Cross Lingual Information Retrieval (CLIR)

Instrutor :

Arnab Bhattacharya
Department of Computer Science
Indian Institute of Technology, Kanpur
`arnabb@cse.iitk.ac.in`

Group Members:

Hariom Panthi	**Mithlesh Kumar**	**Rishav Raj**
150263	150410	150587
`phari@itk.ac.in`	`mithlesh@iitk.ac.in`	`rishraj@iitk.ac.in`

Abstract

This is the report for the course project of CS657 (Information Retrieval) offered in the 2017-18 II semester. In this project the task is to retrieve relevant documents from an English corpus in response to a query expressed in Hindi. We compared a monolingual English run with a Hindi to English cross-lingual run.

We tried dictionary based approaches, embedding based approaches and combination of two, to map a query in source language into an equivalent query in the language of the target document collection. Then terrier is being used with BM25 model to retrieve documents. On assignment data set, our best cross-lingual performance was 49.22% of the monolingual performance.

1 Introduction

Now a days web is full of multilingual content. Because of this there are many cases where users can read and understand documents written in foreign language but unable to express their information need in that language. This has aroused the need of cross lingual Information Retrieval methods.Since the past decade researchers are looking at ways to retrieve documents in a language in response to a query in another language.The main advantage of CLIR is that the user can search the information without limited by the linguistic barriers.

CLIR is not same as Machine translating(MT) source language query to target language query. Because both of them focuses on different things. MT tries to preserve syntactic as well as semantic features of language while for the purpose of CLIR the translation may not be syntactic correct. CLIR requires simple but correct word to word translation.

2 Problem Statement

In our project the task is to test the performance of systems in retrieving the relevant documents in response to a query in the same and different language from that of the document set. More specifically, given a query in Hindi (source language), the system has to retrieve 1000 relevant documents in English (target language). CLIR is mainly done in two ways, translating the source language query into target language and then retrieve documents or translate the whole corpora in source language and then do the retrieval. But the second approach is not recommended because it will require a lot of resources and time, so we used the first approach.

3 Dataset

We use the data set that was given to us for assignment 1(English) and assignment 2(Hindi). Queries are almost similar in both Hindi and English data set. There were 50 queries in the data set. For learning the word embeddings we use **IITB parallel corpus**[2]. This parallel corpus consist of 14,92,827 sentences in both Hindi and English.

4 Approaches

4.1 Dictionary based approach[1]

A bilingual *dictionary* containing 1,98,572 words is used to translate words from Hindi to English. The dictionary contains phrases(like proverbs) along with individual words.
We first begin by taking the complete query as the key to be searched in the dictionary. If not found, we search for all the queries formed by taking consecutive words which have their lengths as 1 less than the length of the original query. We repeat this process either till we find a phrase which is present in the dictionary and can be translated to it's English counterpart, or till we end up with single words to be dealt with.
Quite a few possibilities arise when we end up getting single words after running the above described algorithm. The best case scenario would be that a one-to-one mapping is present for the concerned Hindi word in the dictionary. Then, the translated word could directly be used as the original word's replacement. But complicacy arises if one Hindi word can be mapped to more than one English word, which is often the case. We opted the following methods to deal with this situation.

- Choose any random English word from the list of candidate words available in the dictionary. This method was opted to form a baseline which we could use to compare the performance of other methods
- Include *k* random English words from the list in the translated query. The intuition behind this is that retrieval accuracy should increase if synonyms are included in the translated query.
- Learn word embeddings for both Hindi and English words simultaneously by using a merged corpus(Hindi and English sentences are shuffle merged) and then select the *top k* English words from the list which are similar to the concerned Hindi word. (Cosine similarity was used for comparison)

The above discussed methods have an assumption in common, *i.e.* we're able to find the Hindi word in the dictionary. But how do we deal with those words which aren't present in the dictionary. One option is to leave these words out. However, this may adversely affect out retrieval system because nouns such as names of people, countries, teams, clubs, *etc.* would be left out and these words may heavily contribute to retrieve the user's required document. Opting against this method, we decided to *transliterate* the words not present in the dictionary.
Transliteration however, has it's own set of problems. Transliteration is basically done by mapping similar auditory characters of one language to another. But due to subtle differences in the transliterated word, the new word obtained may completely miss the mark for it's usefulness(Figure 1). For these reasons, we used a *transliteration dictionary* containing 14,919 words instead of transliterating it via the auditory transformations. The concerned word is transliterated character-wise only when it's not present in the original dictionary as well as the transliteration dictionary.

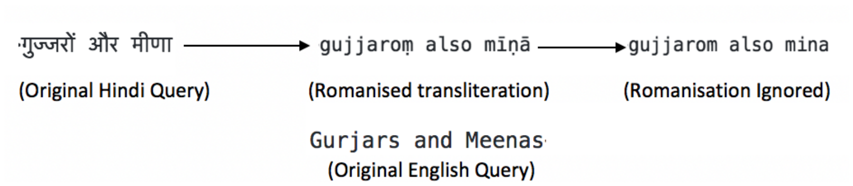

Figure 1: Challenges in transliteration (Both *gujjarom* and *mina* would be unable to retrieve documents related to *Gurjars* and *Meenas*)

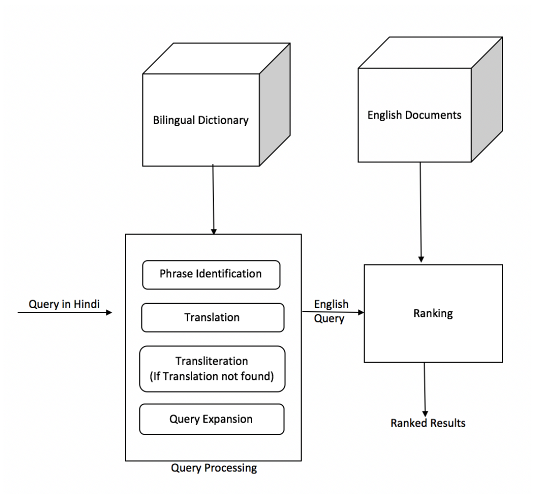

Figure 2: Schematic view of the proposed Dictionary Approach

Algorithm 1 Dictionary based Algorithm

```
1: for each word $h_i$ in source query language do
2:     if $h_i \in$ translation dictionary OR $h_i \in$transliteration dictionary then
3:         if word $h_i \in$ translation dictionary then
4:             retrieve set of all possible translation $E_i$
5:             for each word $e_i \in E_i$ do
6:                 compute cosine-similarity$(h_i, e_i)$
7:             end for
8:         end if
9:         use top k words
10:        if $h_i \in$transliteration dictionary then
11:            use English transliteration of this word
12:        end if
13:    else
14:        use auditory transformation method to do transliteration
15:    end if
16: end for
```

4.2 Embeddings Based Method[3]

We learnt word2vec embeddings of 300 dimensions separately on Hindi as well as English dataset. Using those embedding, we used following two approaches:

1. **Similarity Matrix Based Approach:** We took $\langle h, e \rangle$ pairs from bilingual dictionary, where h and e are Hindi and English words/phrases respectively.We converted these pairs into $\langle f_h, f_e \rangle$ pairs. f_h is feature vector of Hindi word/phrase obtained by taking average of all word embeddings in the phrase h. Similarly, f_e is obtained. We then learnt a similarity matrix W, which minimizes the squared loss of Wh and e. The matrix W, when multiplied with Hindi feature vector f_h transforms it into English feature vector f_e.

   We used this W matrix to translate query in Hindi language to English language by multiplying the embedding of hindi word with W and finding the most similar word in English word embeddings. We put a threshold value of 0.4 to pick the translated word. That is similarity of translated word vector and Wh must be atleast 0.4. If this similarity is less than 0.4 or if the embedding of Hindi query word is not present, we used transliteration to convert it into English language.

2. **Cosine Similarity Retrieval Model**: We converted each English document into feature vector of 300 dimensions by taking the average of embeddings of all words in them. Similarly, we converted each queries into feature vector. For each query, we computed cosine similarity with each documents and retrieved 1000 documents with highest similarity.

$$cosine - similarity(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{ab}}{||\mathbf{a}||||\mathbf{b}||}$$

5 Experiments

We used Terrier 4.2 for retrieving documents in English language using translated queries. We used stopwords removal and Porter Stemmer in English language. We did not use stemming and stopwords removal for queries in Hindi language. We used gensim to trains Word2vec embeddings. We also used default query expansion provided by Terrier for translated queries.

6 Results

Method used:		**MAP**
Dictionary Based	1 English word chosen randomly for each Hindi word	0.1202
	k = 3 English synonyms chosen for each Hindi word if present	0.1348
	Top k similar English words chosen (embeddings from merged corpus) (*k = 1*)	0.1603
	Top k similar English words chosen (embeddings from merged corpus) (*k = 2*)	0.1907
	Top k similar English words chosen (embeddings from merged corpus) (*k = 3*)	0.1025
Similarity Matrix Based (Learnt embeddings for Hindi and Englsih Datasets separately)		0.0536
Cosine similarity Based Retrieval Model (Averaged the embeddings for Query and Documents)		0.0280

Table 1: The MAP accuracy obtained by the original English queries was 0.3874 while the best result obtained by our model was 0.1907 which is just 49.22% of the 0.3874 target. Possible reasons for this observation have been discussed in the conclusion.

7 Analysis of Results/Conclusion

Table 2 shows translations of some Hindi words and cosine similarity with top-4 english words. Except for *mandir*, it is giving wrong translations. For names like *Lalu*, it is giving some random english words as its translation. For *Prasad*, it is giving translation of other prasad (food used as offering in temples). Therefore, it is not able to translate names in Hindi language. One possible fix for this might be the use of Named Entity Recognition(NER) tool in Hindi Language.

Word in Hindi	Word in English	Cosine similarity value
vivad	event	0.65
	phenomenon	0.62
	calamity	0.61
	situation	0.60
mandir	temple	0.73
	tomb	0.70
	statue	0.65
	palace	0.64
Lalu	tent	0.57
	abdomen	0.566
	head	0.563
	sword	0.561
Prasad(name)	devotee	0.58
	disciple	0.57
	lustre	0.568
	darshan	0.563

Table 2: Example to Illustrate Similarity Matrix based approach

From table 1, we see that cosine similarity based retrieval model performs very poorly. One possible reason might be that feature vector of each document is not carrying enough information of original document. Better representations of document features might give better reults.

Inefficiency of all of the above approaches lead to a single problem. They are not able to decide when to transliterate a Hindi word and when to translate it. So, NER in Hindi Language could be used to solve it. As we know, nouns such as names, place, teams etc heavily contribute to fulfill user's information need, So transliteration becomes important factor in CLIR task. In our case we first tried simple character based transliteration algorithm and then added transliteration dictionary along with it. As expected, the second approach gives better results, since in the second case transliteration was more accurate. So main problem of getting unsatisfactory results is that our transliterations are not quite accurate. Most of the times it contains wrong characters. Better transliteration algorithm may improve results. Since in the transliterated word all the character were not wrong. So, indexing using character n-grams might alleviate this problem.

One way to improve accuracy using embeddings based methods is to use huge and diverse dataset to learn embeddings so that out of vocabulary words are as less as possible. Had we been availed with a larger dictionary and a larger dataset, accuracy for both of the methods would have improved. One last factor which works adversely for our translation algorithm is that the dataset used by us has News Headlines as the queries rather than normal conversion type queries which makes the translation difficult as the News Headlines can sometimes be very vague and depend upon the user's intellect for how to decode it's meaning and thereby it's translation.

Acknowledgments

- We have used `https://github.com/sanskrit/sanskrit/blob/master/sanskrit/transliterate/sanscript.py` implementation to do the auditory character based transliteration.

- We have used this bilingual dictionary. `https://github.com/bdrillard/english-hindi-dictionary/blob/master/English-Hindi%20Dictionary.csv`

- We have used this transliteration dictionary. `https://github.com/anoopkunchukuttan/crowd-indic-transliteration-data/blob/master/crowd_transliterations.hi-en.txt`

References

[1] Jagadeesh Jagarlamudi & A Kumaran ,Multilingual Systems Research Microsoft Research India Bangalore, INDIA,Cross-Lingual Information Retrieval System for Indian Languages. 2007

[2] Anoop Kunchukuttan & Pratik Mehta & Pushpak Bhattacharyya The IIT Bombay English-Hindi Parallel Corpus. 2017

[3] Ivan Vulic & Marie-Francine Moens, Monolingual and Cross-Lingual Information Retrieval Models Based on (Bilingual) Word Embeddings. SIGIR'15