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, Bangladesh
munirulmansur@hotmail.com, naushad@bracu.ac.bd, mumit@bracu.ac.bd

Abstract

In this paper, we study the outcome of using n-gram (n character slice of a longer string)based algorithm for Bangla text categorization. To analyze the efficiency of this methodology we used one year Prothom-Alo news corpus. Our results show that n-grams of length 2 or 3 are the most useful for categorization. Using gram lengths more than 3 reduces the performance of categorization.

Keywords: n-gram, Prothom-Alo, Text categorization, Text classification, Zip's Law.

I. INTRODUCTION

The widespread and increasing availability of text documents in electronic form increases the importance of using automatic methods to analyze the content of text documents. The method of using domain experts to identify new text documents and allocate them to well-defined categories is time-consuming, expensive and has its limits. As a result, the identification and categorization of text documents based on their contents are becoming imperative. Text categorization, also known as text classification, concerns the problem of automatically assigning given text passages (paragraphs or documents) into predefined categories. The task of text categorization is to automatically classify documents into predefined classes based on their content. A number of statistical and machine learning techniques has been developed for text classification, including regression model, k-nearest neighbor [10], decision tree, Naïve Bayes [4], [11], support vector machines [5], n-gram based [2], [7], [13], and so on. Such techniques are currently being applied in many areas, including language identification, authorship attribution, text genre categorization, news categorization [2], recommendation systems [4], [6], [12], Spam filtering [8] etc. In this paper we analyze the performance of n-gram based text categorization for Bangla

II. N-GRAM BASED TEXT CATEGORIZATION

A. What is n-grams?

Before describing the reasoning behind selecting n-gram for text categorization, we will give a short description of what n-gram is. An n-gram is a subsequence of n-items in any given sequence. In computational linguistics n-gram models are used most commonly in predicting words (in word level n-gram) or predicting characters (in character level n-gram) for the purpose of various applications. For example, the word "বাংলা" would be composed of following character level n-grams.

Table I Different n-grams for the word "বাংলা".
(Spaces are shown with '_')

	বাংলা
Unigrams	ব, া, ং, ল, া, _
Bi-grams	_ব, বা, াং, ংল, লা, া_
Tri-grams	_বা, বাং, াংল, ংলা ,লা_
Quad-grams	_বাং, বাংল, াংলা, ংলা_

So, an n-gram is a character sequence of length *n* extracted from a document. It is an n character slice of a longer string [2]. In this work one word from the document was represented as the set of overlapping n-grams. Here also leading and trailing spaces were considered as the part of the word [13]. Typically, *n* is fixed for a particular corpus of documents and the queries made against that corpus where corpus is a huge text.

B. Why n-gram based text categorization?

Human languages invariably have some words which occur more frequently than others. One of the most common ways of expressing this idea has become known as Zipf's Law, which we can re-state as follows *"The n^{th} most common word in a human language text occurs with a frequency inversely proportional to n."* [2]. In other words, if f is the frequency of the word and r is the rank of the word in the list ordered by the frequency, Zipf's Law states that $f = \frac{k}{r}$. [13]. The implication of this law is that there is always a set of words which dominates most of the other words of the language in terms of frequency of use. This is true both of words in general, and of words that are specific to a particular subject. Zipf's Law implies that classifying documents with n-gram frequency statistics will not be very sensitive to cutting off the distributions at a particular rank. It also implies that if we are comparing documents from the same category they should have similar n-gram frequency distributions. By using n-grams the system can achieve language independence. In most word-based information retrieval systems, there is a level of language dependency. Stemming and stop list processing are both language specific.

C. Why Character Level n-gram?

Words derived from the same root word tend to generate many of the same n-grams, so a query using one form of a word will help cause documents containing different forms of that word to be retrieved. The sliding window approach used in n-gram based text categorization allows us to capture n-grams corresponding to words, as well as pairs of words. The n-gram "of co", for example, is the first n-gram in the phrase "of course." This paper is based on the work of [2] and [13], who worked on n-gram based text categorization on a computer newsgroup categorization task. We employed the same technique and tried to analyze how this technique performs for Bangla news paper corpus. In this paper n-grams with various lengths were used (from 2 to 4-grams).

III. METHODOLOGY

Text categorization or the process of learning to classify texts can be divided into two main tasks: [13]

- Feature Construction and Feature Selection
- Learning phase

A. Feature Construction and Feature Selection

Texts cannot be directly interpreted by a classifier or by a classifier building algorithm. Because of this an indexing procedure that maps a text into a compact representation of its content needs to be uniformly applied to training, validation, and test document. The choice of representation for text depends on what one regards as the meaningful units of text.[3] A feature can be as simple as a single token, or a linguistic phrase, or a much more complicated syntax template. A feature can be a characteristic quantity at different linguistic levels. [15] In this work different length of n-grams were taken such feature. A document is represented by a feature vector that contains one attribute for each term that occurs in the training collection of documents. If a term occurs in a particular training document, its corresponding attribute is set to its frequency. Thus, each document is represented by the set of terms it consists of. Each distinct character n-gram is a term as well as a distinct feature of a document and the number of times the term occurs in the document is its value. Let us describe how to construct the vector space model from a document collection. For this work training documents or the category files has three document representations:

- Frequency profile
- Normalized frequency profile
- Ranked frequency profile

B. Learning Phase

After defining the document representations the classifier or the learner is trained with predefined categories. Text categorization is a data driven process for categorizing new texts. For this work, we used 1 year news corpus of Prothom-Alo. From that corpus the 6 categories were selected. The following table shows the predefined categories and the corresponding news editorials taken from Prothom-Alo.

Table II. List of predefined categories and their content source.

Defined category	Category Content	Prothom-alo Editorials
Cat1	Business News	অর্থ ও বাণিজ্য
Cat2	Deshi News	বিশাল বাংলা
Cat3	International News	সারা বিশ্ব
Cat4	Sports News	খেলা
Cat5	Technology News	কম্পিউটার প্রতিদিন , প্রজন্ম ডট কম
Cat6	Entertainment	বিনোদন

C. Generating n-gram Profiles

These following steps are executed to generate the n-gram profiles.

C.1 Creation of n-grams

In order to get rid multiple occurrence of new line character, line feed character, tab character was removed and multiple placements of spaces were reduced to one space. The n-grams of n consecutive characters are copied out of the text using a window of n characters length, which is moved over the text n character forward at a time.

C.2 Production of n-grams hash map

Every n-gram is given a unique number, called a hash key. These hash keys are stored in a hash map provided by Java utility package. Each of the generated n-gram has its unique hash key. So, every time a particular n-gram is generated it has its unique hash key and using that hash key the value of it is updated. The hash map basically acts as a table which is used to keep track of how many times each n-gram has been found in the text being studied. Every time an n-gram is picked, the element of the hash map with the number given to the n-gram is increased by one.

C.3 Creation of different document representation

When all n-grams have been extracted from a text and put into three hash maps

- Normal Frequency Profile Hash Map

- Normalized Frequency Profile Hash Map
- Ranked Frequency Profile Hash Map

C.4 Normal Frequency Profile

This hash map just contains occurrences of the n-grams in the given text. This a hash map storing the frequency distribution of all the n-grams in the given text. For example if a document has only 3 bi-grams নব, এত, ীব with frequencies 150 , 75, 50 then the generated profile will be the following

নব =150, এত = 75, ীব = 50

Document Representation:

d = (150, 75, 50).

Fig. 1. Normal Frequency profile generation

C.5 Normalized frequency profile

To generate the normalized frequency profile the previously generated normal frequency profile hash map is used. For this case each occurrence of a n-gram is divided by the sum of the frequency of all extracted n-grams. Using the previous example normalized frequency profile would be the following

নব =150, এত = 75, ীব = 50

Reverse Order Rank:

নব = 1

এত = 2

ীব = 3

Document Representation:

d = (1, 2, 3).

Fig. 2. Normalized Frequency profile generation

This means that the absolute numbers of occurrences will be replaced with the relative frequencies of the corresponding n-grams. The reason for doing this is that similar texts of different lengths after this normalization will have similar hash map. The frequencies stored in the hash map will be numbers between 0 and 1, most of them equal to or very close to zero, since most of the possible n-grams never or almost never occur.

C.6 Ranked Frequency Profile

For this hash map the normal frequency profile hash map is sorted according to the frequency of each of the n-gram generated from the given text. In this ranking the most frequent n-gram get the rank 1, that is a reverse ordering of the count of the n-grams are done. By this ranking the most frequent n-grams get lower ranks and more domains specific n-grams get higher ranks. As a result the higher rank of the n-grams the higher domain specific it is.

নব =150, এত = 75, ীব = 50

150 +75+50 = 275

Normalized frequency:

নব =0.54, এত =0.27, ীব = 0.19

Document Representation:

Fig. 3. Ranked Frequency profile generation

D. Comparing and Ranking n-gram Profiles

We start with a set of pre-existing text categories. From these, we would generate a set of n-gram frequency profiles to represent each of the categories. When a new document arrives for categorization, the system first computes its n-gram frequency profile.

It then compares this profile against the profiles for each of the categories using an easily calculated distance measure. The procedure can be illustrated by the Fig. 4.

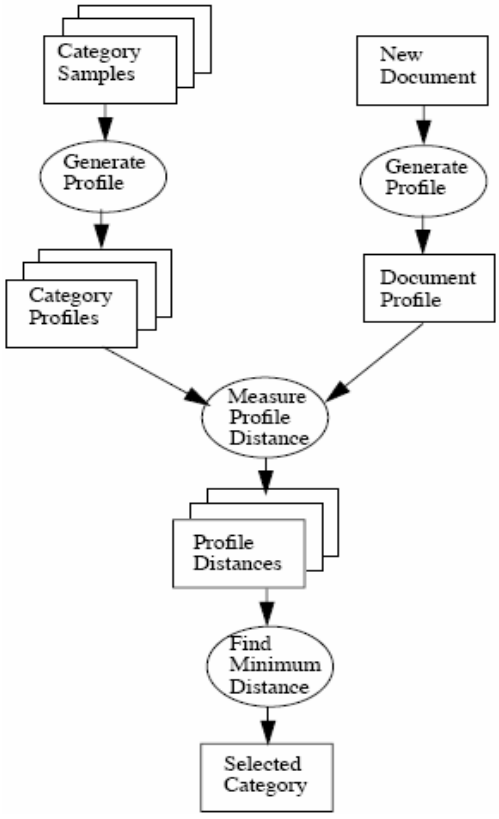

Fig. 4. Classification Procedure.

Measure profile distance is also very simple. It merely takes two n-gram profiles and calculates a simple out of place measure. This measure determines how far out of place an n-gram in one profile is from its place in the other profile. Fig. 5 shows an example of how this measuring distance profile is done while working with

ranked frequency profile. In Fig. 5, ি bi-gram has its rank same for both the category and the test documents profile. So, distance measure will 0. But for the case of এত the category profile has it on third position where as in test profile it is ranked as fifth. So, distance measure will be absolute value of (5-3=2). This scheme is repeated for each of the n-gram produced for the test document. After that the system classifies the document as belonging to the category having the smallest distance.

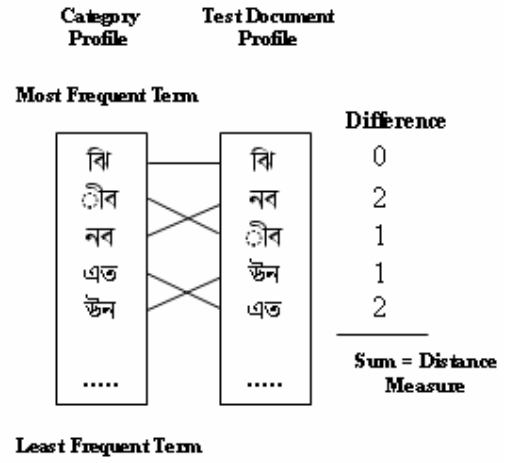

Fig. 5. Measure profile distance

E. Classification of Text

When we want to choose a category for the document, we have to count distances from all the categories profiles. Then we choose the category with the smallest distance from the document profile. As we have the list of distances from all categories, we can order them. Then we can choose most relevant categories for the given document. In this work, we used only the least distance category as the winner.

IV. RESULTS

For our experiment we randomly selected 25 test documents from each of the six categories, defined from the 1 year Prothom-Alo news corpus. So, 150 test cases were generated. All of the test cases were disjoint from the training set. The sizes of the test cases were approximately within 150 to 1200 words.

A. For Frequency Profile

In normal frequency profile for text categorization, our experiment results were below 20% for all predefined category. The figure of 6a illustrates it.

B. For Normalized Frequency Profile

The normalized frequency profile has much better performance than the normal frequency profile. The performance of normalized frequency is shown in Fig. 6b. According to the graph categorization accuracy for grams 2 and 3 are far better than others. The accuracy for grams 3 gets up to 100% for sports category. But entertainment category has very bad performance using the normalized n-gram frequency profile. This is because the entertainment category accumulates many domains of news. As a result the categorization results get fuzzy. Another important aspect of the graph is that for gram 4 the accuracy falls. This reassembles that higher n-grams does not ensure better categorization for Bangla.

C. For Ranked Frequency Profile

For this case ranks different ranks (0, 100, 200, 300, 400, 500, and 1000) were taken for performance analysis.

C.1 Result for rank 0, 100, 200, 300, 400, 500, 1000

Fig. 6c shows the results for rank 0.Here with rank 0 both 2 and 3 length grams have far better performance than other grams. Fig. 6d shows the results for rank 100. Here there was no unigram as the there are less than 100 alphabets in Bangla. But with rank 100 grams having length 2 and 3 has good performance. Again grams with length 4 have bad result. Fig. 6e shows the results for rank 200. Here, 3 length grams have better performance. But for 4 length grams had bad result. Fig. 6f and 6g shows the results for ranks 300 and 400. For rank 300 and 400 the 3 length grams have good performance. Fig. 6h and 6i shows the results for ranks 500 and 1000. For rank 500 and 1000 the 3 length grams have good performance. For 500 and 1000 rank analysis the test cases did not produce such higher ranks bi-grams. But still with these higher rank tri-grams have better results. But one significant fact is that the accuracy of tri-gram fell from 100% to 80% as the ranks were changed from 500 to 1000.

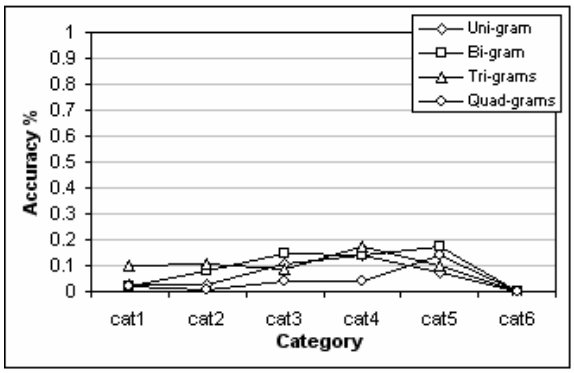

Fig. 6a. Category vs Accuracy for test files with normal frequency profile.

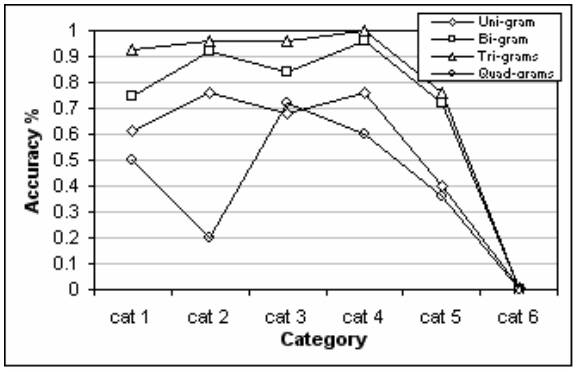

Fig. 6b. Category vs Accuracy for test files with normalized normal frequency profile.

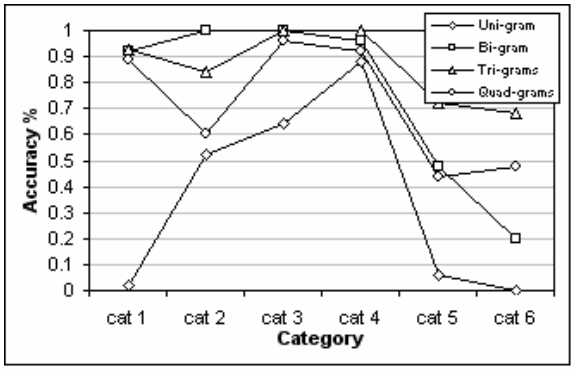

Fig. 6c. Category vs Accuracy for test files with ranked frequency profile taking rank 0.

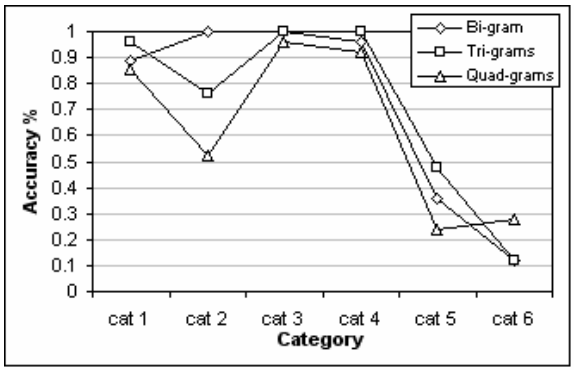

Fig. 6d. Category vs Accuracy for test files with ranked frequency profile taking rank 100.

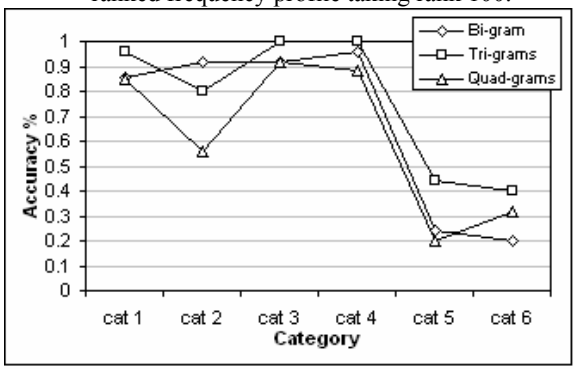

Fig. 6e. Category vs Accuracy for test files with ranked frequency profile taking rank 200.

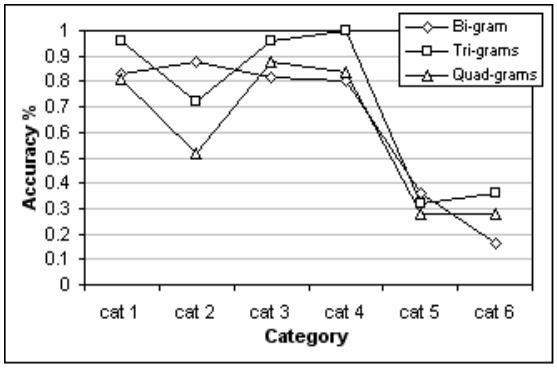

Fig. 6f. Category vs Accuracy for test files with ranked frequency profile taking rank 300.

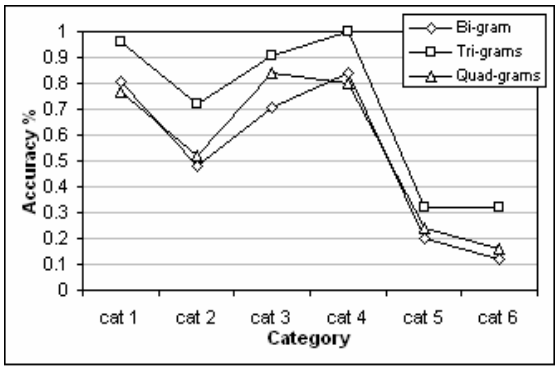

Fig. 6g. Category vs Accuracy for test files with ranked frequency profile taking rank 400

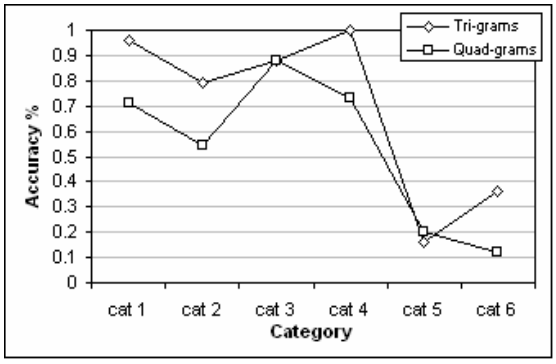

Fig. 6h. Category vs Accuracy for test files with ranked frequency profile taking rank 500.

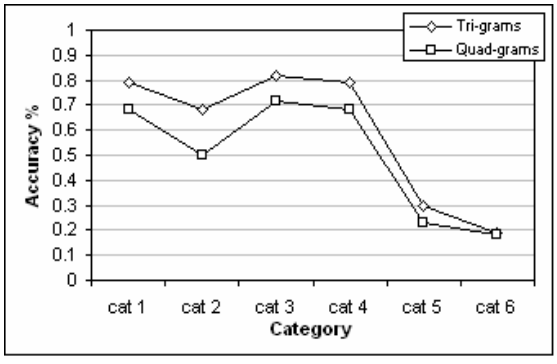

Fig. 6i. Category vs Accuracy for test files with ranked frequency profile taking rank 1000.

V. OBSERVATIONS

Initially performance of text categorization increases with the increase of n (from 1 to 3), but it is not the same as it increases from 3 to 4. This shows that bigger n-grams do not ensure better language modeling in n-gram based text categorization for Bangla. Again character level trigram performs better than any other n-grams. The reason could be that trigram could hold more information for modeling the language. It is an open project for researchers to find the reasoning behind it. This could be a very good research area for both computational linguistics and also for Bangla linguists.

VI. FUTURE WORK

This work was based on Prothom–Alo one year news corpus. So, all the language modeling based on n-grams reflects the Prothom–Alo's style of writing, vocabulary usage, sentence generation etc. By using this training set to categorize other text not related to news can have different result. n-gram based text categorization works well for Bangla but other text categorization techniques should also be tested to have an actual glimpse of which method works well for Bangla.

VII. CONCLUSION

Text Categorization is an active research area in information retrieval. Many methods had been used in English to get better automated categorization performance. n-gram based text categorization is also among the methodologies used in English language for text categorization, having good performance. In this paper we analyzed the efficiency of n-gram based text categorization based on 1 year news corpus of Prothom-Alo. For Bangla, analyzing the efficiency of n-grams shows that tri-grams have much better performance for text categorization for Bangla. It is an open project for researchers to find the reasoning behind it. We also found that Zipf's Law does work for Bangla using character level n-grams, unless the ranked frequency profile could not have better overall performance as the ranks increased.

VII. ACKNOWLEDGEMENT

This work has been supported in part by the PAN Localization Project (www.panl10n.net), grant from the International Development Research Center, Ottawa, Canada, administrated through Center for Research in Urdu Language Processing, National University of Computer and Emerging Sciences, Pakistan.

REFERENCES

[1] Christopher D. Manning and Hinrich schutze, Foundations of Statistical Natural Language Processing, Chapter 16, 1999.

[2] William B. Cavnar and John M. Trenkle, *N-Gram-Based Text Categorization*, In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, 1994.

[3] Fabrizio Sebastiani, Machine Learning in Automated Text Categorisation, ACM Computing Surveys, 1999.

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

[5] Thorsten Joachims, Text Categorization with Support Vector Machines Learning with Many Relevant Features, In The Proceedings of ECML-98, 10th European Conference on Machine Learning, 1997.

[6] Pazzani, M.; Muramatsu, J.; and Billsus, D. Syskill & Webert Identifying interesting web sites. In Proceedings of the Thirteenth National Conference on Artificial Intelligence, 1996.

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

[8] Helmut Berger and Dieter Merkl, A Comparison of Text-Categorization Methods Applied to NGram Frequency Statistics, InAustralian Joint Conference on Artificial Intelligence, 2004.

[9] Youngjoong Ko and Jungyun Seo, Text categorization using feature projections, Proceedings of the 19th international conference on Computational linguistics, 2002.

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

[11] Markus Forsberg and Kenneth Wilhelmsson, Automatic Text Classification with BayesianLearning, http://www.cs.chalmers.se/~markus/LangClass/LangClass.pdf

[12] Raymond J. Mooney, Paul N. Bennett, and Loriene Roy , Book Recommending Using Text Categorization with Extracted Information, In the AAAI-98/ICML-98 Workshop on Learning for Text Categorization and the AAAI-98 Workshop on Recommender Systems, 1998.

[13] Peter Náther, N-gram based Text Categorization, Institute of Informatics, Comenius University, 2005.Unpbulished.

[14] Bangladeshi Newspaper, Prothom-Alo. Online version available online at http //www.prothom-alo.net/

[15] Ciya Liao, Shamim Alpha and Paul Dixon, Feature Preparation in Text Categorization, Oracle Corporation, http://www.oracle.com/technology/products/text/pdf/feature_preparation.pdf