DCU at FIRE 2013: Cross-Language !ndian News Story Search

Piyush Arora, Jennifer Foster, and Gareth J. F. Jones

CNGL Centre for Global Intelligent Content
School of Computing, Dublin City University
Glasnevin, Dublin 9, Ireland
{parora,jfoster,gjones}@computing.dcu.ie

Abstract. We present an overview of our work carried out for DCU's participation in the Cross Language !ndian News Story Search (CL!NSS) task at FIRE 2013. Our team submitted 3 main runs and 2 additional runs for this task. Our approach consisted of 2 steps: (1) the Lucene search engine was used with varied input query formulations using different features and heuristics designed to identify as many relevant documents as possible to improve recall; (2) document list merging and re-ranking was performed with the incorporation of a date feature. The results of our best run were ranked first among official submissions based on NDCG@5 and NDCG@10 values and second for NDCG@1 values. For the 25 test queries the results of our best main run were NDCG@1 0.7400, NDCG@5 0.6809 and NDCG@10 0.7268.

Keywords: Hindi Information Retrieval, Cross Language News Search, Query Translation, Query Summarization

1 Introduction

We describe details of DCU's participation in the Cross Language !ndian News Story Search (CL!NSS) task at FIRE 2013 [4]. The CL!NSS task is an edition of the PAN@FIRE task [5] which focuses on addressing news story linking between English and Indian languages. The task is to identify the same news story written in another language, and is thus a problem of cross language news story detection. It can also be interpreted as duplicate detection where the query is a news document and retrieved documents are equivalent news documents but in a different language, see Fig.1.

Fig. 1. Cross Language News Story Detection

The CL!NSS task can be interpreted as a cross-language information retrieval (CLIR) task where the aim is to retrieve a set of news documents which are similar to a query document, but in a different language. For the task there were a total of 50,691 target documents in the Hindi language with 50 documents in the English language with corresponding manually created relevance data in the Hindi document collection available for system development. For the test data there were 25 further English language documents queries for which the task was to find the relevant documents from the Hindi language collection.

Traditional IR methods involve indexing the target documents using a search engine such as Lucene[1] or Terrier[2], and searching over the indexed documents. Since the language for the source queries and target documents are different in CLIR, a means must be found to cross the language barrier. One option is to use publicly available translation services such as Google Translate [3] or Bing[4] to translate the input queries into the target document language. The traditional IR approach can be represented as shown in Fig. 2

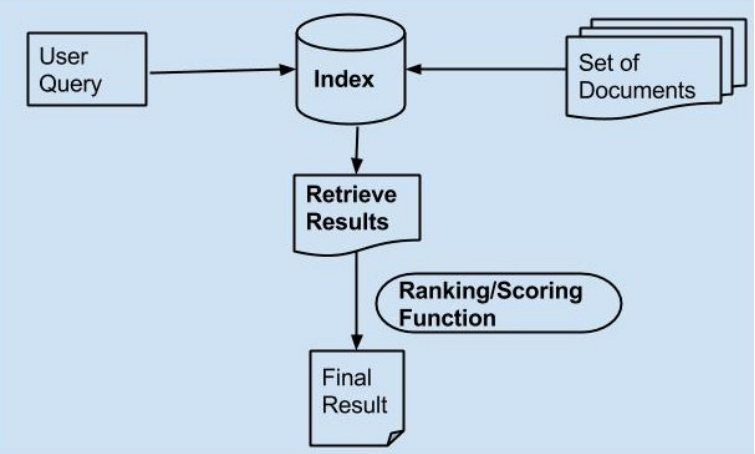

Fig. 2. Traditional IR system

As shown in Fig. 2 the traditional IR system involves 3 main parts-

- Indexing input documents
- Retrieving documents for an input query using different retrieval models
- Scoring the retrieved documents and returning the documents in a ranked order using different metrics

[1] http://lucene.apache.org/core/
[2] http://terrier.org/
[3] http://translate.google.com/
[4] http://www.bing.com/translator

The remainder of this paper is structured as follows: Section 2 outlines our strategy for the CL!NSS task, Section 3 summarizes the task datasets, Section 4 provides a detailed description of our experimental work, Section 5 provides our submission results, and Section 6 concludes the paper with a summary of our work so far and future research plans.

2 DCU Strategy for the CL!NSS task

To address the problem of cross language search for the CL!NSS task we adopted a two step process: in Step-1 we focus on capturing the relevant documents by taking a traditional IR approach while performing query modification, i.e using different heuristics to modify the raw query; in Step-2 we merged the results from the different IR systems and performed re-ranking using the combined scores.

The 2 main steps of our approach are as follows-

- Step 1
  - The input queries were translated separately using Google and Bing translation tools. Our aim in using two translation services was to capture the alternative translation output features of the systems with the objective of maximizing recall in the target document collection, which has been shown to be effective in earlier work on CLIR [6].
  - The Hindi document collection was indexed using the Lucene search library.
  - The translated input queries were searched over the collection of Hindi news documents to retrieve a set of relevant documents.
  - Query modification was performed using different features including:
    - combining translation and transliteration.
    - using a summary of input queries rather than using the complete query document.
    - varying the length of the summary.
    - varying the translation service.
    - using Named Entities information (Person, Location and Organization categories were used).
- Step 2
  - Combining and merging the results of Step-1 using data fusion models.
  - Using a date feature to measure the proximity between the source and retrieved target document.

3 Datasets and Resources

In this section we give a brief overview of the CL!NSS task dataset and the resources we used in this task.

3.1 Training and Test Collections

- **Hindi Document Collection**: The target documents were 50,691 news documents in the Hindi language. All the news documents have 3 main fields: title of the news document, date when the news was published and the content of the news article.
- **English Training Dataset**: The training dataset has 50 documents in the English language. Each of these had 3 main fields (title, date and content) similar to the target documents. The length of training documents varied from a minimum of 4 to a maximum of 68 sentences, with an average length of about 18 sentences
- **English test dataset**: The test dataset had 25 documents in the English language. These also had 3 main fields title, date and content. The length of test documents varied from a minimum of 4 to a maximum of 40 sentences, with an average length of about 14 sentences.

3.2 Resources Used

- **Google Translation**: We used the Google translation service to translate the queries/source document from English to Hindi.
- **Bing Translation**: We also used the Bing translation service to translate the queries from English to Hindi.
- **Summarizer**: We used a summarizer developed at DCU, CNGL [7]. This summarizer scores and ranks the sentences in a document using features as mentioned in [7]. We used the following basic features of the summarizer to generalize our model-
  - skimming: this feature incorporates the position of a sentence in a paragraph. The underlying assumption is that sentences occurring early in a paragraph are more important for a summary.
  - nameEntity: this feature calculates the number of named entities that occur in each sentence. Any word (except the first in a sentence), that starts with a capital letter is assumed to be a named entity.
  - TSISF: this is similar to TF-IDF function but works on sentence level. Every sentence is treated like a document.
  - titleTerm: this feature scores the sentences by matching the overlap with the terms in the title.
  - clusterKeyword: this feature finds the relatedness between words in a sentence.
- **Google Transliteration**- We also used the Google transliteration service to handle words especially Named Entities which are not properly translated by the MT systems.
- **Stanford Core NLP toolkit**- We used the Stanford CoreNLP tool[5] to perform linguistic analysis on the query, including Part-of-Speech (POS) tagging and Named Entity extraction.

[5] `http://nlp.stanford.edu/downloads/corenlp.shtml`

– **Lucene**- We used the open source Lucene search engine library to perform IR, i.e. indexing the input documents and searching the queries over the target collection.

4 Experimental Details/Results

In this section we give full details of the implementation of our system for the CL!NSS task.

4.1 Pre-Processing and Indexing of Target Documents

The input documents were indexed using Lucene. While indexing the documents we used Lucene's inbuilt Hindi Analyzer which performs stopword removal and stemming over the documents. The stopword list we used was obtained by concatenating different standard stopwords list for the Hindi language: i) the FIRE Hindi stopword list[6], ii) the Lucene internal stopword list, and iii) a stopword list created by selecting all the words with document frequency (DF) greater than 5,000 in the target document collection.

4.2 Performing Cross Language Search

To perform cross lingual search the input queries were translated using both Google and Bing translation services. The translated queries were pre-processed using the Hindi Analyzer before being applied for searching over the target document collection.

Table 1. Comparison of baseline runs with FIRE 2012 best result

System	NDCG@1	NDCG@5	NDCG@10	NDCG@20
Palkovski	0.32	0.33	0.34	0.36
Bing	0.54	0.52	0.53	0.55
Google	**0.56**	**0.55**	**0.56**	**0.58**

Table 1 shows results for Lucene search using just the queries translated using Google and Bing translation, and that these perform far better than the best run of the CL!NSS task at FIRE 2012 [5]. However, this performance discrepancy could be explained by the fact that, unlike last year's participants, we had training data for our method which we were able to use during system development.

The baseline system for our experiments are the NDCG values for traditional IR using Google and Bing translation services in isolation, shown in Table 1.

[6] `http://www.isical.ac.in/\~fire/resources.html`

4.3 Main Features for Query Modification

This CL!NSS is quite different from normal CLIR tasks where generally the queries are keywords or statements of information requirement framed as statements ranging about 3-10 words whereas the CL!NSS task has queries which are whole news documents with average length of about 15 sentences.

- **Summarizer** Not all parts of a query document are as important as others to describe the key themes of the document. In fact, some parts of the document distract from the main topical content of the document. To explore this potential problem for the use of a complete news article document as a query, we used a summarizer to score and rank the sentences in a document. The paragraph/sentence content from a document which is more important should be ranked higher. Selecting the top k sentences/paragraphs can give the main representation of a query and prune noise and divergent content. The main question comes in selecting the summary for a given document in terms of its size as we are ranking the paragraphs in a document. To find the optimum length of the summary, we explored a number of alternative summary lengths. Finally we selected the following variants which performed relatively better on the training dataset:
  - Varying length of a summary
    - Summary length is half of the input document length.
    - Summary length is one third of the input document length
    - Summary consist of the top 3 sentences ranked from the input document.
  - Varying translation service: Google and Bing
- **Transliteration** We observed that the Hindi target documents had words which were the translated and transliterated form of input queries as shown in Table 2. The use of the translated or transliterated forms was not predictable, and thus we hypothesized that it is advisable to include both forms in query applied to the IR system. To capture the missing transliteration for the Named Entities in the input queries, we identified them in the input queries and formed a new query consisting of the translated query document supplemented with transliterated Named Entities.

Table 2. Handling Named Entities

English Word	Translated Word	Transliterated Word
Commonwealth Games	राष्ट्रमंडल खेल	कामनवेल्थ गेम्स

- **Using Date** The date of publication of a news article gives an idea of the proximity of another news document. Under the assumption that closer

proximity means that documents are more likely to be related, we gave a small boost of 0.04 to all the retrieved documents which appeared within a window of 10 days before or after the query document. The factor of 0.04 was chosen using a set of experiments to find the optimum performance over the training data.

4.4 Data Fusion

Data Fusion is a well established technique in IR for merging results from multiple retrieval systems or merging results obtained by varying queries and searching over the same system [3]. Each retrieved list of documents has a rank and a score retrieved by the search engine. The scores of document retrieved using different methods or systems need to be normalized before they are combined with the scores retrieved by another system. A standard technique for normalization in data fusion is referred to as the *min-max* method. This is defined as follows:

$$normalized\ \ score = \frac{unormalized\ \ score - minimum\ \ score}{maximum\ \ score - minimum\ \ score}$$

There are 3 standard ways of combining the results across different ranked list which are as follows-

CombSum-1 sum/average: take the average total sum of documents retrieved across different systems and rank the scores accordingly.

CombSum-2 sum/frequency: take the total sum of documents retrieved across different systems and average over the number of systems in which the document was found.

CombMNZ (sum/average)*frequency: take the average total sum of documents retrieved across different systems and multiply by the number of systems in which the document was found.

$$CombMNZ = summation\ \ of\ \ individual\ \ reterieval\ \ results\ * \\ number\ \ of\ \ non\ \ zero\ \ retrievals$$

Data fusion was used in our investigation to combine results of multiple retrieval runs obtained using variations on our query translation and retrieval methods.

4.5 Selected Mechanisms

A range of different features and values were explored for our experiments using the training dataset. Based on our results for this dataset, the following features values shown in Table 3 were selected for our final runs-

– **Using Google Translation**

Table 3. Results of best features on training dataset

System	NDCG@1	NDCG@5	NDCG@10	NDCG@20
Using Google for translation				
One third Summary	0.5408	0.5814	0.5872	0.5907
One third Summary+NE Transliterated	0.5408	0.5757	0.5828	0.5957
3-sentence Summary	0.5918	0.5815	0.5855	0.5897
Complete Query +NE Transliterated	0.5714	0.562	0.5743	0.591
Using Bing for translation				
3-sentence Summary	0.5612	0.556	0.5623	0.5734
One third Summary	0.551	0.555	0.5639	0.5721
Complete Query +NE Transliterated	0.5102	0.5315	0.5463	0.5574

  - Using 1/3 summary of input query.
  - Using 3-sentence summary of input query.
  - Using 3-sentence summary of input query with all Named Entities transliterated using Google transliteration.
  - Using complete input query and all the Named Entities transliterated using Google Transliteration.
- **Using Bing Translation**
  - Using 1/3 summary of input query
  - Using 3-sentence summary of input query
  - Using complete input query and all the Named Entities transliterated using Google Transliteration.

Table 4. System combinations results on training dataset

System	NDCG@1	NDCG@5	NDCG@10	NDCG@20
Run-1	0.5408Ψ	0.5814	0.5872	0.5907
Run-2	0.6224	0.5835	0.5943	0.6022
Run-3	0.6224	0.5733	0.5883	0.5956

We retrieved the top 200 results from each of the 7 features/systems and then we used the CombMNZ data fusion model to combine results from different systems. The top 3 system combinations evaluated over training data are shown in Table 4. These results were created using the following combinations:

- Run-1: Using Google translation and one third summary of queries.
- Run-2: Using Google translation and combining one third summary of queries, 3-sentence summary of queries, one third summary of query with all named entities transliterated using Google transliteration and using whole query with named entities transliterated + incorporating the date factor

– Run-3: Combining all features as discussed in the experimental section i.e including the queries translated using both Google and Bing. Using complete query as well as 1/3 summary and 3-sentence summary of the query with and without NE transliterated all fused together.

These combinations were used for our formal submissions for the CL!NSS task.

5 Test Set Retrieval Experiments and Results

Table 5. Results on test dataset

System	NDCG@1	NDCG@5	NDCG@10	NDCG@20
Run-1	0.74	0.6658	0.6759	0.6849
Run-2	0.74	0.6701	0.7047	0.7042
Run-3	**0.74**	**0.6809**	**0.7268**	**0.7249**
Extra Run-1	**0.74**	**0.6911**	**0.737**	**0.7321**
Extra Run-2	0.74	0.6742	0.7094	0.7092

Our submitted runs were carried out entirely blind, that is, we did not look at the test set prior to applying them to our systems, and used the selected best combinations for the training dataset.

The length of the documents in the training set (average 18 sentences) and testing set (average 14 sentences) is a bit varied. To make sure we don't miss any information, in our 2 additional submission we incorporated one half summary of query with other feature combinations used for the main run-2 and run-3 submissions.

– Extra Run-1: Combining all features of run-3 with one half summary of the queries translated using both Google and Bing with and without NE transliterated.
– Extra Run-2: Combining all features of run-2 with one half summary of the queries translated using Google with and without NE transliterated.

See Table 5 for results of our 3 official run submissions and the 2 extra runs submitted on the testset. Incorporating one half summary of the queries captures more information for the documents in the testset with fewer number of sentences and hence adding this feature shows a slight improvement in the performance over run-3 and run-2 submission.

6 Conclusions and Future Work

The approach of using different features and merging the results performed well and led to retrieval of relevant results with good precision. The results

obtained by our runs ranked first out of the formal submissions for NDCG@5 and NDCG@10, and second for NDCG@1.

There are certain challenges which need to be handled such as dealing with abbreviations such as "MNIK","YSR", movie names, political party names etc. This is a basic problem that needs to be tackled. Handling spelling variants is a significant challenge. Stemming takes care of the affixes. However, the main problem arises with handling the diacritic marks and vowel variations.

In our experiments, we used the standard Lucene scoring function – in Step-1 we would like to explore the alternative scoring functions such as BM25 and other variants. We explored the combination of various features in this study. In further work we plan to try varying the weights of different features rather than simply linearly combining them. We hope to incorporate other techniques for normalizing text, handling the language variations and overcoming the errors made by translated and transliterated tools in later editions of this task.

Acknowledgments. We would like to thank Johannes Levelling and Debasis Ganguly for their suggestions and guidance. This research is supported by Science Foundation Ireland (SFI) as a part of the CNGL Centre for Global Intelligent Content at DCU (Grant No: 12/CE/I2267).

References

1. J. H. Lee: Analyses of Multiple Evidence Combination, In Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 97), pages 267275, Philadelphia, Pennsylvania, USA (1997)
2. N. J. Belkin, P. Kantor, E. A. Fox and J. A. Shaw: Combining the evidence of multiple query representations for information retrieval, Information Processing and Management 31(3):431–448 (1995)
3. W. Bruce Croft: Combining Approaches To Infomration Retrieval, Advances Information Retrieval: Recent Research from the CIIR, Springer (2000)
4. P. Gupta, P. Clough, P. Rosso, Mark Stevenson and R. E. Banchs, PAN@FIRE 2013: Overview of the Cross-Language !ndian News Story Search (CL!NSS) Track. In Proceedings of the Fifth Forum for Information Retrieval Evaluation (FIRE 2013), New Delhi, India (2013)
5. P. Gupta, P. Clough, P. Rosso and M. Stevenson, PAN@FIRE 2012: Overview of the Cross-Language !ndian News Story Search (CL!NSS) Track. In Proceedings of the Fourth Forum for Information Retrieval Evaluation (FIRE 2012), Kolkata, India (2012)
6. G.J.F.Jones and A.M.Lam-Adesina, Exeter at CLEF 2001: Experiments with Machine Translation for Bilingual Retrieval, Proceedings of the CLEF 2001: Workshop on Cross-Language Information Retrieval and Evaluation, Darmstadt, Germany, pages 59-77 (2002)
7. L. Kelly, J. Leveling, S. McQuillan, S. Kriewel, L. Goeuriot, and G. J. F. Jones: Report on summarization techniques, Khresmoi project deliverable D4.4 (2013)