# 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 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.*

## 1. 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, is the process of automatically assigning given text into a set of predefined categories based one its content. Typical text classification systems use a range of statistical and machine learning techniques based on regression model, K-Nearest Neighbor (KNN) [10], Decision Tree, Naïve Bayes [4], [11], Support Vector Machines (SVM) [5], n-gram based [2], [7], [13], and so on.. In this paper, we analyze the performance of n-gram based text categorization technique for Bangla.

## 2. N-gram based text categorization

### 2.1. What are n-grams?

An n-gram is a sub-sequence of n-items in any given sequence, where the sequence items or "grams" can be anything, from characters to words. 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" contains the character level n-grams shown in Table 1.

**Table 1: Different n-grams for the word "□□□□□" (spaces are shown with '_')**

| | □□□□□ |
|---|---|
| Unigrams | ব , □, □, ল , □, _ |
| Bi-grams | _ব , □□, □□, □ল , □□, □_ |
| Tri-grams | _□□, □□□, □□ল , □□□ ,□□_ |
| Quad-grams | _□□□, □□□ল , □□□□, □□□_ |

So, a character-level n-gram is simply a character sequence of length *n*, i.e., an n-character slice of a longer string, extracted from a text [2]. Consequently, a word, which includes the leading and trailing spaces as well, is then represented as a sequence of overlapping n-grams [13]. The value of *n* is typically fixed for a particular corpus.

### 2.2. Why n-gram based text categorization?

The experience with natural languages that some words occur more frequently than others is formally expressed by what is known as Zipf's Law. In [2], Cavnar and Trenkle summarize Zipf's Law as *"The $n^{th}$ most common word in a human language text occurs with a frequency inversely proportional to n"*. That is, $f \propto \frac{1}{r}$, where *f* is the frequency of the word and *r* is the rank of the word in the list ordered by the frequency [13]. There are several implications of Zipf's Law. The first is that a relatively small set of words occur far more frequently than the rest of the words in a language. The inverse relationship implies that any classification algorithm using n-gram frequency statistics is not overly sensitive to limiting the n-grams below a particular rank. And, that texts of the same category should have similar n-gram frequency

profiles. One important benefit of using n-grams is to achieve language and domain independence, which is not trivial with most word-based information retrieval systems, which tend to use language specific stemming and stop list processing [16].

### 2.3. Why Character Level n-gram?

For n-gram based text classification to be effective, the various inflected forms of a root word should somehow "resolve" as being related to the same word. It turns out that the character-level n-grams of different morphological variations of a word tend to produce many of the same n-grams. This allows the information retrieval systems to collect the different forms of the same word by using the n-grams of one of the forms of the word as the key. Another advantage is the sliding window approach of character-level n-grams, which allows the model to capture the context across word boundaries as well. 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).

## 3. 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

### 3.1. Feature construction and feature selection

A Classifier cannot directly interpret a text, so the raw text must first be mapped into a compact representation. The choice of the representation however varies across applications, and depends on what one considers the meaningful units of texts are. [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, the different lengths of n-grams are used as features. The document is first mapped onto a feature vector. The feature vector has an associated set of attributes, one for each term that occurs in the training corpus. The attributes value is set to the frequency with which a term occurs in a particular document. 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 its value is the number of times the term occurs in the document. 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

### 3.2. 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. Table 2 shows the predefined categories and the corresponding news editorials taken from Prothom-Alo.

**Table 2: 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 | □□□□□□□□□□ □□□□□□□□□ , □□□□□□□ ডট কম |
| Cat6 | Entertainment | বিনোদন |

### 3.3. Generating n-gram profiles

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

**3.3.1. 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 are computing using a sliding window which moves forward *n* characters at a time.

**3.3.2. 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 is used to basically maintain a frequency count of each n-gram found in the text.

**3.3.3. Creation of different document representation.** After extracting the n-grams from a text, we create three different has maps representing the different frequency profiles:

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

**3.3.4. 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, □□ = 50
**Document Representation:**
d = (150, 75, 50).

**Figure 1: Normal Frequency profile generation**

**3.3.5. CNormalized frequency profile.** To generate the normalized frequency profile the previously generated normal frequency profile hash map is used. For this case each occurrence of an 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, □□ = 50
**Reverse Order Rank:**
নব = 1
এত = 2
□□ = 3
**Document Representation:**
d = (1, 2, 3).

**Figure 2: Normalized Frequency profile generation**

This normalized frequency profile uses the relative frequencies instead of the absolute number of occurrences of the n-grams. The rationale behind the normalization is to remove the effect of the length of the text. Most of the frequencies would of course be zero or very small because most n-grams would rarely, if ever, occur in a text.

**3.3.6. 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, □□ = 50
150 +75+50 = 275
**Normalized frequency:**
নব =0.54, এত =0.27, □□ = 0.19
**Document Representation:**
d = (0.54, .27, 0.19).

Figure 3: Ranked frequency profile generation

## 3.4. Comparing and ranking n-gram profiles

We begin by creating the n-gram frequency profile to represent the set of predefined categories., using the testing corpus. Now, to assign a given text a category from this set, its n-gram frequency profile is computed. The profile is then compared against the pre-computed profiles of the predefined categories using the "profile distance" metric. Figure 4 shows the comparison process, and Figure 5 shows an example of how to compute the distance between two ranked frequency profiles. In Fig. 5, □□ bi-gram has its rank same for both the category and the test documents

profile, producing a 0 distance; but for the case of এত the category profile has it on third position where as in test profile it is ranked as fifth, producing a distance of 5-3=2. The final distance is the sum of all the individual n-gram distances, and the text is classified as one of the predefined categories with the smallest distance from the text.

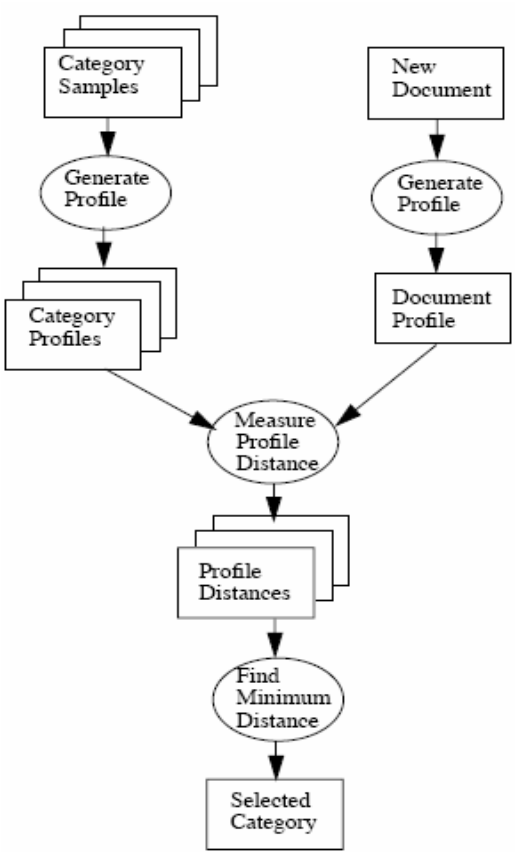


**Figure 4: Classification Procedure**

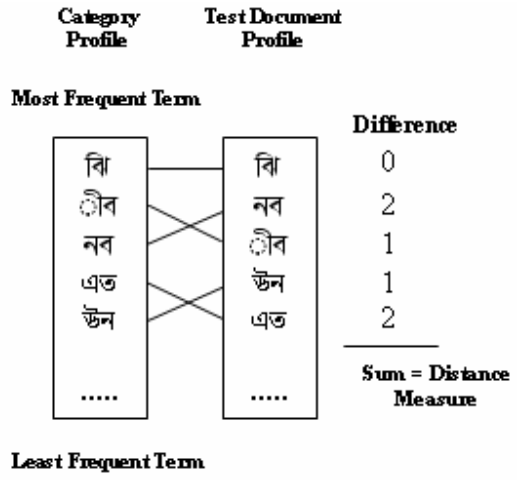


Figure 5: Measure profile distance

### 3.5. 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.

## 4. 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.

### 4.1. 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.

### 4.2. 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 Figure 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.

### 4.3. For ranked frequency profile

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

**4.3.1. 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.

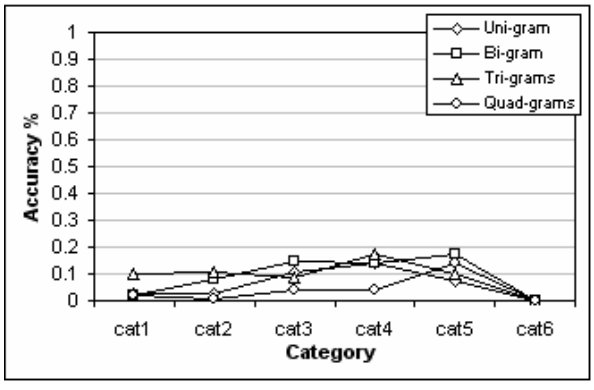


**Figure 6a: Category vs Accuracy for test files with normal frequency profile**

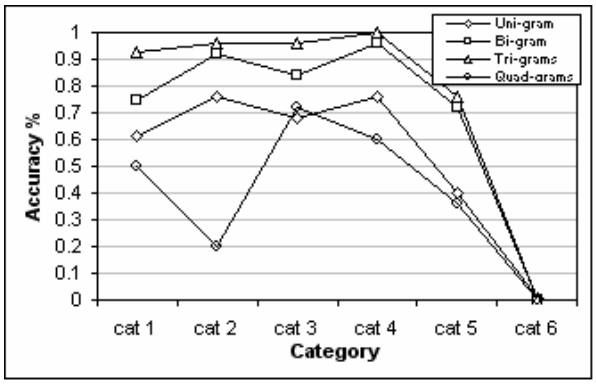


**Figure 6b: Category vs Accuracy for test files with normalized normal frequency profile**

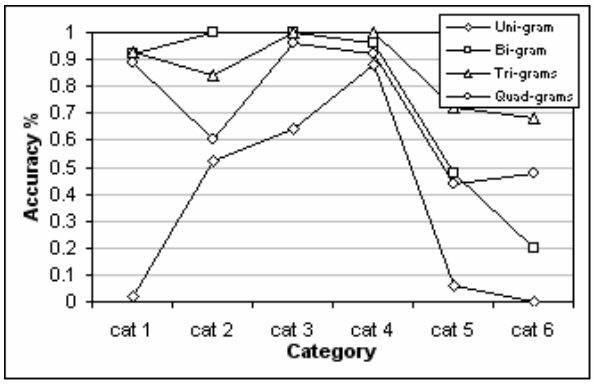


**Figure 6c: Category vs Accuracy for test files with ranked frequency profile taking rank 0**

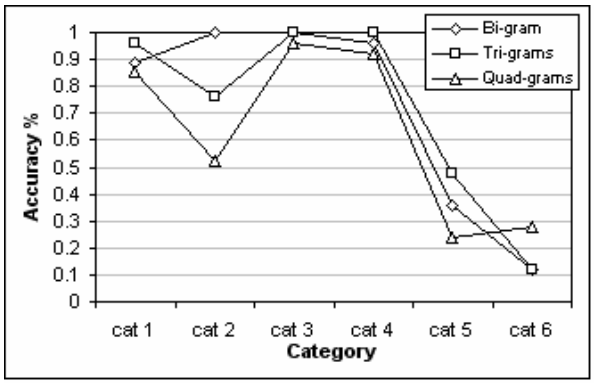


**Figure 6d: Category vs Accuracy for test files with ranked frequency profile taking rank 100**

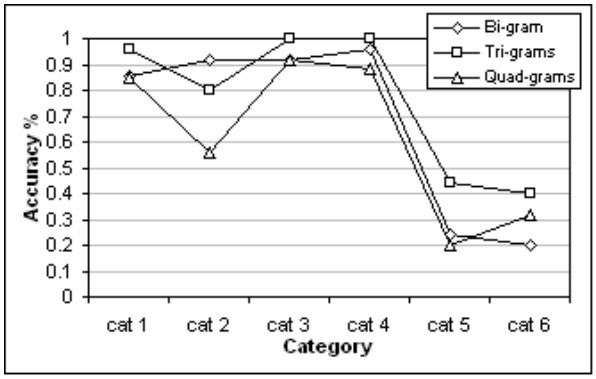


**Figure 6e: Category vs Accuracy for test files with ranked frequency profile taking rank 200**

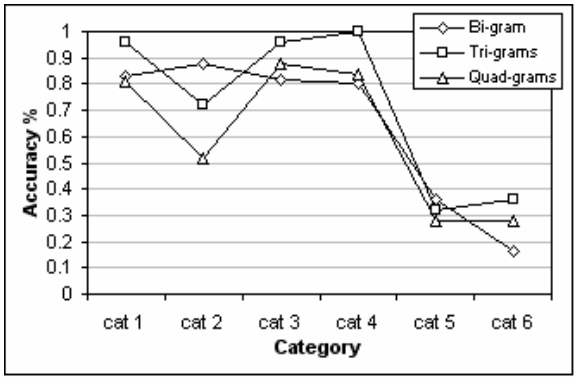


**Figure 6f: Category vs Accuracy for test files with ranked frequency profile taking rank 300.**

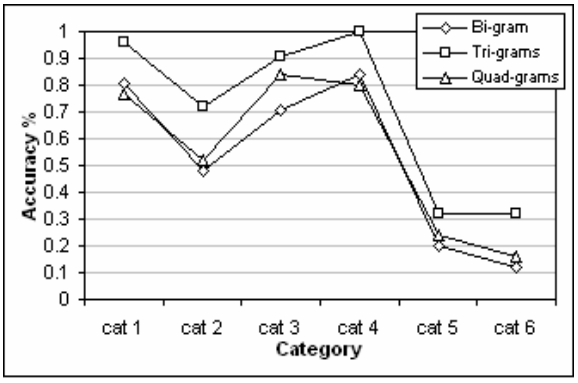


**Figure 6g: Category vs Accuracy for test files with ranked frequency profile taking rank 400**

.

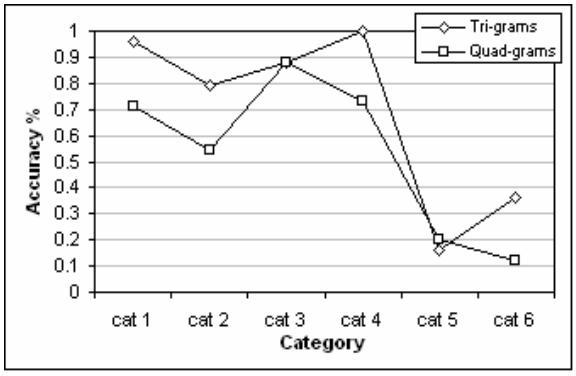


**Figure 6h: Category vs Accuracy for test files with ranked frequency profile taking rank 500**

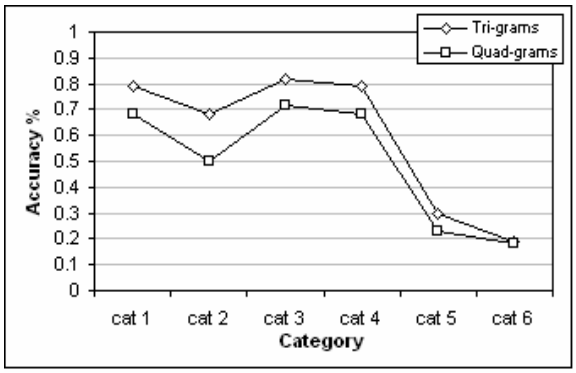


**Figure 6i: Category vs Accuracy for test files with ranked frequency profile taking rank 1000**

## 5. 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.

## 6. 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.

## 7. 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 evaluate the n-gram based text categorization scheme using a year's text from of the Prothom-Alo newspaper. 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.

## 8. 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.

## 8. References

[1] C.D. Manning and H. Schutze, *Foundations of Statistical Natural Language Processing*, Chapter 16, 1999.

[2] 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.

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

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

[7] J.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] H. Berger and D. Merkl, "A Comparison of Text-Categorization Methods Applied to NGram Frequency Statistics", In *Australian Joint Conference on Artificial Intelligence*, 2004.

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

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

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

[12] R.J. Mooney, P.N. Bennett, and L. 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] P. Náther, "N-gram based Text Categorization, Institute of Informatics", Comenius University, 2005, unpublished.

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

[15] C. Liao, S. Alpha and P. Dixon, “Feature Preparation in Text Categorization”, Oracle Corporation, http://www.oracle.com/technology/products/text/pdf/feature_preparation.pdf

[16] E. Miller, D. Shen, J. Liu and C. Nicholas, “Performance and Scalability of a Large-Scale N-gram Based Information Retrieval System”, Journal of Digital Information, 2000.