Theme Based English and Bengali Ad-hoc Monolingual Information Retrieval in FIRE 2010

Pinaki Bhaskar, Amitava Das, Partha Pakray and Sivaji Bandyopadhyay,

pinaki.bhaskar@gmail.com, amitava.santu@gmail.com, parthapakray@gmail.com, sivaji_cse_ju@yahoo.com

Department of Computer Science and Engineering
Jadavpur University, Kolkata-700032, India

Abstract. This paper presents the experiments carried out at Jadavpur University as part of the participation in the Forum for Information Retrieval Evaluation (FIRE) 2010 in ad-hoc mono-lingual information retrieval task for English and Bengali languages. The experiments carried out by us for FIRE 2010 are based on stemming, zonal indexing, theme identification, TF-IDF based ranking model and positional information. The document collection for English and Bengali contained 1,23,047 and 1,25,586 documents respectively. Each query was specified using *title, narration and description* format. 75 queries were used for training the system while the system was tested with 50 queries in each of English and Bengali.

1 Introduction

The Forum for Information Retrieval Evaluation (FIRE) is a forum for Information Retrieval evaluation mainly focused on Indian Languages. the present paper reports about the system that we developed for the ad-hoc monolingual information retrieval for English and Bengali languages. Ad-hoc monolingual Information retrieval involves the study of systems that accept queries (or information needs) in natural language and return objects related to that query. For the present ad-hoc monolingual information retrieval task, the FIRE 2010 organizers provided the corpus and the query sets. Various techniques have been used so far in the area of Monolingual Information Retrieval. These techniques can be broadly classified [1] as controlled vocabulary based and free text based systems at very high level. Some of the earlier systems that were developed for Indian languages include cross language Hindi headline generation [2] and English to Hindi question answering system [3]. In our previous participation in Cross Language Evaluation Forum (CLEF 2007) [4] a semi-automatic query term list was prepared but for the present task an automatic n-gram phrase detection technique has been developed for both the query processing and the document processing tasks. 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. In our previous participation in FIRE 2008 [7] an IR System was proposed based on stemming, zonal indexing; TF-IDF based ranking model and positional information.

2 Corpus Statistics

The corpus for ad-hoc mono-lingual retrieval was made available by the FIRE 2010 organizers. Its objective is to evaluate the effectiveness of retrieval systems in retrieving accurate and complete ranked lists of documents in response to fifty one-time information needs. The FIRE 2010 ad-hoc task focuses specifically on South Asian languages. We participated in the ad-hoc monolingual tasks for English and Bengali.

2.1 Test Data

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). Corpus was sub divided into many other sub-divisions like District, State, Sports, Editorial, and Government etc. Corpus statistics for both the languages, Bengali and English, are in tabular format in Table 1 and Table 2 respectively.

Source Name	Anandabazar Patrika
Source URL	http://www.anandabazar.com/
Time-Period	1st September 2004 - 30th September 2007
Encoding	UTF-8
Total number of documents in the corpus	1,23,047
Corpus Size(MB):	966M (File System ext3)
Markup	<DOC> : Starting tag of a document. <DOCNO> </DOCNO> : Contains document identifier. <TEXT> </TEXT> : Contains document text. </DOC> : Ending tag of a document.

Table 1: Bengali Corpus Statistics

Source Name	The Telegraph
Source URL	http://www.telegraphindia.com
Time-Period	1st September 2004 - 30th September 2007
Encoding	UTF-8
Total number of documents in the corpus	1,25,586
Corpus Size(MB)	580M (File System ext3)
Markup	<DOC> : Starting tag of a document. <DOCNO> </DOCNO> : Contains document identifier. <TEXT> </TEXT> : Contains document text. </DOC> : Ending tag of a document.

Table 2: English Corpus Statistics

2.2 Topics

In FIRE 2010, 50 Bengali topics and 50 English topics are present. Each of these topics is subdivided into four different parts: query identifier (num), a title (title), description (desc), and more details about the topic (narr). Table 3 presents a sample Bengali topic from FIRE 2010.

3 Pre-Processing

3.1 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 are separated in the XML query file

with appropriate tag set as shown in the Table 3.

<top lang="bn"> <num>38</num>
<title> সৌরভ-চ্যাপেল বিরোধের অস্বস্তিকর নিরসন </title>
<desc> ভারতীয় কোচ গ্রেগ চ্যাপেল ও সৌরভ গাঙ্গুলির মধ্যে কলহ সংক্রান্ত নথি খুঁজে বার করো। </desc>
<narr> প্রাসঙ্গিক নথিতে গ্রেগ চ্যাপেল ও সৌরভ গাঙ্গুলির মধ্যে পারস্পরিক অভিযোগ বিনিময়, সংবাদমাধ্যমে চ্যাপেলের বি সি সি আইকে লেখা ই-মেল ফাঁস সংক্রান্ত তথ্য থাকা প্রয়োজন। </narr>
</top>

Table 3: FIRE 2010 Bengali Topic Number 38

3.1.1 Cleaning of Tags

The FIRE 2010 data is well structured with the several tags. The title, the description and the narrative fields that are identified with the tags are extracted from the document. Then all the tags are removed from the documents.

3.1.2 Extract Title words and Keywords

The title and the description have been processed. The stop words and the common words like 'Describe' are removed from all the fields. Proper query words are retrieved from the title field and the list of keywords from the narrative field.

3.2 Corpus Pre-Processing

Both the English and the Bengali news corpus made available as part of FIRE 2010 is 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 removes the suffixes from every word by [8] for English and [9] for Bengali and keeps them in the original order of their occurrence in the document.

4 Theme Clustering

Theme clustering algorithms partition a set of documents into groups or clusters. Documents are described and clustered using a set of theme keywords and values (known as the data representation model). While clustering the documents, they are all distinct as tokens, but multiple documents may have the same representation in this model. So it could be defined as cluster bags. Theme clustering algorithms work over bags of themes like sets except that they allow multiple identical theme words.

4.1 Rule-Based Theme Detection

Term frequency plays a key role in IR to identify document relevance. But in many documents relevant words may not occur frequently or irrelevant words may occur with sufficient frequency. To resolve this, the rule-based theme detection technique has been proposed here. The rules have been devised based on statistics of the corpus. The idea of detecting theme is to identify discourse level most relevant semantic nodes in terms of word or expressions. Theme is a set of significant keywords in the document collection. The crucial features of theme detection are as follows:

4.1.1 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 (IDF). 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. Significant Keywords are identified using TF-IDF, Positional and Distribution factor. The distribution function for each query term in a document is evaluated as follows:

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

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

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

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

Top ranked *n* significant words in each document are identified as theme words. The value of *n* varies according to the size of document. For the present experiment the *n* is the 5% of total document word count. Figure 2 shows document level theme detection.

4.2 Clustering

The categorization task assumes an existing classification, or clustering, of documents. By contrast, the task of document clustering is to create, or discover, a reasonable set of clusters for a given set of documents. As was the case for information retrieval, a reasonable cluster is defined as one that maximizes the within-cluster document similarity, and minimizes between-cluster similarities. There are two principal motivations for the use of this technique in an ad-hoc retrieval setting: efficiency, and the **cluster hypothesis**.

The **cluster hypothesis** [10] takes this argument a step further by asserting that retrieval from a clustered collection will not only be more efficient, but will in fact improve retrieval performance in terms of recall and precision. The basic notion behind this hypothesis is that by separating documents according to topic, relevant documents will be found together in the same cluster, and non-relevant documents will be avoided since they will reside in clusters that are not used for retrieval. Despite the plausibility of this hypothesis, there is only mixed experimental support for it. Results vary considerably based on the clustering algorithm and document collection in use [11].

Applying clustering technique to our three sample documents results in the following term-by-document

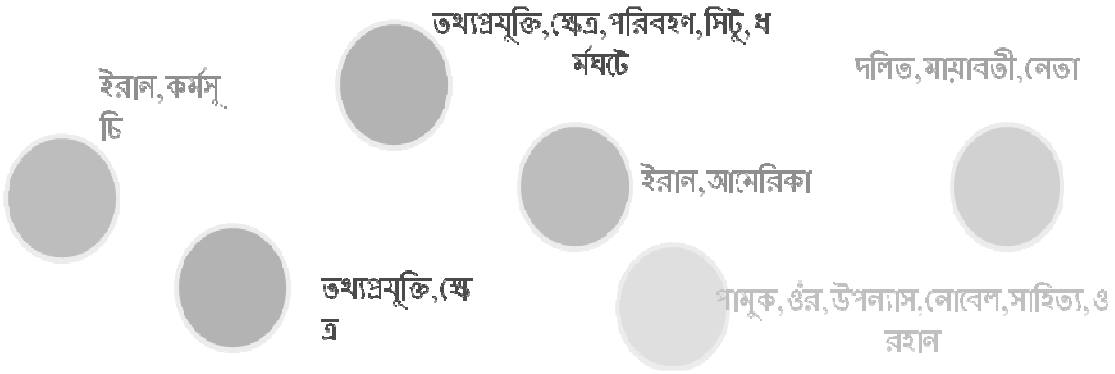

Figure 2: Document level Theme Detection.

matrix, A, where the columns represent Docl, Doc7 and Doc13 and the rows represent the terms politics, sport, and travel.

$$A = \begin{bmatrix} election & cricket & hotel \\ parliament & sachin & vacation \\ governor & soccer & tourist \end{bmatrix}$$

To verify this scheme, the normalized vectors for Doc l and our hypothetical (3, 6, 3) document end up as identical vectors. Now let us return now to the topic of determining the similarity between vectors. Updating the similarity metric given earlier with numerical weights rather than binary values, gives us the following equation.

$$s\left(\vec{q}_k, \vec{d}_j\right) = \vec{q}_k . \vec{d}_j = \sum_{i=1}^{N} w_{i,k} \times w_{i,j}$$

This equation specifies what is known as the dot product between vectors. Now, in general, the dot product between two vectors is not particularly useful as a similarity metric, since it is too sensitive to the absolute magnitudes of the various dimensions. However, the dot product between vectors that have been normalized has a useful and intuitive interpretation: it computes the **cosine** of the angle between two vectors. Note that if for some reason the vectors are not stored in a normalized form, then the normalization can be incorporated directly into the similarity measure as follows.

$$s\left(\vec{q}_k, \vec{d}_j\right) = \frac{\sum_{i=1}^{N} w_{i,k} \times w_{i,j}}{\sqrt{\sum_{i=1}^{N} w_{i,k}^2} \times \sqrt{\sum_{i=1}^{N} w_{i,k}^2}}$$

Of course, in situations where the document collection is relatively static and many queries are being performed, it makes sense to normalize the document vectors once and store them, rather than include the normalization in the similarity metric. Calculating the similarity measure and using a predefined threshold value documents are classified using standard bottom-up hard clustering k-means technique here.

We need a set of initial cluster centers in the beginning. Then we go through several iterations of assigning each object to the cluster whose center is closest. After all objects have been assigned, we recompute the center of each cluster as the centroid or mean $\vec{\mu}$ of its members (see figure 2), that is $\vec{\mu} = \left(1/\left|c_j\right|\right) \sum_{x \in c_j} \vec{x}$. The distance function is the **cosine vector** similarity function here.

Figure 3 illustrates documents after clusters have been formed. Table

4 shows a snapshot of cluster index, which has been used later with a simple similarity measure function to identify the closest document cluster of a particular query

1 Given: a set $X = \left\{ \vec{x}_1, ..., \vec{x}_n \right\} \subseteq R^m$

2 a distance measure $d : R^m \times R^m \rightarrow R$

3 a function for computing the mean $\mu : P(R) \rightarrow R^m$

4 Select *k* initial centres $\vec{f}_1, ..., \vec{f}_k$

5 **while** stopping criterion is not true **do**

6 **for** all clusters c_j **do**

7 $c_j = \left\{ \vec{x}_i \mid \forall \vec{f}_i \, d\left(\vec{x}_i, \vec{f}_j \right) \leq d\left(\left(\vec{x}_i, \vec{f}_i \right) \right) \right\}$

8 **end**

9 **for** all means $\vec{f}_j$ **do**

10 $\vec{f}_j = \mu(c_j)$

11 **end**

12 **end**

Figure 2: The K-means clustering algorithm.

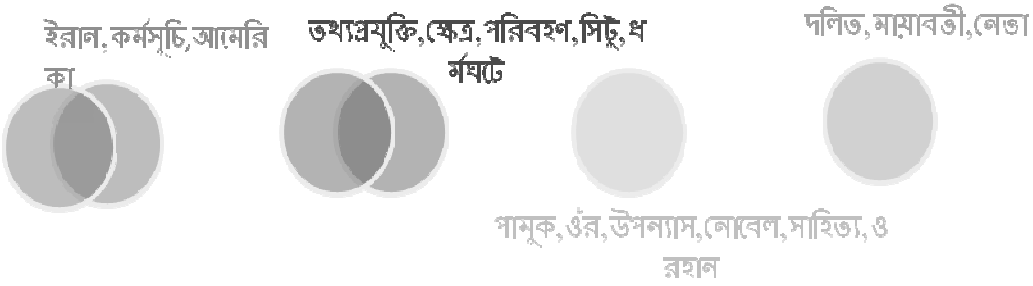

Figure 3: Documents after Clustering

5 Ranking

In ad-hoc monolingual retrieval, the user enters a query describing the desired information. The system then returns a ranked list of documents. The present work has concentrated on systems that rank documents according to their estimated relevance to the query.

The present ranking system works at two levels. Apache Lucene[1], a open source free customized search engine has been used here as the base system. Lucene produces the first level ranking based on standard IR techniques. Ranked document list has been re-ranked using the second method, which is a query focused relative ranking method.

[1] http://lucene.apache.org/

Themes	Doc ID
ইরান, করিয়া, কর্মসূচি, কারণেই, আমেরিকা, পদক্ষেপ	Doc1, Doc78, Doc45, Doc135
তথ্যপ্রযুক্তি, ক্ষেত্র, শিল্প, পরিবহণ, সিটু, ধর্মঘটে	Doc22, Doc177, Doc37
দলিত, রাম, কাঁসি, মায়াবতী, রাজনৈতিক, নেতা	Doc32, Doc56, Doc79, Doc101, Doc83
পামুক, ওঁর, উপন্যাস, নোবেল, সাহিত্য, ওরহান	Doc12

Table 4: Cluster Index of Documents

5.1 Lucene-based System

Lucene is a standard IR system. It works well with bag-of-keywords as an input instead of natural language query. Hence a query expansion technique along with theme detection technique (Section 4.1) has been developed to generate query for Lucene. The generated query can be described as a bag-of-keywords.

The theme detection technique has been applied to the query component *title* and *description*. This technique generates a bag-of-theme expression, which has been supplied to the Lucene as an input query and Lucene finally generates a ranked document list. The ranked document list is then re-ranked by the Query Focused Relative Ranking, described in the next Section.

5.2 Query Focused Relative Ranking

A similarity measure has been developed to compute the nearest theme cluster (Section 4.2) of any query. The expanded query described in Section 5.2 used here to generate the term-by-query matrix (Section 4.2). The normalized cosine similarity measure (Section 4.2) has been used here to calculate similarity distance between term-by-query and term-by-theme matrix. The theme cluster with smallest distance from the term-by-query matrix has been chosen as a desired document set. This document set is then passed through standard IR (Lucene) engine to generate a new rank among them.

Finally document wise ranked score obtained from standard Lucene and by Query Focused Relative Ranking has been accumulated to generate the final ranked document list.

6 Experiments and Evaluation

One run each for English and Bengali was submitted as part of the ad-hoc monolingual retrieval task. First two fields of the three parts of a query, namely, *title, and description* were used for identifying the query terms. The FIRE 2010 organizers provided the relevance judgements. The output of the current system was evaluated with the help of TREC evaluation tool (eval).

The run statistics for the runs submitted to FIRE 2010 are described in Table 5. 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), Binary preferences (Bpref) and Reciprical rank of top relevant document (Recip_Rank).

Scores	Bengali	English
MAP	0.4002	0.4027
GM_AP	0.3185	0.2495
R-Prec	0.3894	0.3873
Bpref	0.3424	0.3479
Recip_Rank	0.6912	0.6773

Table 5: Bengali and English Run Statistics

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.

7 Conclusion and Future Works

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.

Our experiments suggest that simple TF-IDF 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 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.

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. S. Bandyopadhyay, A. Das and P. Bhaskar : English Bengali Ad-hoc Monolingual Information Retrieval Task Result at FIRE 2008. In: Forum for Information Retrieval Evaluation (FIRE), Kolkata, India (2008)
8. Porter, M.F.: An Algorithm for Suffix Stripping. Program 14(3), 130–137 (1980)
9. Manning, C. D., Raghavan, P., Schütze, H. : Introduction to Information Retrieval, ch. 6. Cambridge University Press, Cambridge (2000)
10. Jardine, N. and van Rijsbergen, C. J. (1971). The use of hierarchic clustering in information retrieval. Information Storage and Retrieval, 7, 217-240.
11. Willett, P. (1988). Recent trends in hierarchic document clustering: A critical review. Information Processing and Management, 24(5), 577-597.