Volume 5, Issue 7, July 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Language Identification: Contrivance Learning Process using Web Based Disquisition

[1]Rashmi S, [2]M Hanumanthappa, [3]Mallamma V Reddy

[1, 2] Department of Computer Science & Applications, Bangalore University, Bangalore, India

[3]Department of Computer Science, Rani Channamma University, Vidyasangam, Belgaum, India

Abstract--- ***Language identification is the foremost task in the study of linguistics .The projections of language identification & conversions such as Google translate or any other hypothetical translator works in wonders. The mechanism of detecting the language performed by these translators is a real marvel. Hence in this divertissement it is of the primary importance to study the methods of identifying the language. In this paper, the methodologies of recognizing some of the Natural Languages such as English, Kannada, Hindi & Telugu is explained on the basis of N-Gram algorithm and the respective vowels and consonants of each of the languages are retrieved and stored for building the syntactic structure of the corpus.***

Keywords--- ***N-gram algorithm, Naïve Bayes, Nearest-Neighbour, Stop words, Text categorization, Text Classification, Unicode.***

I. INTRODUCTION

Growth of internet is increasing everyday and the news has done wonders for our morale. With the advent of internet, the online documents and repositories are becoming huge in number and in size. The pandemic usages of these documents are relatively more when compared to the hardcopy. This exhibit the feature of turbulence as the languages of various documents are dissimilar and language identification flatters with more prevalence. In culmination of this, the language identification and text categorization rooted on the content becomes indispensable & essential. The process of instinctively designating the given text into a set of predefined categories based on the content is called as Text Classification alias Text Categorization [1]. Some of the techniques used for the Text Categorization are Naïve Bayes [2], Decision Tree [3], Support Vector Machine (SVM) [4], K-Nearest Neighbor (KNN)[5],[6] and so on

In this research paper we describe the task of identifying the language on web documents and perform tokenization of these Unicode and draw the vowels and consonants of the particular language. This is also an investigation of character encoding and development of an interface to serve the above said purpose.

II. BACKGROUND RESEARCH

Language Identification is not any man touched subject. The concept ways back to the history which is analogous to gold mining. Given a possible set of languages and its associated languages code and names, apply the contexts to the code and compare it and would give you the exact language. The task is quite different when electronic machine, computer is involved in it. Computer need to be trained prior the task of text categorization. It is now ready to accept the input which then classifies the language to its respective class. Hence language identification is a classification problem of data mining.

The unsurpassed popular text categorization method is Cavnar and Trenkle (1994). This method uses the character frequency of a language and then subjects to mark the classification based on the relative distance of the frequency of various language. Bayesian model (Dunning 1994) is one of the variant of this model. Language identification is carried out for Linguistics as well. Johnson (1993) constructed a list of stop words. Stop words are defined as the words that are most habitual in use in any Natural Language. The search engines are programmed to disregard these stop words when retrieving the result of a search query. In table [1], a list of stop words of four languages are shown

Table 1- Stop words [7][8][9][10]

English	**Kannada**	**Hindi**	**Telugu**
The	ಇದು	में	ఏమి
Be	ಏನು	क्या	ఉంటుంది
As	ಎಂದು	होना	వంటి

Stop words are very common in any language hence language identification can be rooted to the list of stop words. With the list prepared the input was compared. The highest overlap of these stop words for any language wins the race. Tests on part of speech and tokenization were carried out to recognize the language by Grefentette (1995) and Giguet (1995) respectively.

Every Language uses white space as a delimiter. So once a white space is found, delimit the word and is sent for comparison based on the stop words that is built. This is suitable for a document which has one universal language however for languages such as Thai, Japanese, and Chinese it would be difficult to adopt this approach where there is no definite word bound. Nevertheless is relatively less when compared to other most populous web languages.

III. N-GRAM BASED TEXT CATEGORIZATION

3.1 N-gram Elucidation

N-gram is a series of n-items for any given sequence of context, where the series of n-items or “grams” includes anything from characters to string or to complex sentences. For a given context, it is said to be 1-gram if all the items in the sentence is split into 1 string, it is called 2-gram if it contains 2-items of the sentence, and so on respectively. Table [2] discusses the various 2-gram words in the four languages.

Table 2- N-gram for different languages

Language	Word	1-gram	2-gram	3-gram	4-gram	5-gram
English	What are you doing?	What	What are	What are you	What are you doing	Nil
		Are	are you	Are you doing		
		You	you doing			
		Doing				
Hindi	आप क्या कर रहे हैं:	आप	आप,क्या	आप क्या कर	आप क्या कर रहे	आप क्या कर रहे हैं:
		क्या	क्या,कर	क्या कर रहे	क्या कर रहे हैं:	
		कर	कर,रहे	कर रहे हैं		
		रहे	रहे,हैं			
		हैं:				
Kannada	ನೀವು ಏನು ಮಾಡುತ್ತಿದ್ದೀರಿ	ನೀವು	ನೀವು ಏನು	ನೀವು ಏನು ಮಾಡುತ್ತಿದ್ದೀರಿ	Nil	Nil
		ಏನು	ಏನು ಮಾಡುತ್ತಿದ್ದೀರಿ			
		ಮಾಡುತ್ತಿದ್ದೀರಿ				
Telugu	మీరు ఏమి చేస్తున్నారు	మీరు	మీరు,ఏమి	మీరు ఏమి చేస్తున్నారు	Nil	Nil
		ఏమి	ఏమి,చేస్తున్నారు			
		చేస్తున్నారు				

3.2 Why’s and wherefore’s of Text Categorization

Cavnar and Trenkle (1994) redefined the zipf’s laws as “The nth most common word in a human language occurs with a frequency inversely proportional to n”. This is shown using the formula

$f \propto 1/t$ where f is the frequency of the word and r is the rank of the word in the list ordered by the frequency [1]. The basic surmise of zipf’s law is that, there are certain set of words which transpire proportionally more persistent than the other words in the sentence. Dialectal and domain sovereignty can be attained by using n-gram for a language. The need of n-gram is evident for language identification as it enables the collection of various word forms for the given context. It must be noted that character level sliding window succour the boundaries for the word. These key features are essential when determining the language.

3.3 Data set

The experiments conducted in the present work, we deploy Wikipedia as it contains more number of languages when compared to other datasets such as EUROGOV, TCL.

Wikipedia and its statistics

Distribution of linguistic features using Wikipedia is shown in the below in the table [3]. The table shows the number of documents and the diverse varieties of languages available in Wikipedia. This figure stands out from other text corpus and the main reason behind choosing the Wikipedia for the present work

Table 3- Wikipedia Report

	Languages	Documents	Document Length (bytes)
Wikipedia	**69**	**4687**	**1484.5 ± 5086.8**

3.4 Document Representation

The aim of the paper is text categorization for a language. In view to this, one language is kept as universal making no unrealistic assumptions. For this purpose it is important to proceed in the notion that we do not know any source language and that is being found out. The use of byte n-grams was studied and perceived as this does not require encoding detection.

IV. MODELS

A diverse number of models were used for conducting the experiment, as described below. Nearest-Neighbour & Naïve Bayesian models are the popular ones in subject to text characterization. This has been delineated in the following section.

4.1 Nearest-Neighbour Models

This is a much sought-after model in the field of linguistics. Nearest-Neighbour Model classifies the test document say D based on the language of the closet training set say Dr by representing the training document as a sole state. This is done to show the cost of classification is proportional to the diverse number of documents sets [12]. It calculates the distance measure based on following key attributes

Skew Divergence (SKEW): the use of smoothing factor is prominent here along the linear interpolation of r and s axis. The smoothing factor is denoted by $\alpha=0.9$

$Z\ \alpha(r,s)=D(r||\ \alpha s + (1- \alpha)r)$

Where

$D(r||s)= \sum\Sigma ri(Log2ri- Log2sj)$

4.2 Naïve Bayesian

This is famous for its robust and simple execution techniques for text summarization. The test document is considered here is D. The model relies on calculating the probabilities of likelihood of patterns [11], [12], [13].

$$\begin{aligned} p(C_k|x_1,\dots,x_n) &\propto p(C_k,x_1,\dots,x_n) \\ &\propto p(C_k)\ p(x_1|C_k)\ p(x_2|C_k)\ p(x_3|C_k)\ \cdots \\ &\propto p(C_k)\prod_{i=1}^{n} p(x_i|C_k)\,. \end{aligned}$$

$$p(C_k|x_1,\dots,x_n) = \frac{1}{Z}p(C_k)\prod_{i=1}^{n} p(x_i|C_k)$$

V. EXPERIMENTAL METHODOLOGY

The experiment was carried on the interface that was developed to identify the language. The interface primarily identifies four native languages such as English, Hindi, Kannada and Telugu. This task has been classified into two phases

- Characteristic Generation and feature Stipulation
- Training Phase

5.1 Characteristic Generation and Feature Stipulation

In order to identify the language of the document, the classifier must be fed with the knowledge of various languages. This is crucial as representation varies along with the usage of applications. Feature stipulation is identifying and representing the characteristics unique to any language. In the present work, various stop words of the four languages such as English, Kannada, Hindi and Telugu are stored for the catastrophic view of these languages and thus the stop words are used to identify the language. The fundamental idea behind the stop words is that, every language contains some of the basic terminologies that are regimented deep within. This forms the ground work to carry out the further process.

5.2 N-gram Construction

The stop words of Feature stipulation phase is taken further in the N-gram construction. The stop words of various languages are broken down into the family of N-gram profiles. The first step in the formation of N-gram profile is to recognize the various special occurrences such as white spaces, newline character, special characters, and tab characters. The next step is to remove all such occurrences. Multiple spaces were reduced to one space. The generated N-gram was given an identifier which uniquely identifies itself from rest of the words. This was done with the help of hashing. The generated identifiers were mapped onto the hash table. Every time a unique word is encountered, its respective hash key will be stored in the hash table. The use of hashing and hash tables helps for the frequency count.

5.3 Training Phase

After constructing the list of stop words for each of the four languages and chopping the special occurrences, these words are mapped on to the training corpus. When there is a win- win condition, that particular language is retrieved as the output. This is elucidated in the training phase. Text identification is a data driven module which purely depends on the training set. Hence this forms the classification technique of Data Mining as the given text is compared with the training set of existing dataset. Here the language with the highest word count is chosen as a winning language. Figure[1] shows the representation of the three phases.

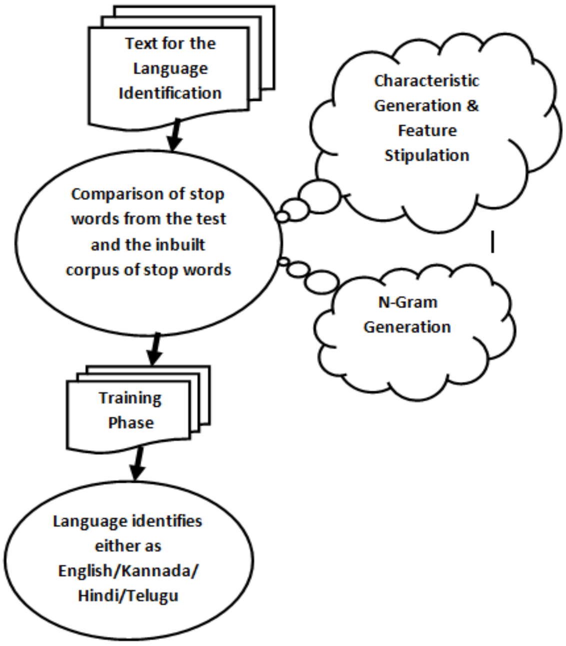

Text for the Language Identification
Characteristic Generation & Feature Stipulation
Comparison of stop words from the test and the inbuilt corpus of stop words
N-Gram Generation
Training Phase
Language identifies either as English/Kannada/ Hindi/Telugu

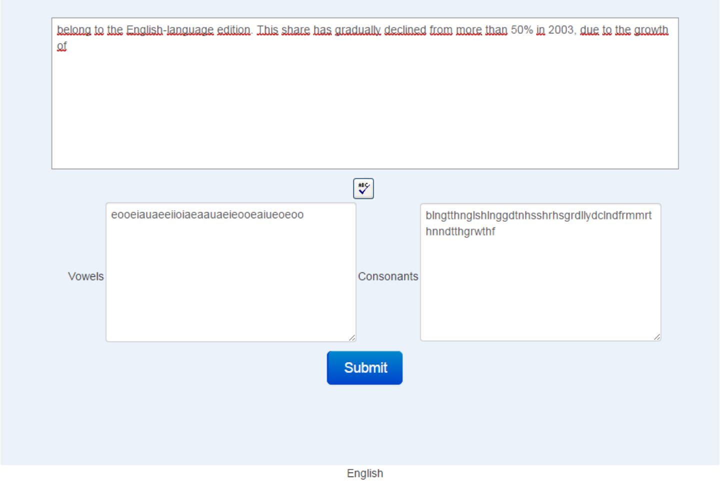
Figure 1- Flow Diagram of phases

VI. RESULTS

For the experiment purpose documents from Wikipedia has been randomly chosen for each of the four languages. This resulted in around 250 test cases. The size varies of the document varies from document to document. The sizes of the documents that were chosen for our experiment were approximately within 100 to 2000 words. This word count represents the formation of words with a single space after the special occurrences and multiple spaces were eliminated as described in the section 5. The experimental analysis is done for the four languages English, Kannada, Hindi & Telugu. This is explained in the next section.

6.1 Experimental Results

This section explains the implementation of the above said mechanisms. The source code was generated by using asp.net as front end and Microsoft Word and Microsoft Excel as the back end. The word document contains stop words of all the four languages which is evaluated by the source code in order to recognize the distinct language. Figure 2 shows the experimental results for one of the four languages and the language were successfully identified as English. The same has been studied for the other three languages namely Kannada, Hindi and Telugu. The results were tested and verified.



belong to the English-language edition. This share has gradually declined from more than 50% in 2003, due to the growth of

Vowels: eooeiauaeeiioiaeaauaeieooeaiueoeoo

Consonants: blngtthnglshlnggdtnhsshrhsgrdllydclndfrmmrt hnndtthgrwthf

English

Figure 2- Experimental Output

6.2 Algorithm

Table 4 shows a part of the working algorithm that was developed during the work. The code has manifested only for the language Kannada. Perhaps the interface was developed to forecast the document identification for all the four languages mentioned in the earlier sections. The algorithm efficiently identifies the language out of the four source languages and the vowels and the consonant of the identified language is also displayed as shown in the figure-2.

Table 4: Algorithm of the working interface

```
Step 1:   Input the text or document which is either in
any of the following four languages ( English,
          Hindi, Kannada, Telugu)
     ➢   Document document = new
          Document(PageSize.LETTER);
Step 2:   we create a writer that listens to the document
and directs a PDF-stream to a file
     ➢   PdfWriter.GetInstance(document, new
          FileStream("Chap0101.pdf",
          FileMode.Create));
Step 3:    We open the document
     ➢   document.Open();
Step 4:   Proceed with Language identification with
each character in the context being compared
          with the stop words and its associated
Unicode and character code
     ➢   foreach (var image in images)
           iTextSharp.text.Image pic =
iTextSharp.text.Image.GetInstance(image,
          System.Drawing.Imaging.ImageFormat.Jpeg)
;
           document.Add(pic);
           document.NewPage();
           catch (DocumentException de)
           Console.Error.WriteLine(de.Message);
           catch (IOException ioe)
           Console.Error.WriteLine(ioe.Message);
Step 5:   we close the document
     ➢   document.Close();
          string lanCode =
ld.Detect(TextBox1.Text);
          reader = new SpeechSynthesizer();
          Hashtable hashtable = GetHashtable();
           string[] eachelemrnt =
TextBox1.Text.Split(' ');
          foreach (string singletext in
eachelemrnt)
           for (int i = 0; i < eachelemrnt.Length;
i++)
            if (keys == eachelemrnt[i])
          for (int i = 0; i < eachelemrnt.Length;
i++)
          if (keys == eachelemrnt[i])
           txtcon.Text = txtcon.Text +
eachelemrnt[i];
Step 6:   The analogy of the same paradigm is
applied for other three languages (English, Hindi,
Telugu)

Step 7:   The stored vowels and Consonants of
the recognized language is retrieved along with
the name of the language
//END
```

6.3 Result Evaluation

The accuracy and time complexity is utmost important for any experimental results. In this section the experimental results are evaluated to prove its ranking. When a document consists of one language as universal, the evaluation imposed no challenge. The accuracy of 94% was achieved where the recall and precision were found to be 0.93 and 0.95 respectively. Time complexity is equal to O (1). This work was stimulated with these comparatively good results however though the initial performance is very much convincing, the interface do not ensure the correct identification when the document consists of multiple languages.

VII. FUTURE ENHANCEMENTS

The present work fortifies the research under the field of linguistics. The same work can be carried to eliminate the challenges and to overcome the downside of the present system. The future work concentrates on evaluating a multi-lingual document using the existing interface. The mechanism explained in the current system can be applied and extended for other languages. The next step is to congregate towards building the syntactic representation for various languages to count on the grammar and the possible grammar mistakes.

VIII. CONCLUSION

Text summarization or text categorization is vigorous research field that has been in the agile from some time now. Though language identification seems an easy topic to touch on, it suffers from various challenges and shortcomings due to the ambiguous nature of any natural language and merging of topics between the languages. Some of the popular approaches for the purpose are naïve Bayes, N-gram, Nearest- Neighbour. The eloquence of the methodologies to achieve text categorization is explored in a much better way. In this paper, above said techniques are sifted to make the text categorization more efficient and robust. The methods proposed in the present work deviates from the existing ones and the evaluation of the experiment is also shown.

REFERENCES

[1] Analysis of N-Gram Based Text Categorization for Bangla in a Newspaper Corpus Munirul Mansur, Naushad UzZaman and Mumit Khan Center for Research on Bangla Language Processing, BRAC University, Dhaka, Bangladeshmunirulmansur@hotmail.com, naushad@bracu.ac.bd, mumit@bracu.ac.bd

[2] W.B. Cavnar and J.M. Trenkle, "N-Gram-Based Text Categorization", In Proceedings of SDAIR-94,

[3] 3rd Annual Symposium on Document Analysis and Information Retrieval, 1994.

[4] R.J. Mooney and L. Roy, "Content-Based Book Recommending Using Learning for Text Categorization", In the proceedings of DL-00, 5th ACM Conference on Digital Libraries, 1999.

[5] J. Fürnkranz, "A Study Using n-gram Fetures for Text Categorization", http//citeseer.ist.psu.edu/johannes98study.html, 1998.

[6] http://en.wikipedia.org/wiki/Language_identification

[7] J.P.R. Gustavsson, "Text Categorization Using Acquaintance", Diploma Project, Stockholm University,http://www.f.kth.se/~f92-jgu/Cuppsats/cup.html, 1996, unpublished.

[8] http://en.wikipedia.org/wiki/Stop_words

[9] Categorized Text Document Summarization in the Kannada Language by sentence ranking, Jayashree, R et all, Intelligent Systems Design and Applications (ISDA), 2012 12th International Conference, IEEE sponsored. print ISBN 978-1-4673-5117-1. DOI:10.1109/ISDA.2012.6416635

[10] Evaluating Effect of Stemming and Stop-word Removal on Hindi Text Retrieval, Amaresh Kumar Pandey, Proceedings of the First International Conference on Intelligent Human Computer Interaction 2009, pp 316-326, Springer.

[11] Study of Indexing Techniques to Improve the Performance of Information Retrieval in Telugu Language Kolikipogu Ramakrishna et al, International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 1, January 2013) 482

[12] Use of Statistical N-Gram Models in Natural Language Generation For Machine Translation, *Fu-Hua Liu*, Proceedings of 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), , IEEE. , vol.1, p.636-9 in 2003

[13] M. Forsberg and K. Wilhelmsson, "Automatic Text Classification with BayesianLearning", http://www.cs.chalmers.se/~markus/LangClass/LangClass.pdf

[14] http://en.wikipedia.org/wiki/Naive_Bayes_classifier