English Bengali Ad-hoc Monolingual Information Retrieval Task Result at FIRE 2008

Sivaji Bandhyopadhyay
Department of Computer Science and Engineering
Jadavpur University, Kolkata-700032, India
sivaji_cse_ju@yahoo.com

Amitava Das
Department of Computer Science and Engineering
Jadavpur University, Kolkata-700032, India
amitava.santu@gmail.com

Pinaki Bhaskar
Department of Computer Science and Engineering
Jadavpur University, Kolkata-700032, India
pinaki.bhaskar@gmail.com

ABSTRACT
This paper presents the experiments carried out at Jadavpur University as part of the participation in the Forum for Information Retrieval Evaluation (FIRE) 2008 in ad-hoc mono-lingual information retrieval task for English and Bengali languages. The experiments carried out by us for FIRE 2008 are based on stemming, zonal indexing; TFIDF based ranking model and positional information. The document collection for English and Bengali contained 125,638 and 123,047 documents respectively. Each query was specified using *title, narration and description* format. 25 queries were used for training the system while the system was tested with 50 queries in each of English and Bengali.

1. Introduction

Ad-hoc monolingual Information retrieval research involves the study of systems that accept queries (or information needs) in natural language and return objects related to that query. The objects could be text documents, passages, and images, audio or video documents. For the present task concentration is only on news text documents. Corpus and the query set were provided for the experiments as part of the Forum for Information Retrieval Evaluation (FIRE) 2008 Ad-hoc Monolingual Information Retrieval task for English and Bengali languages.

Various techniques have been used so far in the area of Mono-Lingual Information Retrieval. These techniques can be broadly classified [1] as controlled vocabulary based and free text based systems at very high level. However, it is very difficult to create, maintain and scale a controlled vocabulary for general purpose IR systems in a general domain for a large corpus. Very little work has been done in the past in the areas of IR and CLIR involving Indian languages. A few other information access systems were built apart from this task such as cross language Hindi headline generation [2], English to Hindi question answering system [3] etc. Our aim was to find out the most optimum algorithm for the ad-hoc mono-lingual information retrieval. Conceptually a general pre-processing technique like stemming, stop word removal, phrase detection, paragraph detection etc. was done for the query set and document set both. In our previous participation in Cross Language Evaluation Forum (CLEF 2007) [4] we proposed a semi-automatic query term list preparation but for the present task an automatic n-gram phrase detection technique has been developed for both the query processing and the document processing. Very little work has been done in the past in the areas of IR and CLIR involving Indian languages. The International Institute of Information Technology (IIIT) in Hyderabad, India built a monolingual web search engine for various Indian languages, which is capable of retrieving information from multiple character encodings [5]. The Government of India has initiated a consortia project titled "Development of Cross–Lingual Information Access System" [6], where the query would be in any of the six different Indian languages (Bengali, Hindi, Marathi, Telugu, Tamil, Punjabi) and the

output would be also in the language desired by the user.

2. Corpus Statistics

The corpus for ad-hoc mono-lingual retrieval was made available as par of FIRE 2008. Each language data consists of four consecutive years of news from the archives of two reputed newspapers published from Kolkata: The Telegraph (English) and Anandabazar Patrika (Bengali). The copyright issues for the newspaper corpus were taken up with the publishers by the FIRE 2008 organizers. Corpus was sub divided into many other sub-divisions like District, State, Sports, Editorial, and Government etc. Corpus statistics for both the languages are in tabular format below.

Bengali Corpus Statistics

Total number of documents in the corpus	123,047
Total number of wordforms in the corpus	1245019
Avgerage number of wordforms in a document	317

Table 1 Bengali Corpus Statistics

English Corpus Statistics

Total number of documents in the corpus	125,638
Total number of wordforms in the corpus	1032019
Avgerage number of wordforms in a document	274

Table 2 English Corpus Statistics

3. Our Approach

The experiments carried out by us for FIRE 2008 are based on dynamic zonal indexing, distribution and positional factor of a particular query term and TF-IDF based ranking model. The method of a dynamic zonal indexing based on the number of paragraphs in the document was applied on the document collection. Sentences within a paragraph were treated as belonging to the same zone. Hence the number of zones in a document is dynamic in nature, it depends on the number of paragraphs in the document. Hence documents are divided into dynamic number of zones. Two different pre-processing techniques for query and corpus were defined. Firstly from the query set all stemmed words are collected and same stemming module is used for corpus also. For the query search technique a standard Index file has been created. The Index file consists of query terms (identified from the training set of 25 queries) and a relevant weight for each document. During retrieval, the weight of a particular document with respect of a particular query is the summation of weights calculated for the document against the total number of query terms. The relevance weight of a document for a particular query term is calculated using the following equation:

$$W_d = (\sum F_p / W_t) * (1 / D_f) * (\text{TF-IDF})$$

Where W_d = Calculated weight for a document

F_p = Positional frequency of a query term in a particular sentence contains the query term

W_t = Total number of words in the sentence contains query term

D_f = Distribution Function (Detail in section 3.2)

3.1 Term Distribution Model

An alternative to TF-IDF weighting is to develop a model for the distribution of a word and to use this model to characterize its importance for retrieval. That is, we wish to estimate P_i (k) that measures the distribution pattern of the *k* occurrences of the word w_i in a document. In the

simplest case, the distribution model is used for deriving a probabilistically motivated term weighting scheme for the vector space model. But models of term distribution can also be embedded in other information retrieval frameworks. Apart from its importance for term weighting, a precise characterization of the occurrence patterns of words in text is arguably at least as important a topic in Statistical NLP as Zipf's law. Zipf's law describes word behaviour in an entire corpus. In contrast, term distribution models capture regularities of word occurrence in subunits of a corpus (e.g., documents, paragraphs or chapters of a book). In addition to information retrieval, a good understanding of distribution patterns is useful wherever we want to assess the likelihood of a certain number of occurrences of a specific word in a unit of text. Most term distribution models try to characterize how informative a word is, which is also the information that is identified by inverse document frequency. One could cast the problem as one of distinguishing content words from non-content (or function) words, but most models have a graded notion of how informative a word is. In the present work, the distribution pattern of a word within a document formalizes the notion of informativeness. This is based on the Poisson distribution; one motivates inverse document frequency as a weight optimal for Bayesian classification and the final one, residual inverse document frequency, can be interpreted as a combination of *idf* and the Poisson distribution.

3.2 Distribution Function

The distribution function for each query term in a document is evaluated as follows:

$$D_f = \sum_{i=1}^{n} (S_i - S_{i-1})/n$$

Where n=number of sentences in a document with a particular query term

S_i=sentence id of the current sentence containining the query term

and S_{i-1}=sentence id of the previous sentence containing the query term

4. Pre-Processing

4.1 Query Pre-Processing

Corpus pre processing and query pre processing are done separately. In Query pre processing the three components of the query i.e. title, description and narration are considered. These three components were separated in the XML query file with appropriate tag set. A list of 440 domain independent Bengali stop word was created manually. A similar list of … English stop words is also used. The stop word removal and suffix stripping algorithm works simultaneously. The details of working principal of the suffix stripping algorithm are in the section 5. Before the stop word removal a rule based n-gram term detection algorithm is applied. It searches for common terms in the title, description and narration section and give a higher score for the common terms. The query term list is ordered based on the score associated with each query term. An example of an English query is in Figure 1

<title>FIFA World Cup 2006***</title>***
<desc>Find documents reporting on the 2006 FIFA World Cup which was held in Germany in which Italy won the title.**</desc>**
<narr>Information regarding the FIFA World Cup 2006 which was held in Germany and the teams which qualified for the quarter-finals, semi-finals, final and Italy's win over France in the final is relevant.***</narr>***

Figure 1. Sample English Query

The suffix stripper module strips the suffixes from the surface form of the word and gives the stemmed output.

4.2 Corpus Pre-Processing

Both the English and the Bengali news corpus made available as part of FIRE 2008 was in XML format. A cleaning process was applied on the news corpus to extract the *title* and the *news body* from every document. The process of stop word removal first removes the stop words from the document and the suffix stripping module then strips the suffixes from every word and keeps them in the original order of their occurrence in the document.

5. Stemming (English) and Suffix Stripping (Bengali)

For the English corpus pre-processing we use standard Porter stemmer [7] module. A suffix stripper algorithm was used for identifying the stems of Bengali surface words occurring in the documents. The algorithm clusters same type of words (i.e., words that probably share the same stem) together. A minimal string matching algorithm compares two strings and gives a score to each on the basis of smallest string among the two; how many characters have to be deleted and/or added to reach the longest string. A suffix list is used during the string matching operation. Hypothetically we assume the smallest string among the clusters, i.e., the cluster centre, to be the stem of the words present in the cluster.

6. Dynamic Zonal Indexing

In zonal indexing [8], a particular document is divided into n number of zones/regions, say, w1, w2 , …… ,wn. In the present work, the document is initially divided into the paragraphs and each paragraph is given a weight *1/p*, where the *p* is the paragraph number. Each sentence in the paragraph and each word position within a sentence are given similar weights based on their position of occurrence. The weight of a query word occurring in a document is calculated as the product of the weights of the paragraph, the sentence and the word position where the query word occurs. The relevance weight of a document for a query term is the summation of the weights due to the occurrence of the query term at various places in the document according to the method described.

7. Ranking

In ad-hoc monolingual retrieval, the user enters a query describing the desired information. The system then returns a ranked list of documents. There are two main models. Exact match systems return documents that precisely satisfy some structured query expression, of which the best known type is Boolean queries, which are still widely used in commercial information systems. But for large and heterogeneous document collections, the result sets of exact match systems usually are either empty or huge and unwieldy, and so present work has concentrated on systems which rank documents according to their estimated relevance to the query. It is within such an approach that probabilistic methods are useful, and so we restrict our attention to such systems henceforth.

The index file is searched for each query term identified for each query. The relevance weights associated with each matching document corresponding to each query term are summed up to give the relevance weight of a document for the whole query. Finally, the matching documents are ranked on the basis of the relevance weight of the document for each query.

8. Experiments and Evaluation for English and Bengali Ad-hoc Mono-Lingual Task

One run each for English and Bengali was submitted as part of the ad-hoc monolingual retrieval task. All the three parts of a query, namely, *title, description* and *narration* were used for identifying the query terms. The FIRE 2008 organizers provided the relevance judgements. The output of the current system was evaluated with the help of TREC[1] evaluation tool.

[1] Text Retrieval Conference (TREC), http://trec.nist.gov/trec_eval/

The run statistics for the runs submitted to FIRE 2008 are described in Table 3. Only the following evaluation metrics have been listed for each run: mean average precision (MAP), Geometric Mean Average Precision (GM-AP), (document retrieved relevant for the topic) R-Precision (R-Prec) and Binary preferences (Bpref). Clearly the evaluation metrics suggests the lack of robustness in our system. There is certain query topics for which the performance of both the English and the Bengali retrieval systems were quite good.

	MAP	GM-AP	R-Prec	Bpref
English	0.0024	0.0001	0.0111	0.0202
Bengali	0.0200	0.0004	0.0415	0.0583

Table 3 Run Statistics

The unavailability of good stemmers for Indian languages is a reason of our system performance. Simple suffix stripping may not be an ideal case always for highly inflectional Indian languages like Bengali. So, in order to deal with the highly inflective Indian languages we need robust stemmers. The present dynamic zonal indexing technique deals with paragraph zones; for the short document or a document with only one paragraph the score becomes high. As a remedy the sentence number factor is added with the functional equation, but it is not a well known normalization factor. Only a hand-crafted stop word list is used, but a dictionary of Function Word may increase system performance. Other valid reason is word sense disambiguation problem. In Indian languages (except Hindi and Marathi) no WordNet is readily available. So presence of any sense tag might have increased the performance of the system.

9. Conclusion and Future Works

Our experiments suggest that simple TFIDF based ranking algorithms with positional information may not result in effective ad-hoc mono-lingual IR systems for Indian language queries. Any additional information added from corpora either resulting in query expansion could help. Application of certain machine learning approaches for query expansion through theme detection or event tracking may increase our performance. Document-level scoring entailment technique also could be a new direction to be explored. Application of word sense disambiguation methods on the query words as well as corpus would have a positive effect on the result. A robust stemmer is required for the highly inflective Indian languages.

10. REFERENCES

[1] Oard, D.: Alternative Approaches for Cross Language Text Retrieval. In: AAAI Symposium on Cross Language Text and Speech Retrieval, USA (1997)

[2] Dorr, B., Zajic, D., Schwartz, R.: Cross-language Headline Generation for Hindi. ACM Transactions on Asian Language Information Processing (TALIP) 2(3), 270–289 (2003)

[3] Sekine, S., Grishman, R.: Hindi-English Cross-Lingual Question-Answering System. ACM Transactions on Asian Language Information Processing (TALIP) 2(3), 181–192 (2003)

[4] S. Bandyopadhyay et al.: Bengali, Hindi and Telugu to English Ad-Hoc Bilingual Task at CLEF 2007

[5] Pingali, P., Jagarlamudi, J., Varma, V.: Webkhoj: Indian Language IR from Multiple Character Encodings. In: WWW 2006: Proceedings of the 15th International Conference on World Wide Web, pp. 801–809 (2006)

[6] CLIA Consortium: Cross Lingual Information Access System for Indian Languages. In:

Demo/Exhibition of the 3rd International Joint Conference on Natural Language Processing, Hyderabad, India, pp. 973–975 (2008)

[7] Porter, M.F.: An Algorithm for Suffix Stripping. Program 14(3), 130–137 (1980)

[8] Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval, ch. 6. Cambridge University Press, Cambridge (2000)