

ANALYSIS OF N-GRAM BASED TEXT
CATEGORIZATION FOR BANGLA IN A NEWSPAPER
CORPUS.

Munirul Mansur
Student ID: 02101043

Department of Computer Science and Engineering
August 2006



BRAC University, Dhaka, Bangladesh

DECLARATION

In accordance with the requirements of the degree of Bachelor of Computer Science and Engineering in the division of Computer Science and Engineering, I present the following thesis entitled "*Analysis of N-gram based text categorization for Bangla in a newspaper corpus*". This work was performed under the supervision of Dr. Mumit Khan.

I hereby declare that the work submitted in this thesis is our own and based on the results found by our self. This thesis, neither in whole nor in part, has been previously submitted for any degree.

Signature of
Supervisor

Signature of
Author

ABSTRACT

The goal of any classification is to build a set of models that can correctly predict the class of different objects. Text categorization is one such application and can be used in many classification task, e.g. news categorization, language identification, authorship attribution, text genre categorization, recommendation systems etc. In this paper we analyze the performance of n-gram based text categorization for Bangla in a Bangladeshi newspaper, Prothom-Alo corpus.

ACKNOWLEDGEMENT

My great thanks to my supervisor, Dr. Mumit Khan, for showing special patience as I pursued my thesis. He enthusiastically supported me to do this work. He always gave suggestions and constructive criticism in the development of this thesis.

Table of Contents

Topic		
Chapter I: Introduction	2
Chapter II: N-gram based Text Categorization	3
2.1 Why N-gram Based Text Categorization?	4
2.2 Reasoning Behind Selecting N-gram	5
2.3 Why Character Level N-gram ?	6
Chapter 3: Methodology	8
3.1 Document Representation	9
3.2 Learning Phase	10
3.3 Generating N-gram Profiles:	11
3.3.1 Preconditioning is performed on the text:	11
3.3.2 The text is converted to n-grams:	11
3.3.3 Each n-gram is assigned a hash key	11
3.3.4 The n-grams are inserted into a hash map	12
3.3.5 Creation of different hash maps	12
3.3.5.1 Normal Frequency Profile	12
3.3.5.2 Normalized Frequency Profile	13
3.3.5.3 Ranked Frequency Profile	13
3.4 Comparing N-gram Profiles:	14
3.5 Classification Of Text	16
Chapter IV: Results	17
4.1 Test Data:	18
4.2 Results for Frequency Profile	18
4.3 Results for Normalized Frequency Profile	19
4.4 Results for Ranked Frequency Profile	19
4.4.1 Results for rank 0	20
4.4.2 Results for rank 100	20
4.4.3 Results for rank 200	21
4.4.4 Results for rank 300 and 400	22
4.4.5 Results for rank 500 and 1000	23
Chapter V: Observation		25
Chapter VI: Future works	27
Chapter VII: Conclusion	29
Chapter VIII: References	31

List of Tables

Name of the tables		
<i>Table 1: Different n-grams for the word "বাংলা".</i>	4
Table 2: List of predefined categories and their content source.	10

List of Figures

Name of the Figures		
Figure1: Normal Frequency profile generation	12
Figure 2: Normalized Frequency profile generation	13
Figure 3: Ranked Frequency profile generation	14
Figure 4: Classification Procedure	15
Figure 5: Measure profile distance	15
Figure 6: Category vs Accuracy for test files with normal frequency profile	18
Figure 7: Category vs Accuracy for test files with normalized normal frequency profile	19
Figure 8: Category vs Accuracy for test files with ranked frequency profile taking rank 0.	20
Figure 9: Category vs Accuracy for test files with ranked frequency profile taking rank 100.	20
Figure 10: Category vs Accuracy for test files with ranked frequency profile taking rank 200.	21
Figure 11: Category vs Accuracy for test files with ranked frequency profile taking rank 300	22
Figure 12: Category vs Accuracy for test files with ranked frequency profile taking rank 400.	22
Figure 13: Category vs Accuracy for test files with ranked frequency profile taking rank 500.	23
Figure 14: Category vs Accuracy for test files with ranked frequency profile taking rank 1000.	23

**CHAPTER 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 [9], decision tree, Naïve Bayes [4] [10], Support Vector Machines [5], using n-grams [2], [13] and so on [3].

Such techniques are currently being applied in many areas of English like language identification, authorship attribution, text genre categorization, news categorization [2], recommendation systems [4] [6] [7] [11], Spam filtering [8] etc.

In this thesis we experimented whether n-gram based text categorization works for Bangla and analyze its performance. In the following section we will describe elaborately the reasoning behind why selecting n-gram for text categorization , how they are used, what results we got, our observations and future works that are to be done.

**CHAPTER II:
N-GRAM BASED TEXT CATEGORIZATION**

2.1 What is a N-gram?

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 sub-sequence 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 1: Different n-grams for the word "বাংলা". (leading and trailing spaces were considered as the part of the word, which is 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. Typically, n is fixed for a particular corpus of documents and the queries made against that corpus where corpus is a huge text. To generate the n-gram vector for a document, a window of characters in length is moved through the text, sliding forward by a fixed number of characters (usually one) at a time. At each position of the window, the sequence of characters in the window is recorded.

2.2 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."*

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:

$$f = k/r$$

(1.1)

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. Furthermore, there is a smooth continuum of dominance from most frequent to least which is true for the frequency of occurrence of N-grams, both as inflection forms and as morpheme-like word components which carry meaning. 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. We have built an experimental text categorization system that uses this idea.

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. Furthermore, one cannot assume that words are always separated by spaces. For example, in some Asian languages, different words are not separated by spaces, so a sentence is composed of many consecutive characters. Grammar knowledge is needed to separate those characters into words,

which is a very difficult task to perform. By using n-grams, the system does not need to separate characters into words. [13]

2.3 Why Character Level N-gram?

Many current classifiers are based on keyword (that are usually single words) extracted from the documents. In many of the classification approaches, a keyword is assumed to be a unique representative of a distinctive concept or semantic unit. However, the reality is different: A word may represent several different meanings, and different words may refer to the same meaning. These are the problems of polysemy and synonymy. For example, the word bank can be a section of computer memory, a financial institution, a steep slope, a collection of some sort, an airplane maneuver, or even a billiard shot. It is hard to distinguish those meanings automatically. Similarly, in medical literature, myocardial infarction has the same meaning as minor heart attack.

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 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.”

Words are the constituent part of a document. Now if a specific word occurs in a document more frequently then the its constituent character level representations will also occurs frequently. That is by doing n-gram modeling on the character level we implicitly model a document in its word level representation.

This thesis is based on the work of [2] and [13] those worked on N-gram based text categorization on a computer newsgroup categorization task. We employed the same technique and tried to analyze whether this technique

works for Bangla news corpus and how well it performs. In this work N-grams with various lengths were used (from 2 to 4-grams).

**CHAPTER III:
METHODOLOGY**

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

- Document Representation
- Learning phase

3.1 Document Representation

An important task in text categorization is to prepare text in such a way, that it becomes suitable for text classifier, i.e. transform them into an adequate document representation. Typically, text documents are unstructured data. Before learning process, we must transform them into a representation that is suitable for computing.

One requirement for text categorization algorithms is that the training data must be represented as a set of feature vectors. The problem raised when using a feature vector representation is to answer the question, “**What is a feature?**” In text categorization, 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. A straightforward approach for representing text as feature vectors is the set-of-words approach: A document is represented by a feature vector that contains one attribute for each word that occurs in the training collection of documents. If a word occurs in a particular training document, its corresponding attribute is set to its frequency or some other related feature. Thus, each document is represented by the set of words it consists of.

For this work training documents or the category files has three n-gram based document representations:

- Frequency profile
- Normalized frequency profile

- Ranked frequency profile

3.2 Learning Phase

In the learning phase the classifier is trained with pre classified documents.

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 2:

List of pre defined categories and their content source.

Defined category	Prothom-Alo Editorials
Technology News Category	কম্পিউটার প্রতিদিন, প্রজন্ম ডট কম
Sports News Category	খেলা
Deshi* News Category	বিশাল বাংলা
International News Category	সারা বিশ্ব
Entertainment News Category	বিনোদন
Business News Category	অর্থ ও বাণিজ্য

* Deshi news category contains regional news of Bangladesh.

3.3 Generating N-gram Profiles:

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

3.3.1 Preconditioning is performed on the text:

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. Bangla has the advantage of not having uppercase and lowercase letters so no more precondition were done on the texts.

3.3.2 The text is converted to n-grams:

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 one character forward at a time.

3.3.3 Each n-gram is assigned a hash key

Every possible n-gram is given a number, a so called hash key. These hash keys are stored in an 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 the n-gram is added to the hash map. This whole process of assigning hash key is handled internally by Java provided utility package. The number of occurrence of the n-gram may be a huge number, for that double value is used as the value to a corresponding hash key in the hash map.

3.3.4 The n-grams are inserted into a hash map

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. An n-gram is placed in the hash map. Note

that the size of the hash map is unaffected by the size of the text being analyzed.

3.3.5 Creation of different hash maps

When all n-grams have been extracted from a text then they are put into three hash maps:

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

3.3.5.1 Normal Frequency Profile

This hash map just contains occurrences of the n-grams in the given text. This is 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:

<p>নব = 150 , এত = 75 , ীব = 50</p> <p>Document Representation: $d = (150, 75, 50)$.</p>
--

Fig1: Frequency profile generation

Where the documents representation records that নব, এত, ীব have frequency of 150, 75 and 50.

3.3.5.2 Normalized Frequency Profile

To generate the normalized frequency profile the previously generated normal frequency profile hash map is used. For this case each frequency 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:

$$\begin{aligned} \text{নব} &= 150, \text{এত} = 75, \text{ীব} = 50 \\ 150 + 75 + 50 &= 275 \\ \text{Normalized frequency: } \text{নব} &= 0.54, \text{এত} = 0.27, \text{ীব} = 0.19 \\ \text{Document Representation: } d &= (0.54, 0.27, 0.19). \end{aligned}$$

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.

3.3.5.3 Ranked Frequency Profile

The last hash map is the reflection of Zip's Law for the corresponding document. 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.

$$\begin{aligned} \text{নব} &= 150, \text{এত} = 75, \text{ীব} = 50 \\ \text{Reverse Order Rank:} \\ \text{নব} &= 1 \\ \text{এত} &= 2 \\ \text{ীব} &= 3 \\ \text{Document Representation: } d &= (1. 2. 3). \end{aligned}$$

Fig 3: Ranked Frequency profile generation

In the figure above, নব is the most frequent having the frequency value 150 so it gets rank 1. Similarly, এত and ীব gets their rank according their frequency.

Basically 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.

3.4 Comparing N-gram Profiles:

For comparing, we start with a set of pre-existing text categories. From these, we would generate a set of different 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 following figure.

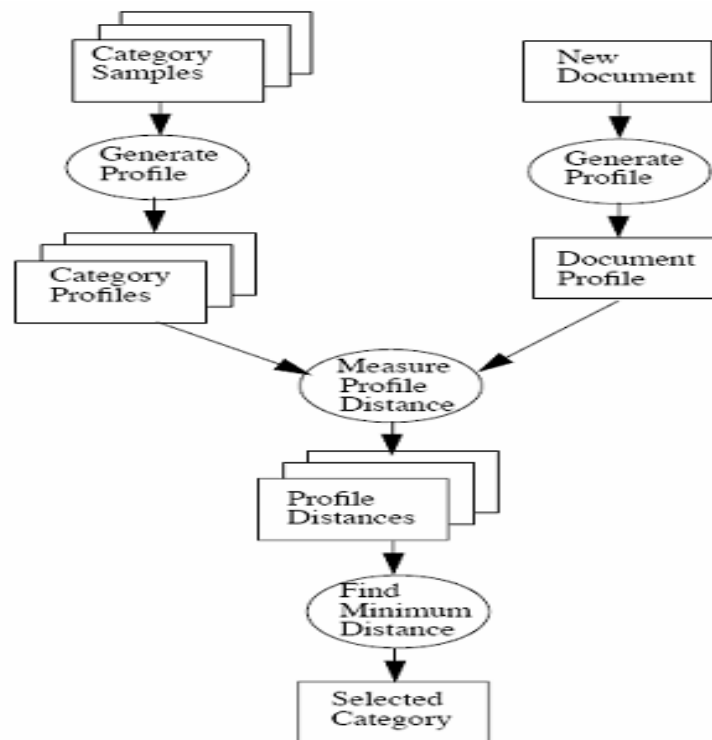


Fig 4: Classification Procedure

Measuring 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.

The following figure shows an example of how this measuring distance profile is done while working with ranked frequency profile.

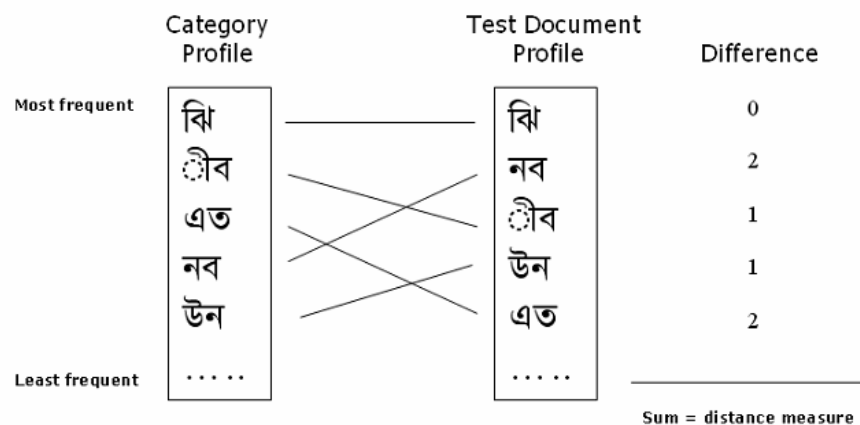


Fig 5: Measure profile distance

Here, বি 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.

3.5 Classification Of Text

When a category for a given document is to be decided, distances are counted from all the categories profiles. As we have the list of distances from all categories, we can order them and then we can choose most relevant categories for the given document. In this thesis, we used only the least distance category as the winner.

CHAPTER IV: RESULTS

4.1 Test Data:

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.

4.2 Results for Frequency Profile

In normal frequency profile for text categorization, our experimented resulted below 20% for all predefined category.

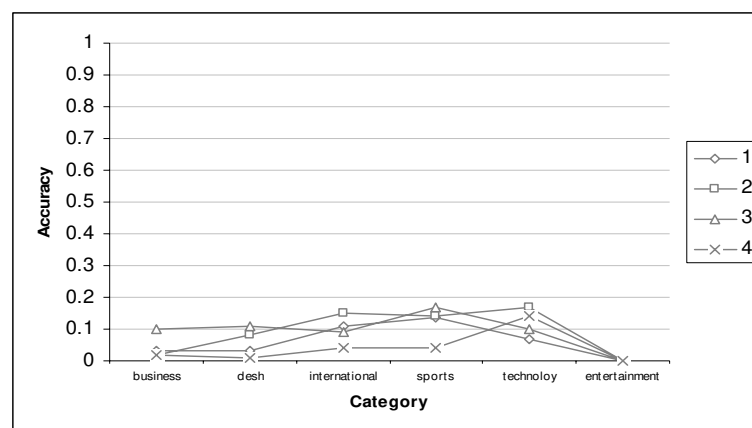


Fig 6: Category vs Accuracy for test files with normal frequency profile

In the above graph, the X axis contains the predefined categories name and Y axis shows the accuracy level of the classifier when test documents are given for text categorization purpose. The numerical values of 1 to 4 on the right side of the graph represent the gram value used as n-grams.

4.3 Results for Normalized Frequency Profile

The normalized frequency profile has much better performance than the normal frequency profile. The performance of normalized frequency is given below:

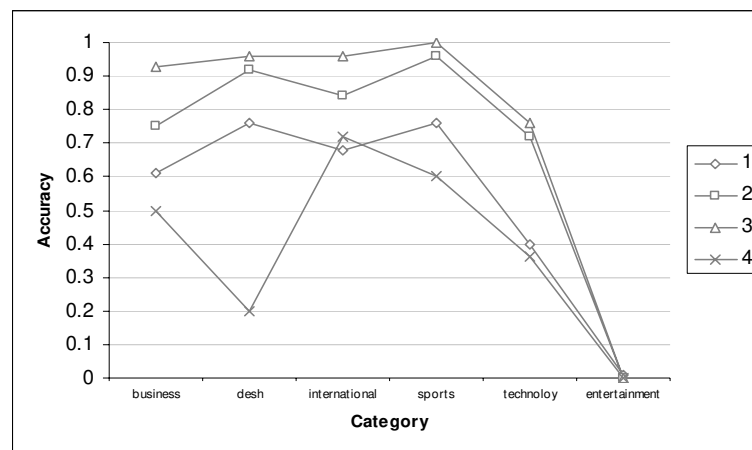


Fig 7: Category vs Accuracy for test files with normalized normal frequency profile

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 work well for categorization purpose that is the information provided by the 4 character slice grams influences less for Bangla text categorization.

4.4 Results for Ranked Frequency Profile

For ranked frequency profile different ranks (0, 100, 200, 300, 400, 500, 1000) were taken for performance analysis.

4.4.1 Results for rank 0

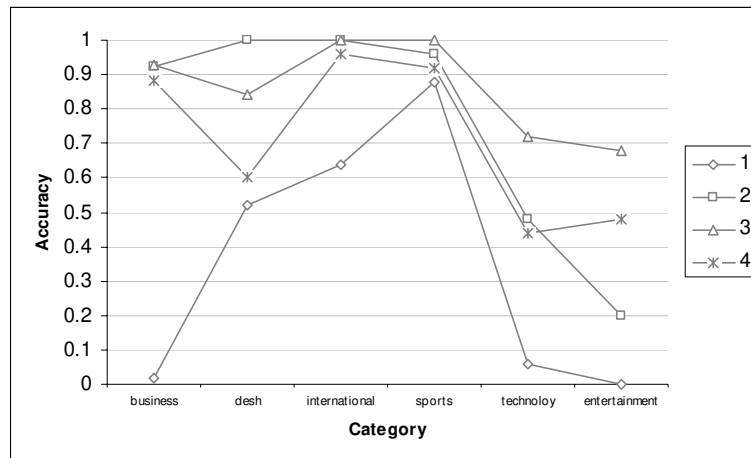


Fig 8: Category vs Accuracy for test files with ranked frequency profile taking rank 0.

Here with rank 0 both 2 and 3 length grams have far better performance than other grams. Here for the sports category the classifier has the best result for all the grams from 1 to 4. The entertainment news category gets much better accuracy results than using normalized frequency profile document representation method.

4.4.2 Results for rank 100

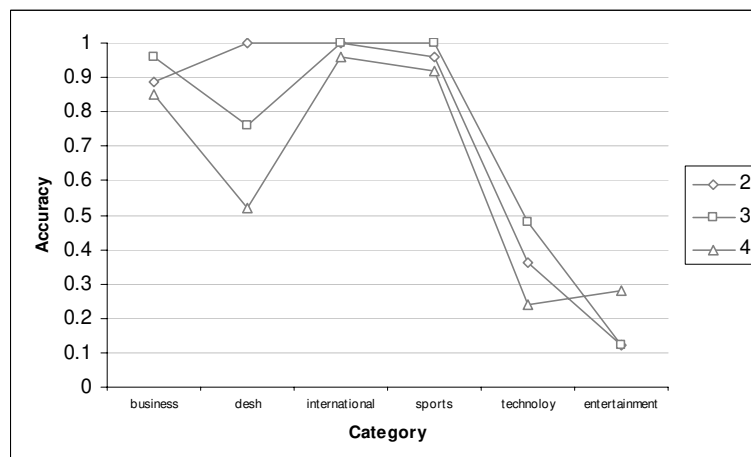


Figure 9: Category vs Accuracy for test files with ranked frequency profile taking rank 100.

Here there was no unigram as 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. The accuracy of the classifier for deshi category falls from 100% to little above 50% when gram value is change from 2 to 4. Again using the tri-grams the classifier shows an excellent performance for deshi, international and sports news category. The accuracy of the business news category falls as rank value changes from 0 to 100.

4.4.3 Results for rank 200

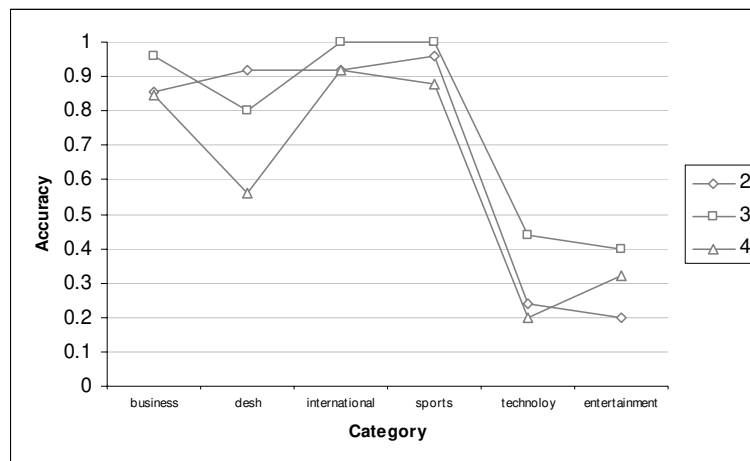


Fig 10: Category vs Accuracy for test files with ranked frequency profile taking rank 200.

Here, 3 length grams have better performance. But for gram length of 4 has bad result for all the categories.

4.4.4 Results for rank 300 and 400

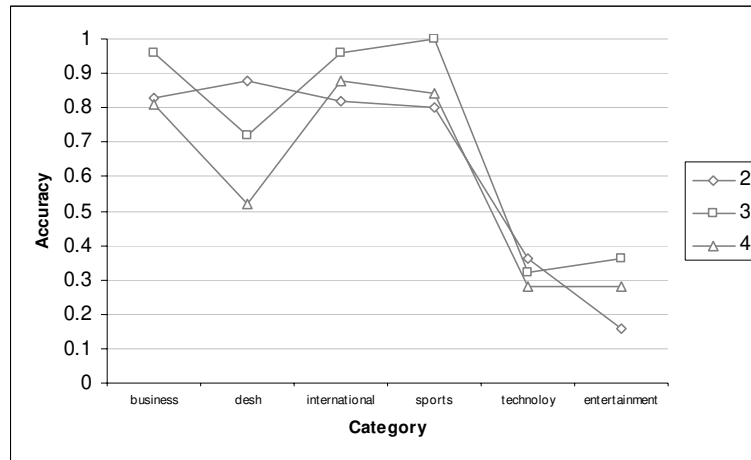


Fig 11: Category vs Accuracy for test files with ranked frequency profile taking rank 300.

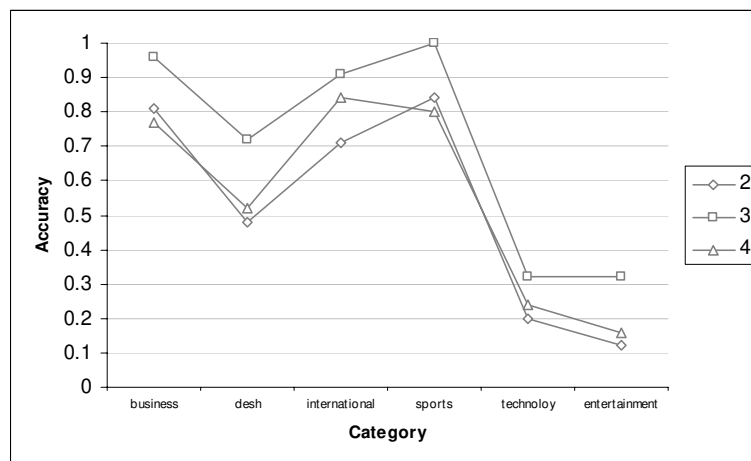


Figure 12: Category vs Accuracy for test files with ranked frequency profile taking rank 400.

For rank 300 and 400 the 3 length grams have good performance than other grams though it gets 100% accuracy for sports news category.

4.4.5 Results for rank 500 and 1000

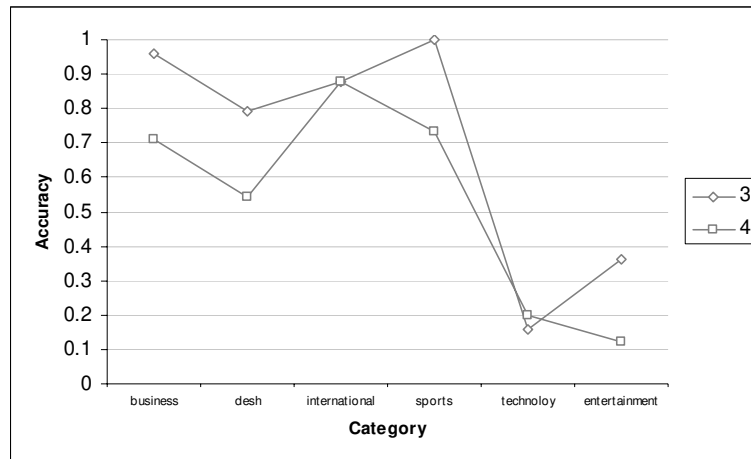


Figure 13: Category vs Accuracy for test files with ranked frequency profile taking rank 500.

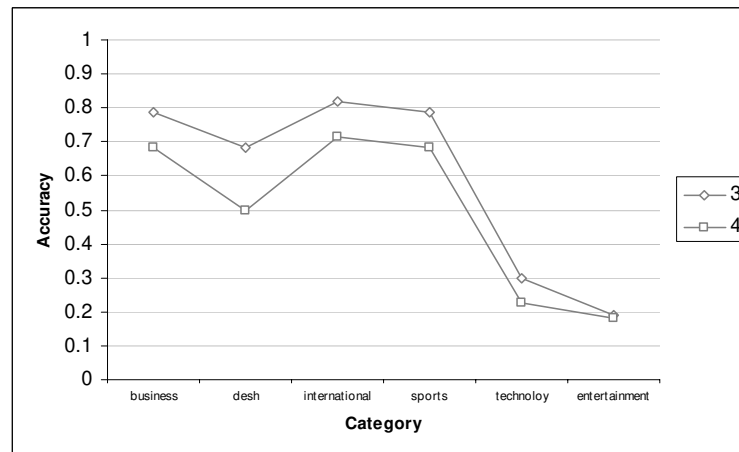


Fig 14: Category vs Accuracy for test files with ranked frequency profile taking rank 1000.

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.

CHPATER V
OBSERVATION

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.

In our experiment we have also seen that character level trigram perform 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.

The rank analysis showed that taking very high (rank 1000) or very low (rank 0) does not give good results. But a rank 100, 200, 300 has good results. The reason is that very high ranked n -grams are very infrequent in a document.

CHAPTER VI
FUTURE WORKS

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.

Again the categories created for the learning phase had different size texts. Text categorization is based on data, the more domains specific the training data will be, the more effective the categorization performance will be. So for better categorization the domain specific categories need to be created.

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.

CHAPTER 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 increase ed.

CHAPTER VIII
REFERENCE

Books:

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

Articles from published conference proceedings:

- [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] Lang, K. NewsWeeder: Learning to filter netnews. In Proceedings of the Twelfth International Conference on Machine Learning, 1995.
- [8] Helmut Berger and Dieter Merkl, A Comparison of Text-Categorization Methods Applied to N-gram Frequency Statistics, In Australian Joint Conference on Artificial Intelligence, 2004

- [9] Johannes Fürnkranz, A Study Using n-gram Features for Text Categorization, <http://citeseer.ist.psu.edu/johannes98study.html> , 1998

Technical Reports:

- [10] Markus Forsberg and Kenneth Wilhelmsson, Automatic Text Classification with Bayesian Learning, <http://www.cs.chalmers.se/~markus/LangClass/LangClass.pdf>
- [11] 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
- [12] Markus Forsberg and Kenneth Wilhelmsson, Automatic Text Classification with Bayesian Learning, <http://www.cs.chalmers.se/~markus/LangClass/LangClass.pdf>

Theses:

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

Other:

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