ISSN No. 2278-3083
Volume 1, No.4, September – October 2012
International Journal of Science and Applied Information Technology
Available Online at http://warse.org/pdfs/ijsait03142012.pdf

Natural Language Identification and Translation Tool for Natural Language Processing

Dr.M.Hanumathappa[1], Mallamma.V. Reddy[2]
[1, 2] Department of Computer Science and Applications, Jnanabharathi Campus,
Bangalore University, Bangalore-56, India
hanu6527@bub.ernet.in, mallamma_vreddy@bub.ernet.in

ABSTRACT

The language identification problem refers to the task of deciding in which natural language a given text is written this is the one of the major challenge in the Natural Language Processing. Once the language being identified next translation is to be carried out. Translation is the communication of the meaning of a source-language text by means of an equivalent target-language text. Whereas interpreting undoubtedly antedates writing, translation began only after the appearance of written literature. In this paper we are presenting the modules for identification and translation of English to Indian languages or Indian Interlingual Languages here particularly for English, Kannada and Telugu. We have analyzed the Morphology of each language for translating a text from source language to target language. To test these modules we have build our own morphological analyzer and generator, part of speech tagger, and designed virtual keyboards for Transliteration and Translation with the help of BUBShabdasagar-2011 Bilingual dictionary which consist around 20,000 words.

Keywords : Cross Language Information Retrieval (CLIR), Part of Speech Tagger (POS), Parsing, Morphology ,Natural Language Processing (NLP), Stemming **.**

1. INTRODUCTION

The rapid growth of the lesser-known languages in the Internet has created a need of language identification for applications like multilingual information retrieval, machine translation, spell checking etc. This task is complicated by three factors viz., the varying sizes of the character sets used to encode different languages, the usage of a variety of character sets for a single language and the same script being shared by more than a language. Automatic treatments of these texts, for any purpose requiring Natural Language processing. Such as WWW indexing and interrogation or providing reading aids, necessitates a preliminary identification of the language used. For example, morphological based stemming has proven important in improving information retrieval and applying language specific algorithms implies knowing the language used. Likewise, any system that involves dictionary access must identify language to perform language-specific lemmatization.

Data Mining is about looking patterns in data. Likewise Text Mining is about looking for patterns in text; it is the process of analysing text to extract information that is useful for particular purpose. One of the important challenge of Text mining [1] is Document classification is a supervised learning, in which each instance represents a document and the instance's class is the document's type. Documents are characterized by the words that appear in them. The presence or absence of the word or character is treated as a Boolean attribute, or documents can be treated as bags of words, rather than sets, by taking word frequencies into account. In this paper we are using the concept of document classification for Natural Language Identification namely English, Kannada and Telugu.

Kannada or Canarese is one of the 1652 mother tongues spoken in India. Forty three million people use it as their mother tongue. Kannada has 44 speech sounds. Among them 35 are consonants and 9 are vowels. The vowels are further classified into short vowels, long vowels and diphthongs. It is also one of the 18 Scheduled Languages included in the VIII Schedule of the Constitution of India is recognized as the Official and Administrative language of the state of Karnataka [2]. It belongs to the Dravidian family of languages. Within Dravidian, it belongs to the South Dravidian group. The Dravidian languages stand apart from other family of Indian languages like Indo Aryan, Sino Tibetan and Austro Asiatic by having distinctive structural differences at phonological, morphological, lexical, syntactic and semantic levels.

Telugu is a Central Dravidian language primarily spoken in the state of Andhra Pradesh, India, where it is an official language [3]. According to the 2001Census of India, Telugu is the language with the third largest number of native speakers in India (74 million), 13th in the Ethnologue list of most-spoken languages world-wide, and most spoken Dravidian language. As the English Language has ASCII encoding system for identifying the specification of a character, similarly Indian Languages have encoding systems named *Unicode* such as "UTF-8" , "UTF-16","UTF-32", ISCII. We are here using the character

encoding system for Indian Languages particularly for Kannada and Telugu is Unicode [4] Text Format "UTF-8".

2. NATURAL LANGUAGE IDENTIFICATION

Language identification is a subset of text categorization, is the process of determining which natural language given content is in. Traditionally, identification of written language - as practiced, for instance, in library science - has relied on manually identifying frequent words and letters known to be characteristic of particular languages. More recently, computational approaches have been applied to the problem, by viewing language identification [4] [5] as a special case of text categorization, a Natural Language Processing approach that relies on statistical methods.

The most widely used methods to programmatically identify the language of a given text compare the characteristics (usually called features) of the text with those most common in the various languages. The features most often compared are n-grams [6] [7]. Given a string, an n-gram is defined as any sequence composed by n consecutive characters. The basic idea is that train a language identifier on a large corpus of text from a given language. "Training" means gathering compression/frequency/information data on *n*-gram occurrence. The basic terms as follows:

- Character Unigram: a unique single letter ('a', 'b', …'z' for English, ಅ,ಆ,ಇ,ಈ-Kannadakkannada)
- Character bigram: a unique two-letter long sequence ("aa", "ab" ..)
- Character trigram: a unique three-letter long sequence ("aaa", "aab", ...)
- Character n-gram: a unique n-character long sequence of letters
- N-gram frequency: how frequently an n-gram appears in (some sample) text
- Character encoding: how character is represented. For example, map the integers 0-255 (one byte) to Latin characters (32 ↔ "_", 65 ↔ "A", 97 ↔ "a")

2.1 Text Categorization Problem

The problem of categorization [8] can be described as the classification of documents into multiple categories. We have a set of *n* categories {C_1, C2, …Cn} to which we assign *m* documents {D_1, D_2, … D_m} as shown in Figure 1.

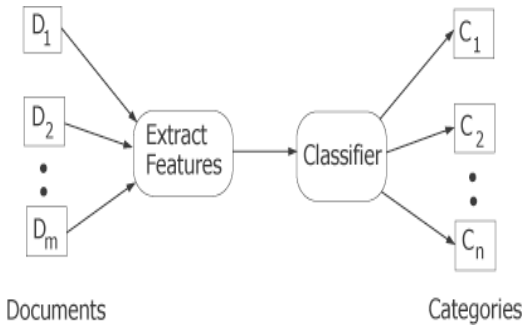

Figure 1: Assignment of Documents to Categories

The *n* categories are predefined with specific keywords that differentiate any category C_i from every other category C_j. The process of identifying these keywords is called feature extraction. Not all the words of a document are useful discriminator, for this Inverted Document Frequency (IDF) method is used to create document representatives. A keyword was weighted higher if it occurred often in one document and rarely across the collection. Such keywords (features) are useful discriminators to isolate relevant documents during retrieval. We compute the fraction of times P_{im} that a word *m* appear in document *i* is shown in equation (1).

$$P_{im} = \frac{c_{im}}{\sum_{i=1}^{N} c_{im}} = \frac{c_{im}}{c_m} \quad (1)$$

Where C_m is the total number of times word *m* appears in a collection of *N* documents.

2.2 Tokenization

Extracting words from text may appear to be simple task. The top-down method breaks text on whitespace characters such as a space, Tab, or a punctuation character. Nonwhitespace characters are concatenated to form a word or *token*. The bottom-up method builds tokens one character at a time from a text stream until a nontoken character is encountered. The simplest definition of a *token* is any consecutive string of alphanumeric characters. Between tokens, we find one or more nontoken characters. A basic tokenizer is easy to implement and shown in Algorithm 1.

Algorithm 1. Tokenizer

Input: Document which contains set of English sentences.
Output: Number of Tokens.

1. Define the set of legal token characters (alphanumeric characters and operational characters) and initialize a token list.
2. Scan a text stream one character at a time; if the current character is not in the ASCII range of 32 to 122, assign a space to the character.
   a. If the character is a token character:
      i. If the previous character was not a token character, add the previous token to the list and create a new token
      ii. Concatenate the current character to the current token. Continue at step 2.
   b. If the current character is a space character:
      i. If the previous character was not a space, add the previous token to the list.
      ii. Create a new token with a space (consecutive space from a token) continue at step 2.
   c. Default: All other characters from individual tokens.
3. Handle the last token.

Example: given the Text stream

This is a CLIR Project

The above function returns the tokens as: **this space is space a space CLIR [8] space Project**. This simple sentence fragment does not have any complex words and easy to split into tokens. Unfortunately, text is rarely this simple, as the following examples of complex tokens shows. The challenges are described here.

2.3 Challenges while generating tokens

1. Nonaphanumeric characters are mixed with alphanumeric characters to form tokens (Yahoo!, AT&T, or Hancock & Co.)
2. A period is usually a sentence separator, but it can be found in abbreviation, the initials for a person, or in the internet address (Mr.John, 1.5m, or 127.1.1.1).
3. Hyphens join words that usually belong to two different tokens to form a single token (X-ray, Large-Scale). A hyphen can be found in range of numbers (32-122).
4. The slash character (/) is a directory separator for file names in some operating systems example UNIX.
5. Web URL and email address es can have a number of embedded Nonaphanumeric characters (www.google.com/search?num=200).
6. A number can be expressed in many forms as it may be real, integer, with a exponent or with a sign.
7. A format of phone numbers varies from country to country.
8. Emoticons are short forms to express the emotions.
9. A string of words as *with respect to* or *kick the bucket* is interpreted as a single token even though spaces are found within such tokens. These above complex token should be treated as individual units of text.

2.4 Stemming

Word Stemming [9] is common form of language processing in most Information Retrieval (IR) systems. Word stemming is an important feature supported by present day indexing and search systems. Idea is to improve recall by automatic handling of word endings by reducing the words to their word roots, at the time of indexing and searching. Stemming is usually done by removing any attached suffixes, and prefixes from index terms before the assignment of the term. Since the stem of a term represents a broader concept than the original term, the stemming process eventually increases the number of retrieved documents.

Why do we need Word Stemming in the context of Free Text Searching?

Free text-searching, searches exactly as we type in to the search box, without changing it to thesaurus term. It is difficult for the end user to decide upon which all terms to key in and get the results. At this point word stemming will be needed. It is observed that in most cases, morphological variants of words have similar semantic interpretations and can be considered as equivalent for the purpose of IR applications. It also reduces the *dictionary size*, that is, the number of distinct terms needed for representing a set of documents. A smaller dictionary size results in a saving of storage space and processing time.

How Word Stemming works?

Stemming broadens our results to include both word roots and word derivations. In natural language processing, *conflation* is the process of merging or lumping together non-identical words, which refer to the same principal concept. It is commonly accepted that removal of word-endings (sometimes called *suffix stripping*) is a good idea; removal of prefixes can be useful in some subject domains (chemistry is an obvious example), but is not so widely practiced. The most obvious method for comparing the usefulness of Stemmers (Automatic programs that stem the word based on some algorithm) for the field of IR is by their impact on IR performance, using a testing system and a 'test collection' of documents, queries and relevance judgments. The process of stemming is important to the operation of classifiers and index builders/searchers because it makes the operations less dependent on particular forms of words and therefore reduces the potential size of vocabularies, which might otherwise have to contain all possible forms. It might be useful to think of stemming as the automatic definition of a group of synonyms for a particular word.

Letter Frequencies in Kannada, Telugu and English

In each language, letter frequencies have properties that may exploit to automatically detect the identity of the language. Each language has a set of letter probabilities collected from a representative corpus. Given a document in an unknown language, the letter frequency of its text is computed and letter probabilities are computed for each letter using binomial formula as shown in equation (2) for each letter. A weighted combination of the probabilities for each letter gives the probability of the identity of the language. The language with the highest probability is selected as the language of the document text.

$$nc_r p^r (1-p)^{n-r} \quad (2)$$

The two parameters of this distribution are *n*, the number of trails, and p, the probability of a occurrence of the Unigram characters in the document.

3. MACHINE TRANSLATION

Once the Language Identification task is completed than the next task is to translate the document. Machine translation [10] [11] is the process of translating from source language text into the target language. Following is a list of challenges one has to face when attempt to do machine translation.

- Not all the words in one language have equivalent words in another language. In some cases a word in one language is to be expressed by group of words in another.
- Two given languages may have completely different structures. For example English has SVO structure while Kannada/Telugu has SOV structure.
- Sometimes there is a lack of one-to-one correspondence of parts of speech between two languages. For example, color terms of Kannada/Telugu are nouns whereas in English they are adjectives.

- The ways sentences are put together also differ among languages.
- Words can have more than one meaning and sometimes group of words or whole sentence may have more than one meaning in a language. This problem is called ambiguity.
- Not all the translation problems can be solved by applying values of grammar.
- It is too difficult for the software programs to predict meaning.
- Translation requires not only vocabulary and grammar but also knowledge gathered from past experience.
- The programmer should understand the rules under which complex human language operates and how the mechanism of this operation can be simulated by automatic means.
- The simulation of human language behavior by automatic means is almost impossible to achieve as the language is open and dynamic system in constant change. More importantly the system is not yet completely understood.

3.1 Bilingual Dictionary

Bilingual dictionary is a crucial part not only for machine translation, but also for other natural language processing applications such as cross-language information retrieval. Creating a bilingual dictionary in the form of lexemes or words is a difficult task as it covers more than one area of meaning, but these multiple meanings don't correspond to a single word in the target language. Basically machine translation systems are linked to electronic dictionaries. The content of the dictionaries must be adequate in both quantity and quality: that is, the vocabulary coverage must be extensive and appropriately selected, and the translation equivalents carefully chosen, if target language output is to be satisfactory or indeed even possible. The size and quality of dictionary limits the scope and coverage of a system, and the quality of translation that can be expected. The dictionary [12] entries are based on lexical stems of specified category, strictly monolingual analysis and generation dictionaries, and transfer dictionaries based on language-pair-specific information.

3.2 Transliteration

The Language transliteration is one of the important area in natural language processing. Machine Transliteration is the conversion of a character or word from one language to another without losing its phonological characteristics. It is an orthographical and phonetic converting process. Therefore, both grapheme and phoneme information should be considered. Accurate transliteration of named entities plays an important role in the performance of machine translation and cross-language information retrieval CLIR [13] [14] process. Transliteration should not be confused with translation, which involves a change in language while preserving meaning. is the acronym of a great variety of techniques, systems and technologies that associate information retrieval (normally from texts) in multilingual environments. Dictionaries have often been used for query translation in cross language information retrieval. However, we are faced with the problem of translating Names and Technical Terms from English to Kannada/Telugu. The most important query words in information retrieval are often proper names. We present a method for automatically learning a transliteration [15] model from a sample of name pairs in two languages as shown in Figure 2.

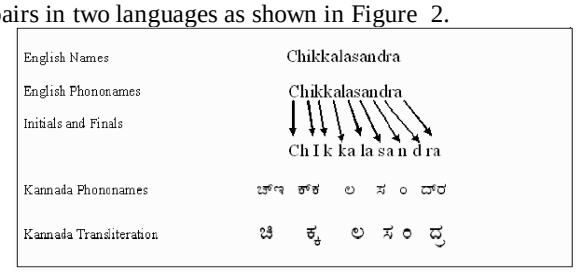

Figure 2: Example: English-Kannada Name Transliteration

4. EVALUATION METRIC

Using the unigram statistical approach for each Language, the proposed model is learnt with a training data set of 100 text lines from each of the three Languages- English, Kannada and Telugu. Language Identification Algorithm 2 used in the proposed model

Algorithm 2. LangId ()

Input: Pre-processed text lines of English, Kannada and Telugu text Documents.

Output: Identify the Language of the document.

1. Do for i = 1 to 3 Language document types
2. Do for k = 1 to 100 text lines of i^{th} document
3. Compare until i==k if yes display the type of language
4. Otherwise display the unknown language

Precision in equation.(3) and Recall equation(4) are the two metrics used to evaluate Information Retrieval (IR) [16] systems, Text Categorization, entity extraction and Q&A systems. They have become the standard measures for Many NLP/IR tasks. These are computed as follows:

$$precision = \frac{Categoriesfoundandcorrect}{Totalcategoriescorrect} \quad (3)$$

$$recall = \frac{Categoriesfoundandcorrect}{Totalcategoriesfound} \quad (4)$$

5. EXPERIMENTAL SETUP, RESULTS AND PERFORMANCE GRAPH

Several corpora were collected to estimate the parameters of the proposed models and to evaluate the performance of the proposed approach. The corpus BUBShabdaSagara-2011 [17] for training consisted of 9000 words in dictionary for Kannada/Telugu. The training corpus composed of a bilingual word list. In the experiment, the performance of word translation extraction was evaluated based on precision and recall rates at the word. Since, we considered exactly one word in the source language and one translation in the target language at a time as shown in Figure 3 and Figure 4.

Performances of the systems were evaluated with the same set of 500 distinguished sentences that were out of corpus. From the experiment we found that the performance is shown in Fig.5 of our systems are significantly well and achieves very competitive accuracy by increasing the corpus size.

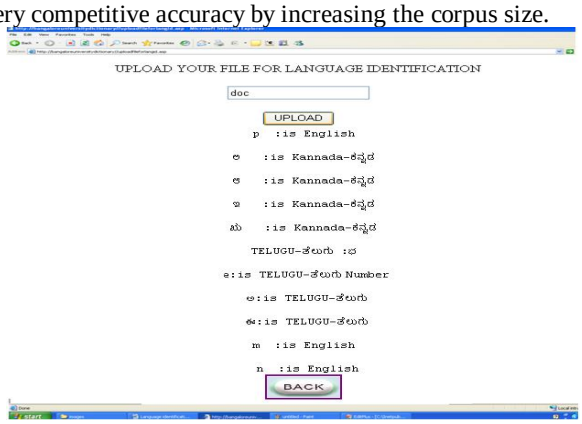

Figure 3: Identifying the different Languages present in a document

Figure 4: Language identification for English, Kannada and Telugu by uploading documents.

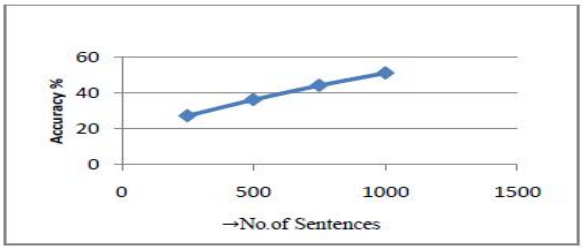

Figure 5: Accuracy increases as word increases.

6. CONCLUSION AND FUTURE WORK

In this paper, a method to identify and separate text lines of English, Kannada and Telugu documents from a trilingual document is presented. The approach is based on the analysis of the *Unigram statistical approach* of individual text lines and hence it requires character or word segmentation. By using bilingual dictionary now we are getting word by word translation. In future we can also use this language identification module for translation. This will be very useful for bilingual machine translation from English to Kannada/Telugu language. One of the major challenges is that English has Subject Verb Object (SVO) structure while Kannada has Subject Object Verb (SOV) structure in Machine translation will be unraveled by using morphology we will get a sentence to be translated.

ACKNOWLEDGEMENT

I owe my sincere feelings of gratitude to Dr. M. Hanumanthappa, for his valuable guidance and suggestions which helped me a lot to write this paper. This paper is in continuation of the major research project entitled "***Cross-Language Information Retrieval"*** sanctioned to Dr. M. Hanumanthappa, PI-UGC-MH, Department of Computer Science and Applications by the University Grant Commission carried out at the Bangalore University, Bangalore, India. We thank to the UGC for financial assistance.

REFERENCES

1. ***Text Mining Applications***, 4th ed., Manu Konchady, Indian prints, 2009.

2. Karnataka Official Language Act. ***Official website of Department of Parliamentary Affairs and Legislation***. Government of Karnataka. Retrieved 2012-07-29.

3. http://en.wikipedia.org/wiki/Telugu_language

4. Gerrit Botha.*, Victor Zimu *a*nd Etienne Barnard. **Text-Based Language Identification For South African Languages**, Published in *South African Institute Of Electrical Engineers*, Vol.98 (4) Dec 2007

5. Penelope Sibun, and Jeffry C Reynar. **L*anguage Identification: Examining the issues***

6. Tommi Vatanen, Jaakko J. V¨ayrynen and Sami Virpioja. *"**Language Identification of Short Text Segments with N-gram Models**"*

7. Bashir Ahmed, Sung-Hyuk Cha, and Charles. *Tappert **"Language Identification from Text Using N-gram Based Cumulative Frequency Addition"*** published in Proceedings of Student/Faculty Research Day, CSIS, Pace University, May 7th, 2007

8. William J. Teahan and David J. Harper. *"**Using compression based language models for text categorization "***

9. http://tartarus.org/~martin/PorterStemmer/

10. ***Machine Translation***, Prof. Abdullah H. Homiedan

11. S. Kereto, C. Wongchaisuwat, Y. Poovarawan. 1993. **Machine translation research and development,**. In proceedings of the Symposium on Natural Language processing in Thailand, pp. 167-195, March

12. Knowles, Francis. **The Pivotal Role of the Dictionaries in a Machine Translation System,** In Lawson, Veronica, ed. Practical Experience of Machine Translation". North-Holland. 1982.

13. Prasad Pingali and Vasudev Varma. **Hindi and Telugu to English Cross Language Information Retrieval at CLEF 2006,** In working notes for the CLEF 2006 workshop (Cross Language Adhoc Task), pp.20-22 September, Alicante, Spain

14. Mallamma. V Reddy and Hanumanthappa. *M. **Kannada and Telugu Native Languages to English Cross Language Information Retrieval,*** Published in the International Journal of Computer Science and Information Technologies (IJCSIT) volume-2 issues-5 September-October 2011. ISSN: 0975-9646, pp. 1876-1880. Available online at www.ijcsit.com/docs/Volume%202/vol2issue5/ijcsit2011020510.pdf

15. Mallamma. V Reddy, Hanumanthappa. M. ***English to Kannada/Telugu Name Transliteration in CLIR: A Statistical Approach***, Published in the *"International Journal of Machine Intelligence (IJMI)"* IISN: 0975-2927 and E-IISN: 0975-9166, Vol.3, Issue 4 available at online at http://www.bioinfo.in/uploadfiles/13258352463_4_30_IJMI.pdf. pp: 340-345

16. ***Modern Information Retrieval,*** Ricardo Baeza-Yates, Berthiero-Neto Mallamma. V Reddy, Hanumanthappa. M. *CLIR Project (English to Kannada and Telugu).* http://bangaloreuniversitydictionary//menu.asp