Acknowledgment

At first I like to thank almighty Allah who gives me ability to perform the Thesis. Then I like to give many thanks to my thesis supervisor Md. Nesarul Hoque, Lecturer, Department of Computer Science and Engineering, who encouraged, supervised and supplied necessary requirements and guideline in performing this work. His inspiration for doing research on Bangla Document Ranking by using Term Frequency and Cosine Similarity let me capable to complete the Thesis. I am grateful to him for his unvarying encouragement and simulating ideas. I also like to give thanks to all my teachers and also that of my friends who gave me advice for the completeness of the thesis. Finally I like to special thank to my parents and all my well wishers for their encouragement and support along my study life.

Shraboni Afroz Samapti

Abstract

Bengali is one of the ten most spoken languages in the world, with almost 200 million speakers. Growing online resources reveal a clear need for Bengali language applications and retrieval systems. The development of the internet, there are huge number of Bangla news articles are published every day on the web from different sources and this amount is growing rapidly day by day. Therefore, people have not enough time to read each document with a specified time limit. Bangla document ranking handles this difficulty with an efficient manner. Although, there are more works have been done in English and other European languages, but there is no contribution are present in bangla language to rank a document. In our thesis, we rank bangla document using term frequency and cosine similarity. We take document as vector and query as vector. We measure term frequency each word in each document. Then finding out score using cosine similarity. Then sorted the scores and ranked the documents. We demonstrate the effectiveness of the technique for bangla document ranking.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Introduction

In a real retrieval application (e.g.,Web search), the retrieval results using the initial query given by the user may not be satisfactory to the user; often, the user would need to revise the query to improve the retrieval/ranking accuracy. For some information seeking activities, the user may modify his query several times for one information need. In such an interactive retrieval scenario, the information available to us is more than just the current user query and the collection of documents. This paper summaries a set of experiments with term weighting for documents, using the measurement of term importance within an entire document collection. Then taking query it finds the relevant documents, if it exists then score the documents and sorting the scores it finds out the ranked documents. It is efficient while searching any document suppose in google users write the word or sentence that is called query, our proposed process will compute the importance of word and rank the documents sequentially which were in maximum times in documents.

1.2 Background and Present State of the Problem

Bangla document ranking is an important aspect but no such research work had done on bangla language. Various methods and techniques have implied to rank documents of different country of different languages. In reference [1] the importance of a term within the entire document collection has found, then number of term matches between a query and a document, they used two functions 1)the raw fre-

quency and 2)the log2 of frequency. Then doing normalization for document length they combined the measures and rank the documents. In reference [2], they used term frequency and inverse term frequency, they compute the document ranking performance using precision and recall. In reference [3], they used MMR method for ranking documents. I have mentioned earlier, no research work yet not performed on bangle. As there are several techniques for document ranking is present, these have their own limitations. One of the big problem of these systems is the accuracy is not better. If we need to implement these systems in real time purpose, the accuracy should increase otherwise it will be inefficient. So, I tried to increase the accuracy.

1.3 Motivation and Aims

Today the field of natural language processing(NLP) is increasing. As a nation, we have the historical background of language movement, which reminds us that everyone has the right to communicate using their own language. The growth in electronically available documents makes research and applications in automatic document ranking more and more important. Huge number of available documents in digital media makes it difficult to obtain the necessary information related to the needs of a user. In order to solve this issue, document ranking system can be used. Many work have done in various languages in different countries but there is no previous works in bangla on the basis of document ranking. It encourages me to do the work on bangla document ranking. Now my aim is to identify the document on the basis of importance of words in bangla so that while doing web search user can see the ranked document corresponding their given query. By using the ranking process, a user can decide if a document is related to his/her needs without reading whole document. Document ranking systems can be categorized as extractive or abstractive according to the way the ranking is created. In extractive ranking approaches, the goals are identifying most important concepts in the input and giving relevant documents found in the document set as an output. In abstractive ranking approaches, first the system understands the texts and then it creates

ranking with its own words. The abstractive ranking is similar to the way a person creates ranking. Abstractive ranking remains as a difficult task in natural language processing.

1.4 Objectives and Specific aims

From the above discussion it is found that bangla document ranking is an important thing in our country while reading any bangla documents or newspapers. The objectives of my research are to identify the relevant documents within the given user query. And the specific aim is to rank relevant document with corresponding query. The query is matched word by word in relevant documents and ranked documents with cosine similarity.

1.5 What is NLP & Document Ranking?

Natural Language Processing, NLP is a branch of artificial intelligence that deals with analyzing, understanding and generating the languages that humans use naturally in order to interface with computers in both written and spoken contexts using natural human languages instead of computer languages.

1.5.1 Document Ranking

Document ranking is the process of ranking documents with document and query.

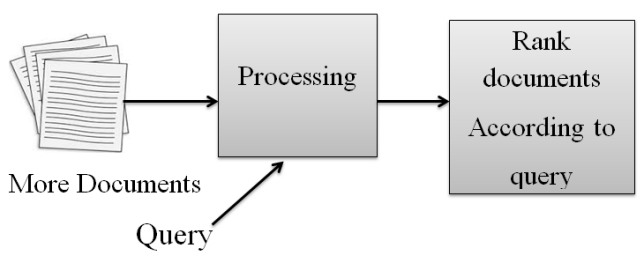

Figure 1.1: **Main Process of our system**

1.6 Organization of the Thesis

The dissertation is organized as follows:

- **Chapter 1 Introduction.** In this chapter we have already discussed bangla document ranking, previous works in this field, motivation , objectives and aim of our thesis. Actually it is the preliminary chapter which describes the basic idea about the project.
- **Chapter 2 Literature Review.** It contains the sufficient description about my thesis topic. Definition of various terms, mathematical background, technology that we have applied for implementing our work.
- **Chapter 3 Proposed System.** It representing the whole process of our system, each part of our system that we have applied in our system.
- **Chapter 4 Implementation.** The chapter named implementation process encloses the procedure of the method implementation.
- **Chapter 5 Experimental Results and Discussion.** This chapter contains the experimental set up and some experimental results that have been done during implementation of the method. We try to represent the performance analysis here.
- **Chapter 6 Conclusion and Future Work.** This chapter concludes our thesis work mentioning the goal that we are achieved in by completing our work. We also mention the correctness and accuracy of our methodology by which we have done the work.

Chapter 2

Literature Review

Prior to many works has been done by many researchers on English for ranking document. They used many methods or technique for ranking document. There are some overview of previous works that is given below–

2.1 Donna Harman,1986

The use of term weighting in a document produces significant gains in perfonnance, up to a 44% improvement in average precision over simple matching. Additionally the following conclusions can be drawn from the experiments.

1. Here three types of measures tested
    - the importance of a term within a document collection
    - the importance of a term within a given document, and
    - the length of a document, are all important in term weighting of documents
2. The two types of term-importance factors, impatience within a collection, and importance within a given document, measure term usage in two complementary places-within a given document and within an entire document collection– and combining them produces a cumulative effect
3. The normalized noise measure of term importance within a document collection is a viable alternative measure to the inverted document frequency

measure. Using the log2, of the frequency of a term within a document instead of its raw frequency produces a superior measure of the importance of a term within a given document.

4. Adding the number of matches between a document and a query to a term weight produced by any combination involving a factor that measures term importance within a collection does not produce significant improvement in performance, at least for this test collection and for full word indexing.

5. It is important to consider the length of a document in ranking. Dividing the total term weight by the log of the document length produces significant performance improvement for this test collection.

2.1.1 Limitation

The future research is needed to resolve some of the issues. The log2 of the frequency and of the document length is only one possible function of these measures. there is no removal stop words and stemming.

2.2 Kent E. Seamons and Dik Lun Lee(1997)

In this paper they used Vector Space Model for ranking document. They showed five methods for ranking method. They are…

2.2.1 Method 1

The full vector is rarely stored internally as is because it is long and sparse. Instead, document vectors are stored in an inverted file that can return the list of documents containing a given keyword and the accompanying frequency information. Besides, direct comparison between the vectors is slow because it would incur N vector comparisons. Vector comparison can be facilitated with an inverted file as follows.

Algorithm 1: Vector Comparison

```
while every query term q in Q do
    retrieve the postings list for q from the inverted file;
    while each document d indexed in the postings list do
        score(d)=score(d)+tf(d,q)*idf(q);
    end
end
Normalize scores;
Sort documents according to normalized scores;
```

With an inverted file, the number of postings lists accessed equals the number of query terms. The computational cost is acceptable for queries of reasonable size. Unfortunately, the computation of the normalization factor is extremely expensive because the term $\sum_{j=1}^{V} w^2(i,j)$ in the normalization factor requires access to every document term, not just the terms specified in the query. Nor can the normalization factor be precomputed under the $tf * idf$ method, because every insertion and deletion on the document collection would change idf and thus the precomputed normalization factors.

2.2.2 Method 2

For this second method, to approximate the effect of normalization we use instead the square root of the number of terms in a document as the normalization factor. While this still favors long documents, the effect of document size is not as significant as it is without any normalization. This normalization factor is much easier to compute than the original one; also, pre computation is possible. With the approximation, the formula becomes

$$sim(Q, D_i) = \frac{\sum_{j=1}^{V} w_{Q,j} * w_{i,j}}{\sqrt{number of terms in D_i}}$$

2.2.3 Method 3

This method lets us further simplify the computation by simply dropping the normalization factor:

$$sim(Q, D_i) = \sum_{j=1}^{V} w_{Q,j} * w_{i,j}$$

That is, the document score equals the inner product of the document and query vectors. Instead of computing the angle between the document and query vectors, this formula computes the length of the projection of the document. vector onto the query vector. It is quite clear that the document score is directly proportional to the length of the document vector.

2.2.4 Method 4

This method only makes use of term frequencies in the calculation and ignores idf. It simplifies the computation as well as saving the file structure needed for storing the df values.

$$sim(Q, D_i) = \sum_{j=1}^{V} w_{Q,j} * tf_{i,j}$$

2.2.5 Method 5

This method ignores the term frequency (tf) information but retains the idf values in determining term weights:

$$sim(Q, D_i) = \sum_{j=1}^{V} w_{Q,j} * w_{i,j} \text{ , where}$$

$$w_{Q,j} = \begin{cases} 1 & j \in Q \\ 0 & \text{Otherwise} \end{cases}$$

and

$$w_{i,j} = \begin{cases} 1 & j \in D_i \\ 0 & \text{Otherwise} \end{cases}$$

The *idf* values have the same effect as before. That is, they diminish the significance of words that appear in a large number of documents.

2.2.6 Method 6

This method is the simplest in the family. It ignores both tf and *idf* values and therefore measures the number of common terms in the document and query vectors.

$sim(Q, D_i) = \sum_{j=1}^{V} w_{Q,j} * w_{i,j}$, where

$$w_{Q,j} = \begin{cases} 1 & j \in Q \\ 0 & \text{Otherwise} \end{cases}$$

and

$$w_{i,j} = \begin{cases} idf_i & j \in D_i \\ 0 & \text{Otherwise} \end{cases}$$

The performance gain is as much as 25 percent over baseline. However, there is no overwhelming evidence to indicate which combination works best. For concept queries, we find it best to use only inverse document frequencies while ignoring term frequencies, but for natural-language queries both frequencies are needed—ignoring either one will produce poor results. Furthermore, the approximate normalization factor we suggest here gives the best performance and is computationally

efficient. They have found that high-frequency terms or terms in the neighborhood of hits selected from context units containing at least one hit give the best feedback performance. In some cases, the gain in average precision is as much as 20–25 percent. On the other hand, although the data shows slight improvement in using terms around hits overhigh-frequency terms, the evidence is not overwhelming. To develop an efficient and effective retrieval system, many implementation issues must be considered.

2.3 Xuehua Shen & ChengXiang Zhai (2003)

In interactive information retrieval [4], at any moment, they may assume that a sequence of "past queries", or "query history", q1……qk-1, are available for a topic, and the current user query is qk. Normally, only qk is used to rank the documents in the collection. They proposed to combine q1, ..., qk-1 together with qk to have a richer model of the user's information need. Our basic retrieval model is the KL-divergence retrieval model. They explore two different strategies –combining query results and combining query models. Combining query results means that we merge the results from q1, ..., qk by taking an average of the rank of each document. Such a method would reward a document that has been ranked high by all the queries. Combining query models means that we estimate a query language model for each query qi using the maximum likelihood estimator, and then take an average of these query models to obtain a contextbased query model, which is then used to rank documents with the KL-divergence ranking formula. The probability of a word w according to the context-based query model is given by

$$p(w|q1, \ldots\ldots q^k) = \frac{1}{k} \sum_{i=1}^{k} \frac{c(w,q^i)}{|q^i|}$$

where, $c(w; qi)$ is the counts of word w in query qi and $|qi|$ is the length of query qi. Note that this is different from concatenating the query text of $q1, ..., qk$ to obtain a long query, since they normalize the counts of a query word by the length of

each query, which can avoid dominance of one single query. For each document d, they also estimate a smoothed document language model using the Dirichlet prior smoothing method. We then rank documents by the KL divergence value of the query model and the document model. That is, document d has a score of

$$\sum p(\frac{w}{q1}, ...q^k) log(1 + \frac{c(w,d)}{\mu p(W|C)}) + log \frac{\mu}{\mu + d}$$

where, $p(w|C)$ is the collection language model and is the Dirichlet prior smoothing parameter . In interactive information retrieval, especially for hard topics, the user would generally need to submit a sequence of queries to the search system. They demonstrate that using query history to expand the current query can consistently improve the retrieval performance.

We have only explored the most basic methods for exploiting the query history; more sophisticated analysis is certainly possible and interesting to explore (e.g., term sequence analysis and unequal weighting of different queries).

2.4 Our Proposed System

We use term frequency and cosine similarity for ranking bangla multiple document. For finding cosine similarity we use the equation

$$\cos(q, d) = \frac{\overrightarrow{q} . \overrightarrow{d}}{|\overrightarrow{q}||\overrightarrow{d}|}$$

We take query as a vector and document as a vector then find similarity between them and calculating scores we rank document.

Chapter 3

Proposed System

We have used two methods for ranking document. One is term frequency and another is cosine similarity.

3.1 Creating Document Vector

The process of creating document vector is given below.

3.1.1 Term Frequency

A term that appears many times within a document is likely to be more important than a term that appears only once.

D1: আমি বাংলাদেশকে ভালবাসি । বাংলাদেশ নদীমাতৃক দেশ।

D2: আমি বাংলাদেশের নাগরিক । কিন্তু বাংলাদেশের সকল নাগরিক তাদের মোলিক অধিকার পায় না।

D3: বাংলাদেশ উন্নয়নশীল দেশ। এখনো উন্নয়নের দিক থেকে পিছিয়ে আছে বাংলাদেশ ।

After performing the term frequency calculation we get the table 3.1. Which is showing the value each word each document.

3.1.2 Stop Word Removing

Documents contain words that do not add information but are necessary for syntactical formation,such as words like এবং , অথবা , কিন্তু etc. Since these words are less useful and less informative, they introduce noise into the document representation. In order to get rid of these kind of words, a stop word removal step is used. Stop

Table 3.1: Parameters of datasets

Term	Doc1	Doc2	Doc3
আমি	1	1	
বাংলাদেশ	2	2	2
ভালবাসি	1		
উন্নয়নশীল			
দেশ			1
নদীমাতৃক	1		
কিন্তু		2	
সকল		1	
মোলিক		1	
অধিকার		1	
পায়		1	
এখনো			1
দিক			1
থেকে			1
পিছিয়ে			1
আছে			1
নাগরিক		2	
না		1	
তাদের		1	

word removal is done using predefined, human-made list of words. Since a predefined list is used, this approach is language dependent. Instead of using these kinds of lists, a frequency threshold can be used. If a word is seen moreless frequently than predefined threshold, that word can be considered as stop word. But decision of threshold is another issue to be considered.

3.1.3 Stemming

In a document a word can be seen in different formats, such as plural vs. singular, present vs. past tense, etc. Most of the time these words have the same meaning and treating them differently is unnecessary. In order to use these words as the same token(concept), stemmers are used. Stremmers are tools that reduce the orginal word forms into roots(stems) of these words. Stemmers are necessary to represent different forms in a single format and to reduce memory usage for storing the words. Also, smaller list of words make it easir to perform calculations. As a result of performing stemming,document representation is less noisy and more dense. The efficiency of a stemmer is important while performing further calculations. In Figure 3.3 ,we can see the full process of ranking.

Sometimes stemmers can do over-stemming such that two words are given the same stem,while it should not be. For example, the words "বাংলাদেশের" and "বাংলাদেশকে" are two different words, which should not be stemmed into the same root. But stemmers can find out their root as "বাংলাদেশ". Another steamming problem is related to under stemming such that two words should have been stemmed into the same word, but have not been. for example, "হাসি" and 'হাসানো' can be found as two different stems,instead of one.

3.1.4 Weighting Term

Then find weight in a document for each term using the equation 3.1

$$Weight += postings[term][doc] * IDF \tag{3.1}$$

Then assigning document IDS we keep them into database as vector after doing following steps which is shown in Figure 3.1.

3.1.4.1 Inverse Document Frequency(IDF)

A term that occurs in a few documents is likely to be a better discriminator than a term that appears in most or all documents.

Then assigning document IDS we keep them into database as vector after doing these steps.

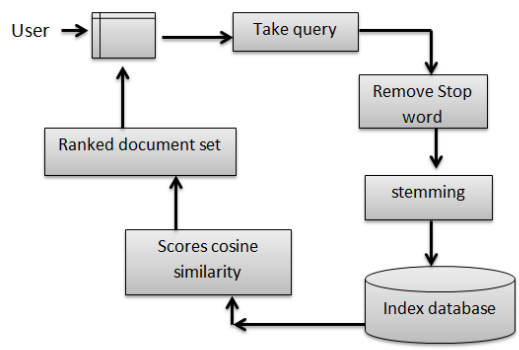

Figure 3.1: **Indexing Document Ids & keep it in a vector.**

3.2 Preprocessing of Taking Query

For taking search query the preprocessing steps have to be done. After taking query step by step process has been done . And then match query with relevent documents. The preprocessing steps are showing in the figure 3.2 below.

3.2.1 Removal of Stop Word

The less importance word should be removed from query. Sometimes input query contain words that do not add information, such type words ‘এবং’ , ‘অথবা’ , ‘কিন্তু ’ etc should remove from document.

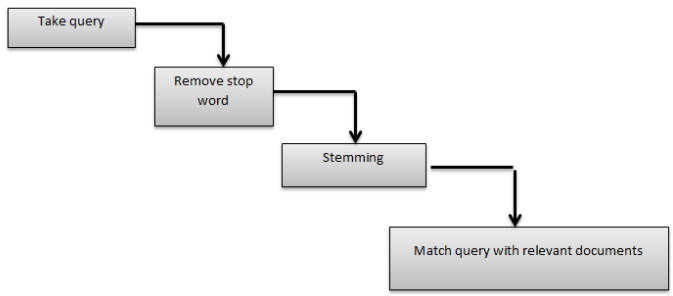

Figure 3.2: **Pre processing of Input query**

3.2.2 Stemming

Finding root words from other similar words which have the same meaning and treating them differently is unnecessary. Figure 3.3 showing the process.

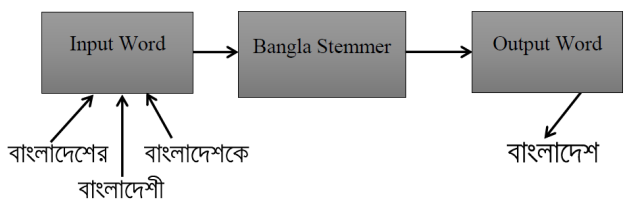

Figure 3.3: **Stemming processing**

3.3 Cosine Similarity

The query matches with relevent documents. If exists, find out the relevent documents and give them a scores using query term's IDF multiplying with weight. Then sort the values and rank the documents.

Example: There are three documents.

D1: আমি বাংলাদেশকে ভালবাসি । বাংলাদেশ নদীমাতৃক দেশ।

D2: আমি বাংলাদেশের নাগরিক । কিন্তু বাংলাদেশের সকল নাগরিক তাদের মৌলিক অধিকার পায় না।

D3: বাংলাদেশ উন্নয়নশীল দেশ। এখনো উন্নয়নের দিক থেকে পিছিয়ে আছে বাংলাদেশ ।

Input Query: বাংলাদেশের নাগরিক হিসেবে বাংলাদেশের উন্নতি চাই।

$$Cosine(document, query) = (document.query)/|documentvt||queryvt|$$

Now in table 3.2 we can see the scores after doing the all process.

Table 3.2: Scoring Each Document

Document Id	Documents	Scores
1	D1	0.145
2	D2	0.543
3	D3	0.345

After sorting scores, we get the ranked documents and we can see that in table 3.3.

Table 3.3: Rank Documents

Document Id	Documents	Scores
2	D2	0.543
3	D3	0.345
1	D1	0.145

3.4 Main Approaches

When the preprocessing is completed, then our main process focuses on the analysis of the processed output by our own developed tools for Bangla document ranking. Our main approach are includes to generating term frequency vector, assigning cosine similarity to the corresponding document are known as document scoring. Then sorting the scores, we rank the documents. In figure 3.4 shows the overall graphical view of our proposed system.

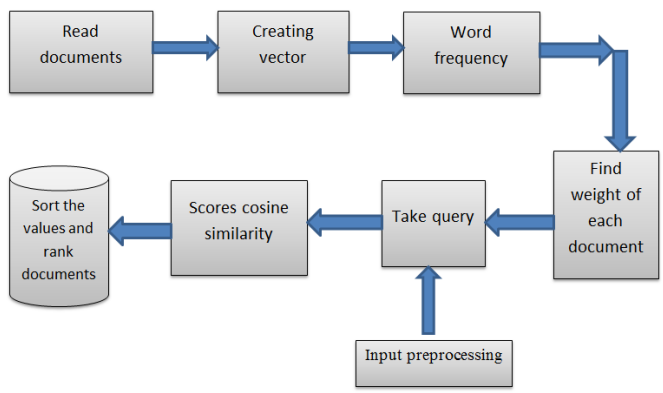

Figure 3.4: **Overall graphical view of our proposed system**

Chapter 4

Implementation

Implementation is the process where we can see how our system works steps by steps.

4.1 Bangla Corpus

Bangla language contains huge range of vocabulary which makes the language variegated in the world. Here, we develop a dictionary data set where 30 documents are used. In this system, we use the dictionary several times. It has two main reasons to access this dictionary. First one is to check a word which is rooted or not and second one is to get the associated POS tag for a word.

4.2 Bangla Stemming

Stemming is an operation that splits a word into its constituent root part and affix without doing complete morphological analysis. Terms with common stems tend to have similar meaning, which makes stemming an attractive option to increase the performance of news categorization task, where morphological analysis would be too computationally expensive. Another advantage of stemming is that it drastically reduce the vocabulary size of highly inflected languages corpus like Bangla. The algorithm of using bangla stemmer

Algorithm 2: bangla-stemmer (word)

```
while each word 2 document do
    dictionary-checkers(word);
    if word 2 dictionary then
        stem=word;
    else
        stem1=stemming(word);
        if stem1 2 dictionary then
            stem=stem1
        else
            stem=stemming(stem1)
        end
    end
end
```

4.3 Bangla Stop Words

Statistical analysis through the documents showed that some words have quite low frequency, while some others act just the opposite. The common characteristic of these words is that they carry no significant information and used just because of grammar. This set of words are usually known as stop words. In the resulting stop word list, there were thus a large number of pronouns, articles, prepositions, and conjunctions. As in various English stop-word lists, there were also some verbal forms. When using, this stop word list, the vocabulary size reduced significantly.

4.4 Term Weighting

Term Weighting of documents can be evaluated by measuring the Term Frequency (TF) and Inverse Document Frequency (IDF). These are the statistical measurement of weight that is intended to determine the importance of a word for a document in a corpus. It can be used for stop-word filtering. This is the combine definition of

Normalized Term Frequency and Inverse Document Frequency.

4.4.1 TF

Term Frequency measures how frequently a word occurs in a document. Different document varies in length. Therefore, a word can be occurring more times in a larger document than shorter. Thus, the raw frequency of a word is divided by the length of the document.

$$tf(t,d) = \frac{f(t,d)}{lengthofd} \tag{4.1}$$

here f(t,d) is the raw frequency of word t in document d. Using equation 4.1 ,term frequency can be defined.

4.4.2 IDF

Inverse Document Frequency measures the importance of a word within a document. TF provides same importance for every word, where IDF provides less weight to the frequent word and high weight to the rare word.

$$idf(t) = log(N/DF) \tag{4.2}$$

Using equation 4.2, the idf can be measured.

The TF-IDF weighting scheme sets a weight to a word t in document d,which is shown in equation 4.3

$$tf - idf(t,d) = tf(t,d) * idf(t) \tag{4.3}$$

The weight of a word t in document d is highest when t occurs many times in a small number of document, and lower when t occurs a very few times in a document or occurs in many documents of the corpus.

4.5 Cosine Similarity between query and document

The similarity measures comparing between document vector and query vector. Though the angle between two vectors considered (0 to 90), than the similarity lies between 1 to 0.

$$Similarity = \cos\theta = \frac{\overrightarrow{A}.\overrightarrow{B}}{|\overrightarrow{A}||\overrightarrow{B}|} \tag{4.4}$$

The equation 4.4 is used for defining similarity.

Chapter 5

Experimental Result & Discussion

In order to execute our system, we have done experiments and discussed of the experiment result.

5.1 Experimental Result

To examine our ranking, we collect 30 bangla documents from the bangla daily newspaper eg Prothom alo, kaler kontho etc. The documents are written and saved in the text files using UTF-8 format. For each document in our corpus, we consider only one human ranking for evaluation. Evaluation of a system produced ranking is done by comparing it to the human ranking. There are some documents here on boimela.

Doc 1 : boimela txt.

কলকাতায় বাংলাদেশ বইমেলা থাকছে ৫০ প্রকাশনী । বইয়ের বন্ধুত্ব সীমানা ছাড়িয়ে- স্লোগানে ১ সেপ্টেম্বর থেকে কলকাতায় শুরু হচ্ছে ১০ দিনব্যাপী 'বাংলাদেশ বইমেলা'। সচিবালয়ে সোমবার সংস্কৃতি সচিব আকতারী মমতাজ এক সংবাদ সম্মেলনে জানান, ষষ্ঠবারের মতো আয়োজিত মেলায় বাংলাদেশের ৫০টি প্রকাশনা প্রতিষ্ঠান অংশ নেবে। ১ সেপ্টেম্বর বিকেল ৫টায় মেলার উদ্বোধন করবেন ইমেরিটাস অধ্যাপক আনিসুজ্জামান। উদ্বোধনী অনুষ্ঠানে থাকবেন পশ্চিমবঙ্গ সরকারের শিক্ষামন্ত্রী পার্থ চট্টোপাধ্যায় ও কবি-প্রাবন্ধিক শঙ্খ ঘোষ। প্রতিদিন দুপুর ২টা থেকে রাত ৮টা পর্যন্ত মেলা উন্মুক্ত থাকবে। শনি ও রোববার বিকাল ৩টা থেকে রাত ৮টা পর্যন্ত মেলা চলবে। জাতীয় গ্রন্থকেন্দ্র ও রপ্তানী উন্নয়ন ব্যুরোর সহযোগিতায় এবং কলকাতায় বাংলাদেশ উপ-দূতাবাসের ব্যবস্থাপনায় বাংলাদেশ জ্ঞান ও সৃজনশীল প্রকাশক সমিতি গত পাঁচ বছর ধরে কলকাতায় বাংলাদেশ বইমেলার আয়োজন করছে।প্রথম তিন বছর মেলাটি গণকেন্দ্র শিল্প সংগ্রহশালায় হলেও গত দুই বছর ধরে রবীন্দ্র সদনের উন্মুক্ত প্রাঙ্গণে হয়। এবারও এই উন্মুক্ত প্রাঙ্গণে বাংলাদেশ বইমেলা বলে জানান

আকতারী মমতাজ।

Doc 2: accident.txt

ট্রেনের ধাক্কায় রাবি শিক্ষার্থীর মৃত্যু। ফোনে কথা বলার সময় পেছন থেকে ট্রেনের ধাক্কায় রাজশাহী বিশ্ববিদ্যালয়ের এক ছাত্রীর মৃত্যু হয়েছে। রোববার বিকাল সোয়া ৪টার দিকে বিশ্ববিদ্যালয়ের চারুকলা গেটের কাছে পদ্মা এক্সপ্রেস ট্রেনের ধাক্কায় শান্তনা বসাক মারা যান। শান্তনা সমাজকর্ম বিভাগের তৃতীয় বর্ষের শিক্ষার্থী। তিনি সিরাজগঞ্জের তাড়াশ উপজেলার মাধইনগর গ্রামের নরেন্দ্রনাথ বসাকের মেয়ে। বেগম রোকেয় হলের আবাসিক শিক্ষার্থী ছিলেন তিনি। প্রত্যক্ষদর্শীরা জানান, শান্তনা বসাক বিশ্ববিদ্যালয়ের চারুকলা অনুষদের পাশের রেল লাইনে হেঁটে হেঁটে মোবাইল ফোনে কথা বলছিলেন। কানে হেডফোন লাগানো ছিল। চারুকলা গেটে দায়িত্বরত পুলিশ সদস্যরা বেশ কয়েকবার তাকে রেল লাইন থেকে সরে যেতে বললেও তিনি খেয়াল করেননি। ওই সময় রাজশাহী থেকে ঢাকাগামী পদ্মা এক্সপ্রেস ট্রেনটি চারুকলা গেট অতিক্রম করছিল। পেছন থেকে আসা ট্রেনটি থেকে বারবার হর্ন বাজালেও ফোনালাপে মগ্ন থাকায় শান্তনা সরেননি। এ সময় পেছন থেকে ট্রেনটি ধাক্কা মারলে শান্তনা গুরুতর আহত হন। পরে শিক্ষার্থীরা তাকে উদ্ধার করে রাজশাহী মেডিকেল কলেজ হাসপাতালে নিয়ে যান।

Doc 3: rajshahi.txt

রাজশাহী মেডিকেল কলেজ হাসপাতালে কর্তব্যরত সহকারী উপপরিদর্শক মো. শফিক বলেন, অতিরিক্ত রক্তক্ষরণের কারণে চিকিৎসা শুরুর আগেই তার মৃত্যু হয়েছে। বিশ্ববিদ্যালয়ের সমাজকর্ম বিভাগের সভাপতি অধ্যাপক ছাদিকুল আরেফিন বলেন, পরিবারের সদস্যদের খবর দেওয়া হয়েছে। তার ভাই মরদেহ নিতে আসছেন বলে জানিয়েছেন। ময়না তদন্তের পর পরিবারের কাছে মরদেহ হস্তান্তর করা হবে। “বিভাগের তৃতীয় বর্ষের ওই শিক্ষার্থীর অকাল মৃত্যুতে আমরা গভীরভাবে শোকাহত।” নগরীর মতিহার থানার ওসি হুমায়ুন কবির বলেন, “বিশ্ববিদ্যালয়ের এক ছাত্রী ট্রেনে কাটা পড়ে মারা গেছে বলে শুনেছি। তবে এটা আত্মহত্যা নাকি দুর্ঘটনা এ ব্যাপারে নিশ্চিত হওয়া যায়নি। বিষয়টি খোঁজ নেওয়া হচ্ছে।”

Input Query : রাজশাহী বিশ্ববিদ্যালয়ের এক ছাত্রীর মৃত্যু হয়েছে

When we apply the proposed system to our Bangla document,this machine generate cosine similarity to corresponding query and documents. Finally, the cosine values of corresponding documents are given in table 5.1. The goal 0f document ranking is to present the relevant documents corresponding users query. This process is slightly similar to text summarization. It helps us to find out the documents

easily that users want.

Table 5.1: Documents with corresponding cosine values

Document Id	Document text file	Cosine similarity
Doc1	Boimela.text	0.14
Doc2	Accident.txt	0.36
Doc3	Rajshahi.txt	0.35

When the cosine similarity is enlisted for the corresponding documents then the document is rearranging according to their root document. Finally taking the top rated document is taken for ranking. The ranked document is:
$Accident.txt > Rajshahi.txt > Boimela.text$

We balanced Our system to rank with the human ranking, calculating the following scores. For document ranking we let "Kh" be the number of ranked document generated by human which is corresponded to the query sentence. "Km" be the number of ranked document generated by system which is corresponded to the query sentence and "r" the number of ranking they share. We defined precision (P), recall (R) and accuracy to compare the two summaries by:

$$P = r/kh\% \tag{5.1}$$

$$R = r/km\% \tag{5.2}$$

$$Accuracy = 2 * r/kh + km\% \tag{5.3}$$

To evaluate the accuracy of our proposed system we examined the 30 documents with the above accuracy equation 5.3. For this we calculate the precision and recall

value using the above equation 5.1 and 5.2 sequentially. The output of the given equation are the input of the proposed system to find the level of accuracy using equation 5.3 . Finally the average accuracy of bangla document ranking is 77.07848 % corresponding with human generated ranking . The accuracy on the basis of equation 4.1 is given Table [5.2] . In Table[5.2],we can see our whole experiment.

5.1.1 Experiment 1

We take 10 documents and according to query human and system give the ranking documents number respectively 7 and 6. The common ranking number between them is 5. So, the accuracy using equation 5.3, we get 76.92308 percent.

5.1.2 Experiment 2

Again, we take different 10 documents and change query, according to query human and system give the ranking documents number respectively 6 and 5. The common ranking number between them is 4. So, the accuracy using equation 5.3, we get 72.72727 percent.

5.1.3 Experiment 3

We take 10 documents which is not matched prior to documents and according to query human and system give the ranking documents number respectively 5 and 6. The common ranking number between them is 3. So, the accuracy using equation 5.3, we get 54.54545 percent.

5.1.4 Experiment 4

We take 10 documents which is not matched with experiment 1,2 and 3 documents. Then according to query which is not matched prior query human and system give the ranking documents number respectively 6 and 7. The common ranking number between them is 6. So, the accuracy using equation 5.3, we get 92.30769 percent.

5.1.5 Experiment 5

We take 10 documents which is not matched with experiment 1,2,3 and 4 documents. Then according to query which is not matched prior, query human and system give the ranking documents number respectively 5 and 4. The common ranking number between them is 4. So, the accuracy using equation 5.3, we get 88.88889 percent.

In Figure 5.1, we can see the accuracy in each experiment and can see the average of total experiments.

Table 5.2: Accuracy Measurement

Experiment number	Document number	Kh	km	r	Accuracy	Average
1	10	7	6	5	76.92308	
2	10	6	6	4	72.72727	
3	10	5	6	3	54.54545	77.07848
4	10	6	6	6	92.30769	
5	10	5	6	4	88.88889	

Finally the average result is around 77.07848%.. The graph of the document ranking is shown in Figure 5.1.

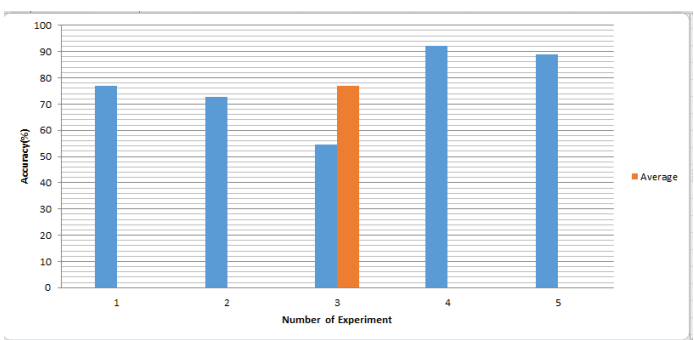

Figure 5.1: **Graphical view of document ranking**

It is very difficult to determine if a ranking is good or bad. The ranking examined methods can be broadly categorized as human evaluation methods and

automatic(machine-based) evaluation methods. A human evaluation is done by comparing system-generated ranking with reference model ranking by human judges.

The main problems with human evaluation are:

- The evaluation process is exhausting
- It sustains from the lack of unity

Two human judges may not agree on each other's judgments. Since automatic evaluation is performed by a machine, it follows a fixed logic and always produces the same result on a given ranking. Since automatic evaluation process are free from human bias, it provides a consistent way of comparing the various ranking systems.

5.2 Discussion

In this thesis, we explain the basic ideas of hoe to rank different documents according to their relevance. The ideas used are very beautiful. They are some fearsome-sounding vector space model for documents. Instead of thinking of documents and queries as strings or letters, we adopt a point of view in which both documents and queries are represented as vectors in a vector space. In this point of view, the problem of determining how relevant a document is to a query is just a question of determining how parallel the query vector and the document vector are. The more parallel the vectors, the more relevant the document is.

This geometric way of treating documents turns out to be very powerful. It's used by most modern web search engines, including (most likely) web search engines such as Google as well as search libraries. The ideas can also be used well beyond search, for problems such as document classification, and for finding clusters of related documents.

Chapter 6

Conclusions and Future Scopes

6.1 Conclusion

In Bangla document ranking, the term frequency and cosine similarity perform the utmost level of accuracy (88.10%) than any other existing methods. The model we've described treats all documents on an equal footing. In addition to, this is the extraction based ranking system, is not real time ranking formation. In the typical web search setting it is important to users that the most relevant documents are top-ranked given the large number of potentially relevant documents. For each term in the dictionary, it's straightforward to pre compute the top 1,000(say) documents for that term. Then for a given multi-term query it's pretty likely that the top search results will come from one of the pre-computed lists of top documents for the terms in the query.

6.2 Future Scopes

The problem is that it computes the cosine similarity for every single document in the corpus. In future, ideas that are used in other methods, will be used together with the proposed approaches to improve the performance of the ranking system. We will compare our method with other methods.

Bibliography

[1] D. W. Harman, "An experimental study of factors important in document ranking," in *Proceedings of the 9th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1986, pp. 186–193.

[2] D. L. Lee, H. Chuang, and K. Seamons, "Document ranking and the vector-space model," *IEEE software*, vol. 14, no. 2, pp. 67–75, 1997.

[3] J. Carbonell and J. Goldstein, "The use of mmr, diversity-based reranking for reordering documents and producing summaries," in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1998, pp. 335–336.

[4] X. Shen and C. X. Zhai, "Exploiting query history for document ranking in interactive information retrieval," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. ACM, 2003, pp. 377–378.