurrence of character combinations for a particular language, are used.

3.1.2 Text Language Identification

A n-gram Test profile is generated for the Language which needs to be identified and is compared with the list of Train profiles generated for different languages. For each n-gram in Test profile, the rank at which it matches with the n-gram in the Train profile is found. The rank is the out-of-place value for the n-gram from Test profile; this is continued for all the n-grams in Test profile. The sum of out-of-place values of the Test profile when compared with a Train profile is recorded which is called the distance measure. The previous steps are repeated for different Train profiles of different languages. The least distance measure value among the recorded distance measure and its corresponding Train profile's language is the most probable language of the Text.

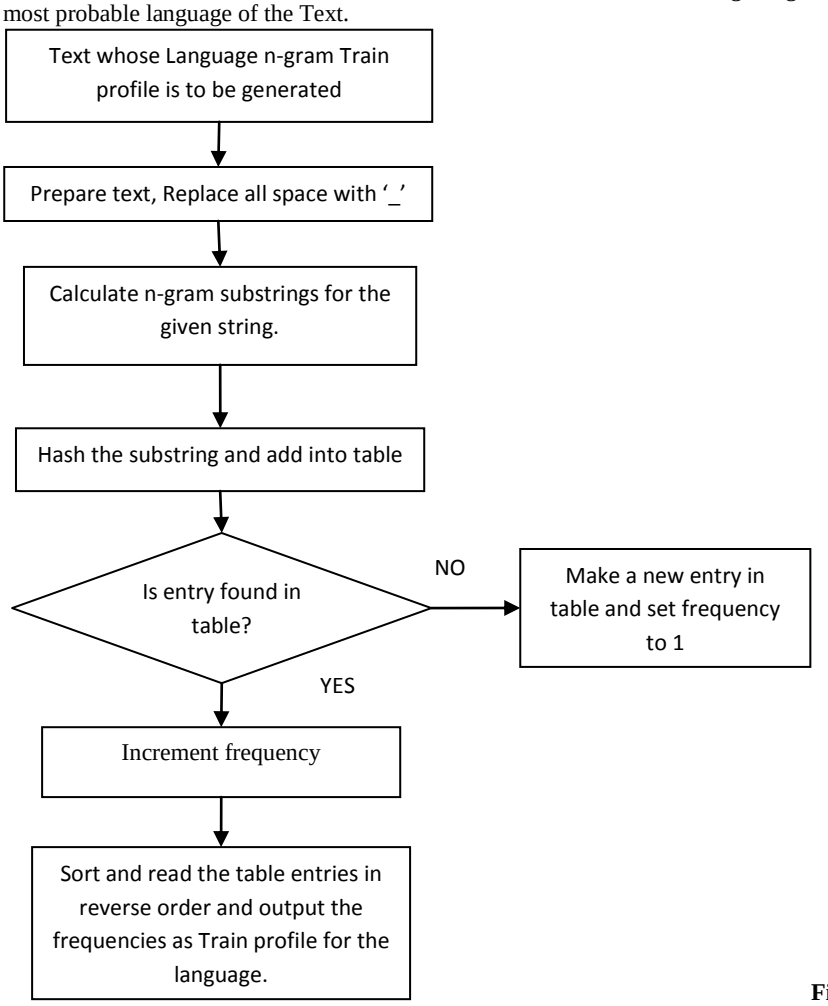

Fig 1: Train Profile generation steps

4. PR OPOSED WORK

The proposed implementation can be used to identify the language of a sentence in a digital online or offline document. This could be a pre-processing step for tasks like POS tagging, Content analysis or some other Natural Language Processing. The implementation is tested mainly on a document which has sentences of Kannada, Telugu and English.

4.1 Train Profile generation

The Train profile is generated using only the last word of the sentences. This reduces the processing time required to generate the Train profile as the training corpus size reduces and hence, n-gram processing is reduced. N-gram processing involves steps to replace space with "_", calculate substrings and create n-gram table. The idea behind including only the last word, in both Telugu and Kannada Language is the sentence format for majority of sentences. It is as follows:

Eg: ರಾಮನು(Noun) ಕಾಡಿಗೆ(Object) ಹೋದನು(Verb).

The verb takes different form as discussed in [1] based on tense, gender and Singular/Plural. Most of the sentences should have some form of Verb in the sentence's last word and hence comparing only the last words should be sufficient to identify the sentence. The N in N-gram is assumed to be three or trigram. Since the Corpus size is small because only last words from sentences are chosen, it has been found that tri-gram gives better performance.

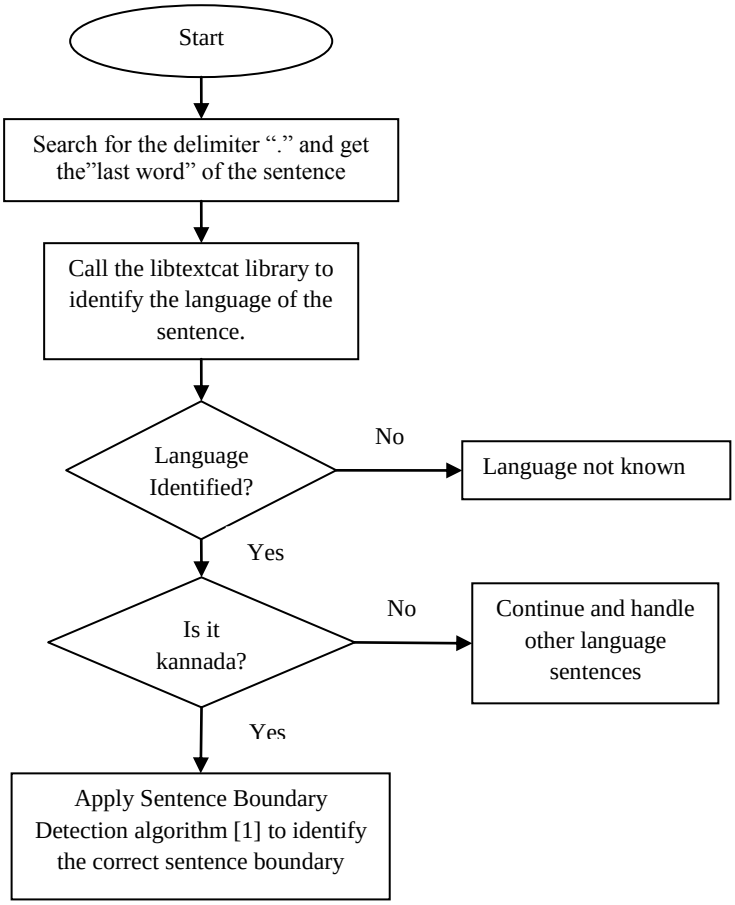

Fig 2: Language Identification and SBD

4.2 Language Identification:

The Language Identification has been tested on document containing Kannada, Telugu and English Sentences. Telugu has been included just to show the accuracy of the implementation as the sentence format for both Telugu and Kannada is almost same. For Identification, last word of the sentences are extracted and sent as parameter to libtextcat library to identify the Language.

4.3 Performance Measurement

The time taken to trigram process the language using complete sentence and only the last word of the sentence is measured. The performance is determined by comparing the values.

The time is measured as follows:

```
#include <sys/time.h>
struct timeval *Tps, *Tpf;
void *Tzp;
Tps = (struct timeval*) malloc(sizeof(struct timeval));
Tpf = (struct timeval*) malloc(sizeof(struct timeval));
Tzp = 0;
gettimeofday (Tps, Tzp);
        <code to be timed>
gettimeofday (Tpf, Tzp);
printf("Total Time (usec): %ld\n", (Tpf->tv_sec-Tps->tv_sec)*1000000+ Tpf->tv_usec-Tps->tv_usec);
```

4.4 Sentence Boundary Detection

After the Language has been identified, if the language of the sentence is Kannada, the algorithm [1] is used to identify if it is a Sentence Boundary. The Sentence Boundary algorithm requires the last word of the sentence before ".". The last word of sentence is compared with the contents of ABBREVIATIONS file, to check it is a abbreviation, if not, then VERBS_SUFFIX file is verified to check if it matches with any verb suffixes. If matched, then it is considered to be a sentence boundary or the control is shifted to last word of the next sentence.

The Language Identification procedure in this paper also shows that only last word of the sentence is sufficient to identify the language. So, both [1] and current procedure can work efficiently together, as the same last word used to identify the language can be used to identify if it is a Sentence Boundary. The steps for Language Identification and Sentence Boundary Detection are as shown in Fig 2.

5. TESTING AND RESULT

The testing has been performed on corpus of Kannada only sentences, Telugu only sentences and combination of Kannada, Telugu and English sentences. The Train profile for Kannada and Telugu has been created using only the last words of the sentences. Since the train data size decreases, the n-gram processing time to create the train profile also reduces. The result of testing using different combination of languages using complete sentence or only last word is given below in Table 1. The graph plotted using the processing time for

complete sentence and only last word is shown. The graph is plotted for 341Kbytes if data and corresponding processing time as shown in Table1. The processing time shown in the Table 1 is complete processing time including string handling functions and n-gram processing. The performance analysis of only n-gram processing of complete sentence and last word only is shown in Table 2. This excludes the string operation to fetch last word.

Table 1. Processing time for complete sentences and last word of sentence.

Test Corpus Language	Type	Size of buffer	Average Processing Time (in usec)
Kannada only sentences	n-gram processing of complete sentences	217K	956863
Telugu only sentences	n-gram processing of complete sentences	214K	931643
Kannada, Telugu and English Sentences	n-gram processing of complete sentences	341K	1965652
Kannada only sentences	n-gram processing of only last word of sentence	217K	851347
Telugu only sentences	n-gram processing of only last word of sentence	214K	824548
Kannada, Telugu and English Sentences	n-gram processing of only last word of sentence	341K	1742085

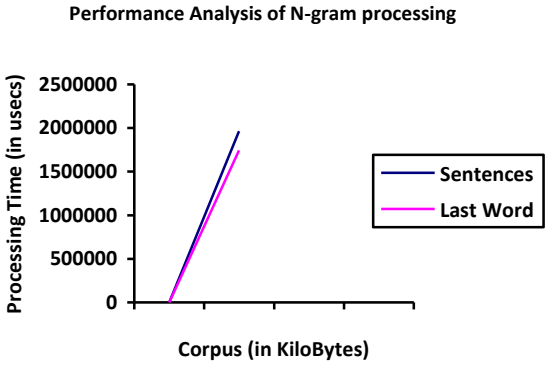

Table 2. Processing time for n-gram processing of complete sentence and last word of sentence.

Sentence	Time (in usec)
ವರ್ಷಗಳ ಹಿಂದೆ ಅಥೆನ್ಸಿನ ಸಮೀಪ ಆಲಿವ್ ಮರಗಳ ಒಂದು ತೋಪು ಇದ್ದಿತು. (complete sentence)	451
ಇದ್ದಿತು. (only last word)	251
ಅದು ಪ್ರಾಚೀನ ವೀರ ಅಕಡೆಮೆಸನ ಸ್ಮಾರಕಕ್ಷೇತ್ರವಾಗಿತ್ತು. (complete sentence)	333
ಸ್ಮಾರಕಕ್ಷೇತ್ರವಾಗಿತ್ತು. (last word)	244

6. CONCLUSION

This paper shows how performance can be improved by identifying the language by applying n-gram processing on last word of sentences for Kannada and Telugu Language. The results are encouraging as 99% of the sentences were identified correctly. The Language Identification technique described above can also be applied as pre-processing step for Sentence Boundary Detection for Kannada Language [1]. Since both require last word of the sentence, much of the processing with respect to string handling is reduced.

7. REFERENCES

[1] Deepamala.N and Ramakanth Kumar.P, "Sentence Boundary Detection in Kannada Language." International Journal of Computer Applications (0975 – 8887) Volume 39– No.9, February 2012.

[2] M.C. Padma, P.A. Vijaya, "Global Approach for Sript Identification using Wavelet Packet based Features", International Journal of Signal Processing, Image Processing and Pattern Recognition, Vol. 3, No. 3, September, 2010.

[3] M.C. Padma, P.A. Vijaya, "Script Identification from Trilingual Documents using Profile based Features", International Journal of Computer Science and Applications, Technomathematics Research Foundation Vol. 7 No. 4, pp. 16 - 33, 2010.

[4] Mallikarjun Hangarge , B.V. Dhandra, "Offline Handwritten Script Identification in Document Images"

International Journal of Computer Applications (0975 – 8887) Volume 4 – No.6, July 2010

[5] U.Pal and B.B.Chaudhuri, "Multi-Script Line identification from Indian Documents," *7th ICDAR*, 2003.

[6] W. B. Cavnar and J. M. Trenkle. "N-gram-based text categorization". In *Proceedings of SDAIR-94, the 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161.175, Las Vegas, Nevada, U.S.A, 1994.

[7] Ted Dunning. 1994. "Statistical identification of language". Technical Report MCCS-94-273, Computing Research Lab, New Mexico State University.

[8] Lena Grothe, Ernesto William De Luca, Andreas Nürnberger. "A Comparative Study on Language Identification Methods." Proceedings of the Sixth International Language Resources and Evaluation (LREC'08). Marrakech, 2008. 980-985.

[9] Yew Choong Chew, Yoshiki Mikami, Robin Lee. " Language Identification of Web Pages Based on Improved N-gram Algorithm." IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 1, May 2011

[10] M. Padro and L. Padro, "Comparing methods for language identication," Proceedings of the XX Congreso de la Sociedad Espanola para el Procesamientodel Lenguage Natural, Barcelona, Spain, 2004.

[11] Shiho Nobesawa and Ikuo Tahara, "Language Identification for Person Names Based on Statistical Information." Proceedings of PACLIC 19, the 19th Asia-Pacific Conference on Language, Information and Computation.

[12] Vatanen, Tommi and Väyrynen, Jaakko J. and Virpioja, Sami. "Language Identification of Short Text Segments with N-gram Models." European Language Resources Association, 2010

[13] B. Ahmed, S. Cha, "Language Identification from Text Using N-gram Based Cumulative Frequency Addition", Proceedings of CSIS 2004, Pace University, May 7th, 2004

[14] http://software.wise-guys.nl/libtextcat/