Using Word Embeddings for Query Translation for Hindi to English Cross Language Information Retrieval

Paheli Bhattacharya, Pawan Goyal and Sudeshna Sarkar

Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
Kharagpur, West Bengal, India - 721302
paheli@iitkgp.ac.in, {pawang,sudeshna}@cse.iitkgp.ernet.in

Abstract. Cross-Language Information Retrieval (CLIR) has become an important problem to solve in the recent years due to the growth of content in multiple languages in the Web. One of the standard methods is to use query translation from source to target language. In this paper, we propose an approach based on word embeddings, a method that captures contextual clues for a particular word in the source language and gives those words as translations that occur in a similar context in the target language. Once we obtain the word embeddings of the source and target language pairs, we learn a projection from source to target word embeddings, making use of a dictionary with word translation pairs. We then propose various methods of query translation and aggregation. The advantage of this approach is that it does not require the corpora to be aligned (which is difficult to obtain for resource-scarce languages), a dictionary with word translation pairs is enough to train the word vectors for translation.
We experiment with Forum for Information Retrieval and Evaluation (FIRE) 2008 and 2012 datasets for Hindi to English CLIR. The proposed word embedding based approach outperforms the basic dictionary based approach by 70% and when the word embeddings are combined with the dictionary, the hybrid approach beats the baseline dictionary based method by 77%. It outperforms the English monolingual baseline by 15%, when combined with the translations obtained from Google Translate and Dictionary.

1 Introduction

English has been a dominating language of the Web for long but with the rising popularity of the Web, native languages have also found their places - now the Web has substantial content in multiple languages. This prompted the task of Cross Language Information Retrieval (CLIR), where the language of the documents being queried is different from the query language. One of the main motivations behind CLIR is to gather a lot of knowledge from a variety of knowledge bases which are in the form of documents in various languages, helping a diverse set of users, who can provide the queries in the language of their choice.

Intuitively, Cross Language Information Retrieval is harder than Monolingual Information Retrieval because it needs to cross the language boundaries either by translating the query or by translating the document or by translating both the query and

the document to a third language. There are many techniques to implement CLIR. One way to translate the query is a token-to-token translation based approach that uses a machine readable dictionary [1, 7, 17]. Another is to employ Statistical Machine Translation (SMT) systems [22, 24, 25] to translate the query. SMT is a machine translation technique that leverages statistical models whose parameters are derived using parallel bilingual corpora. Other methods for query translation include corpus based techniques [18], using online translation services like Google Translate [27] or by using large scale multilingual resources like Wikipedia [6].

Most of these approaches require either a full fledged dictionary, an aligned corpora or a machine translation system, which may not be guaranteed for resource scarce languages. In this paper, we attempt to solve the problem in a scenario when the monolingual corpus is available in both the languages, but may not be aligned. Additionally, a few word pair translations between the two languages are required, but these need not be exhaustive. We study the effectiveness of word embeddings based methods in this scenario.

In word embeddings, words from the vocabulary are mapped to vectors of real numbers in a low dimensional space; and these vectors are called as embeddings. It has been seen that in the distributed space defined by the vector dimensions, syntactically and semantically similar words fall closer to each other. Given a training corpus, word embeddings are able to generalize well over words that occur less frequently as well. In this paper we try to explore how the usage of word embeddings can affect the retrieval performance in a CLIR based system. To the best of our knowledge, no such approach using comparable corpora has been tried out for the CLIR tasks.

Handling Out-Of-Vocabulary (OOV) terms that are not named entities is a major technical difficulty in CLIR task. For Hindi words that are actually part of the English vocabulary, for example, '*kaiMsara*'[1] (meaning, cancer), '*aspataala*' (meaning, hospital), dictionary and corpus based methods had to resort to "transliteration", but the embedding based method captured their contextual cues and was able to find related words in English. Words brought out as translations for '*kaiMsara*' were 'cancer','disease','leukemia', for '*aspataala*' the words that came out as translations were 'hospital','doctor','ambulance'. We perform transliterations only to handle the named entities.

We also propose and compare various techniques for aggregating the target translations using multiple query terms. We find that instead of aggregating the query vector at the source side, if we compute the similarity scores for each query term separately and then aggregate the resulting vectors, it provides better performance. Our proposed word embedding based approach and the hybrid approach (combined with dictionary) could achieve 88% and 92% of the Mean Average Precision (MAP) as reported by the English monolingual baseline, respectively. When combined with translations obtained from Google Translate, it was able to beat the English monolingual MAP by 15%. The methods also showed improvements of 29%, 34% and 68% over [3], a state-of-the-art corpus based approach.

---

[1] All Hindi words have been written in ITrans using `http://sanskritlibrary.org/transcodeText.html`

2 Related Work

2.1 Cross-Language Information Retrieval

People have tried viewing Cross-Language Information Retrieval (CLIR) from various aspects. To start with, [17] uses dictionary based translation techniques for Information Retrieval. They use two dictionaries, one, in which general translation of a query term is present and the other, in which, domain-specific translation of the query term is present. [9] discusses the key issues in dictionary-based CLIR. They have shown that query expansion effects are sensitive to the presence of orthographic cognates and develop a unified framework for term selection and term translation. [2, 10] perform CLIR by computing Latent Semantic Indexing on the term-document matrix obtained from a parallel corpora. After reducing the rank, the queries and the documents are projected to a lower dimensional space.

Statistical Machine Translation (SMT) techniques and its improvements have also been tried out [20, 22, 24]. [8] uses SMT for CLIR between Indian languages. They use a word alignment table that was learnt using an SMT on parallel sentences to translate source language query to target language query. In [22], the SMT technique was trained to produce a weighted list of alternatives for query translation.

Transliteration based models have also been looked into. [26] uses transliteration of the Out-Of-Vocabulary (OOV) terms. They treat a query and a document as comparable and for each word in the query and each word in the document, they find out a transliteration similarity value. If this value is above a particular threshold, then the word is treated as a translation of the source query word. They iterate through this process, working on relevant documents retrieved in each iteration. [3] uses a simple rule based transliteration approach for converting OOV Hindi terms to English and then uses a pageRank based algorithm to resolve between multiple dictionary-translations and transliterations.

[6] uses Wikipedia concepts along with Google translate to translate queries. The Wikipedia concepts are mined using cross-language links and redirects and a translation table is built. Translations from Google are then expanded using these concept mappings. Explicit Semantic Analysis (ESA) is a method to represent documents in the Wikipedia article space as vectors whose components represent its association with the Wikipedia articles. [23] uses it in CLIR along with a mapping function that uses cross-lingual links to link documents in the two languages that talk about the same topic. Both the queries and the documents are mapped to this ESA space, where the retrieval is performed.

[12] leverages BabelNet, a multilingual semantic network. They build a basic vector represenation of each term in a document and a knowledge graph for every document using BabelNet and interpolate them in order to find the knowledge-based document similarity measure.

Similarity Learning via Siamese Neural Network [29] trains two identical networks concurrently in which the input layer corresponds to the original term vector and the output layer is the projected concept vector. The model is trained by minimizing the loss of the similarity scores of the output vectors, given pairs of raw term-vectors and their labels (similar or not).

[27] uses online translation services, Google and Bing, to translate queries from source language to target language. They conclude that no single perfect SMT or online translation service exists, but for each query one performs better than the others.

2.2 Word Embedding

[13] proposed a neural architecture that learns word representations by predicting neighbouring words. There are two main methods by which the distributed word representations can be learnt. One is the Continuous Bag-of-Words (CBOW) model that combines the representations of the surrounding words to predict the word in the middle. The second is the Skip-gram model that predicts the context of the target word in the same sentence. GloVe or Global Vectors [16] is also an unsupervised algorithm for learning word representations. The training objective of GloVe is to learn word vectors such that for any pair, the dot product equals the log of the words' probability of co-occurrence. They use global matrix factorization and local context window methods to build global vectors.

Word embedding based methods have been utilized in many different tasks, such as word similarity [4, 11, 21], cross lingual dependency parsing [11], finding semantic and syntactic relations [4], finding morphological tags [19], identifying POS and translation equivalence classes [5] and in analogical reasoning [21]. [14] uses the word vectors to translate between languages. Once the word vectors of the two languages have been obtained, it builds a translation matrix using stochastic gradient descent version of linear regression that transforms the source language word vectors to the target language space.

2.3 Word Embedding based CLIR

[28] leverages document aligned bilingual corpora for learning embeddings of words from both the languages. Given a document d in a source language and its comparable document aligned equivalent t in the target language, they merge and randomly shuffle the documents d and t. They train this "pseudo-bilingual" document using word2vec. To get the document and query representations, they treat them as bag-of-words and combine the vectors of each word to obtain the representations of query and document. Between a query vector and a document vector, they compute the cosine similarity score and rank the documents according to this metric.

In this paper, we attempt to perform CLIR from Hindi to English using translations obtained from word embedding based methods. The main advantage of word embeddings is that it does not suffer from data sparsity problems. Given a training corpus, they are able to generalize well over words that occur less frequently. Additionally, they are also computationally efficient [13].

3 The Proposed Framework

We use the query translation approach towards Hindi to English CLIR, that is, we translate Hindi queries to English and perform monolingual information retrieval on English

documents. Towards query translation, we first obtain word embeddings for both the source and target languages using corpus for individual languages. Then, we learn a projection function from source to target word embeddings using aligned word pairs, as obtained from the dictionary. Finally, we employ various methods for query translations: one in which every query term in the source language has *k* best translations in the target language. The second, in which we aggregate the query word vectors into a single vector that represents the query as a whole and then obtain *k* best translations for the query itself.

3.1 Dataset

We have used the FIRE (Forum for Information Retrieval Evaluation, developed as a South-Asian counterpart of CLEF, TREC, NTCIR) 2012 and 2008 datasets obtained from [2]. The FIRE 2012 corpus contains 392,577 English documents (from the newspapers – 'The Telegraph' and 'BDNews 24') and 367,429 Hindi documents (from the newspapers – 'Amar Ujala' and 'Navbharat Times'). For FIRE 2008, we used the same number of English documents[3] and 95,215 Hindi documents (from the Hindi newspaper 'Dainik Jagran'). The corpora are comparable but not aligned. The queries for the CLIR task of FIRE were ranging from topics 176-225 and 26-75 for 2012 and 2008, respectively. We use the title field for the experiments. The English-Hindi dictionary is obtained from `http://ltrc.iiit.ac.in/onlineServices/Dictionaries/Dict_Frame.html`. It also contains translations that were multi-word. We exclude these translation pairs for our experiments. We obtain the stopword list from `http://www.ranks.nl/stopwords/hindi` and English Named-Entity Recognizer from `http://nlp.stanford.edu/software/CRF-NER.shtml`

Next, we discuss in detail various steps in our framework.

3.2 Obtaining Word Embeddings for the Source and Target Languages

We use word2vec introduced by [13]. We train the word2vec package[4] for both the monolingual datasets of English and Hindi. We use the CBOW model with a window size of 5 and output vector of 200 dimensions with other default parameters set.

3.3 Learning the Projection of Word Embeddings from the Source to the Target Language Space

We use linear regression to learn a projection from the source to the target language space, similar to an approach used by [14]. The idea is as follows: Given a translation dictionary, we extract the word embeddings of the translation pair $\{x_i, y_i\}$ where $x_i \in \mathbb{R}^{d_1}$ is a d_1- dimensional embedding learnt from the Hindi corpus for x_i and $y_i \in \mathbb{R}^{d_2}$ is a d_2- dimensional embedding learnt from the English corpus for y_i. The aim is to find

---

[2] `http://fire.irsi.res.in/fire/data`

[3] We could not get the actual English documents for 2008 after repeated trials, so we used the updated dataset of 2012. The actual dataset was a subset of 2012 dataset.

[4] Obtained from `https://code.google.com/p/word2vec/`

a translation matrix W from the source to target such that the root mean square error between Wx_i and y_i is minimized.

After obtaining the translation matrix W using linear regression, embeddings for each word in Hindi (w_h) can be multiplied with W to obtain the equivalent vector v of w_h in the target language space ($v = Ww_h$).

3.4 Query Translation Process

Given a query Q and its terms $q_1, q_2, \ldots, q_n$, we first remove the stop-words from the query. We then use the vector space embedding of each query term q_i, along with the embeddings of all the English words, as obtained using the embedding based method described in Section 3.2, to translate this query, while making use of the translation matrix, obtained in Section 3.3. We adopt the following methods for query translation:

– **Word embedding (WE) to translate each query term independently**: In this approach, once we get the word vector of each query term projected in the target language (v), we compute the cosine similarity between the vector embedding of each English word and v and pick the k best translations for this query term. An example of a query and its 3 best translations is as follows:
  Query in Hindi: *2008 guvaahaaTii bama visphoTa se xati*
  Meaning in English: Loss due to 2008 Guwahati explosions
  The translations of the query terms are given in Table 1. *2008* and *guvaahaaTii* are treated as Named Entities (details in Section 3.5) and hence have one translation each. We see that the WE method gives related words for each query term. We add the translations obtained independently from each query term to obtain the final translation but each term is weighted uniformly.

Table 1: Translations of query terms for "*2008 guvaahaaTii bama visphoTa se xati*" using WE

Query Term in Hindi	Meaning in English	Translations using Embeddings
2008	2008, year	2008
guvaahaaTii	Guwahati, a place in India	Guwahati
bama	bomb	explosives, bomb, device
visphoTa	explosion	explosion, blast, accident
xati	loss	degradation, damage, distortion

– **WE weighted**: Assigning weights to query words is necessary to distinguish between words that are important in a query from words that are not. In this approach, we proportionally distribute the weights according to the similarity score for each translated word with the query word(s). We then normalize the translated query so that the weights for all translations terms add up to 1.
– **Combining Similarity Vectors for Translations (SIM Vec)**: In this approach, instead of treating each query term independently, we aggregate the results by combining results from each query term. One possible way is to combine the vector components at the source[5]. Instead, we first map each query term to the target space,

---

[5] We have tried the sum, max and min combinations, but they do not give good result.

then compute similarity values for each query term with the target words, and combine these similarity values. Thus, for a query word q_j, we build a vector V_j, where the ith component of the vector, $V_j[i]$, denotes the similarity value of that particular word with the ith target language word in the vocabulary. Suppose there are 5 words in the English vocabulary - cricket, football, game, laptop and computer and suppose we want to build the similarity vector of the Hindi word *khela*. The cosine similarity values are listed in Table 2. The similarity vector of *khela* can be written as: $[0.64\ 0.69\ 0.8\ 0.32\ 0.25]$

Table 2: Example to illustrate SIM Vec. The table shows Cosine Similarity Values between the Hindi word *khela* (which means 'game') with other English words.

Word in Hindi	Word in English	Cosine Similarity Value
	cricket	0.64
	football	0.69
khela	game	0.8
	laptop	0.32
	computer	0.25

Now, once we obtain such vectors for each query term, these vector components are merged using the summation or the maximum function. The idea behind using the 'summation' function is to find which words in the target language (English) vocabulary is the most similar when there is a contribution by all the source language query terms. The 'maximum' function provides knowledge as to which word in the target language vocabulary is maximally correlated to any of the source language query terms. The formula for finding the resultant query vector (V_{sum} and V_{max}, for the 'summation' and 'maximum' functions, respectively) from the vectors of the similarity values are shown in Equations 1 and 2. n denotes the number of terms in the query and d denotes the number of words in the target language vocabulary.

$$V_{sum}[i] = \sum_{j=1}^{n} V_j[i] \tag{1}$$

$$V_{max}[i] = \max_{j} (V_j[i])$$
$$\forall j, 1 \leqslant j \leqslant n; \forall i, 1 \leqslant i \leqslant d \tag{2}$$

From the resultant vector, we extract the top *k* target language vocabulary words with the highest scores.

3.5 Transliteration of Named Entities

The source language query also contains named entities, which may not be present in the vocabulary. Since no Named-Entity Recognition (NER) tool is available for Hindi, we resort to the transliteration based process. For each Hindi character, we construct

a table of its possible transliterations. For example, the first consonant in Hindi *ka* has 3 possible transliterations in English – *ka, qa, ca*. We apply several language specific rules - a consonant, for instance *ka* in Hindi can have two forms, one that is succeeded by a silent *a*, i.e., *ka* and another that is not, i.e., *k*. The second case applies when it is succeeded by a vowel or another consonant in conjunction (also known as *yuktakshar*). For each transliteration of an OOV Hindi query word h and for each word e in the list of words returned as named entities in English language, we apply the Minimum Edit Distance algorithm between h and e. We then take the word with the least edit distance. Our transliteration concept is based on [3] and gives quite a satisfactory result, with an accuracy of 90%.

4 Experiments

We used Apache Solr version 4.1 as the monolingual retrieval engine. The similarity score for the query and the documents was the default TF-IDF Similarity[6]. The human relevance judgments were available from FIRE. Each query had about 500 documents that were manually judged as relevant (1) or non-relevant (0). We then used the *trec-eval* tool [7] for finding the Precision at 5 and 10 (P5 and P10) and the Mean Average Precision (MAP).

4.1 Baselines

We use the following baselines for comparison. **English Monolingual** corresponds to the retrieval performance of the target language (English) queries supplied by FIRE. **Dictionary** is the dictionary based method where the query translations have been obtained from the dictionary. For words that contain multiple translations, we include all of them. Translations with multi-words are not considered. Named entities are handled as described in Section 3.5. We also use the method proposed by **Chinnakotla et.al [3]** as a baseline since they participated in the FIRE task [15][8]. Finally, **Google Translate** is also used as a baseline, where the Hindi query is translated using Google Translate to English.

Results for these baselines are reported in Table 3. [3] shows improvements over the dictionary since the OOV terms are transliterated and multiple dictionary translations are disambiguated using the contextual cues from the corpus, however it is not able to perform better than the monolingual baseline. **Google Translate**[9] outperforms the monolingual baselines.

4.2 Proposed Word embeddings based approaches

Table 4 shows the performance of the proposed word embedding based approaches for query translation. Among the proposed approaches, **SIM Vec (max)** seems to perform

---

[6] https://lucene.apache.org/core/3_5_0/api/core/org/apache/lucene/search/Similarity.html
[7] http://trec.nist.gov/trec_eval/
[8] [3] is an improved version of [15]
[9] https://translate.google.com/

Table 3: Performance Results for the Baseline approaches

Method	2012 Dataset			2008 Dataset		
	MAP	P5	P10	MAP	P5	P10
English Monolingual	0.3218	0.56	0.522	0.1609	0.248	0.236
Dictionary	0.1691	0.2048	0.2048	0.084	0.1464	0.137
Chinnakotla et.al [3]	0.2236	0.3347	0.3388	0.11	0.15	0.147
Google Translate	0.3566	0.576	0.522	0.178	0.255	0.24

Table 4: Performance Results when Queries are translated using proposed Word Embedding based methods: for WE and WE weighted, # Translations per query term are shown, while for SIM Vec, # Translations for the complete query are shown.

Method	# Translations	2012 Dataset			2008 Dataset		
		MAP	P5	P10	MAP	P5	P10
WE	1word	0.2533	**0.3920**	**0.3840**	0.1284	**0.175**	**0.163**
	2words	**0.2568**	0.3840	0.3720	**0.129**	0.167	0.154
	3words	0.2379	0.384	0.3520	0.127	0.166	0.152
	5words	0.2053	0.328	0.32	0.119	0.145	0.143
WE weighted	3words	0.2802	**0.436**	0.392	0.138	0.191	0.187
	5words	**0.2808**	0.408	**0.408**	**0.14**	**0.218**	**0.209**
	7words	0.2804	0.428	0.402	0.136	0.21	0.196
SIM Vec	Sum - 15words	0.2508	0.364	0.362	0.1276	**0.2137**	**0.1968**
	Sum - 20words	**0.2562**	**0.368**	**0.368**	**0.1282**	0.2108	0.196
	Sum - 25words	0.2493	0.359	0.343	0.1268	0.187	0.1823
	Max - 10words	0.2733	0.4120	0.382	0.138	0.23	0.225
	Max - 15words	**0.2835**	0.408	0.4	**0.144**	0.2416	0.237
	Max - 20words	0.2830	0.4120	0.392	0.14	**0.2471**	0.238
	Max - 25words	0.2812	**0.424**	**0.394**	0.137	0.24	**0.24**

the best on both the datasets. An issue that comes up while using the embedding based methods is whether to include the embeddings of the named entities in the process. For a particular word in the source language w, similar words that showed up are relevant to w but are not translations. For example, the word *BJP* in Hindi (which is an Indian political party) the words that were most similar also included the names of other political parties like *Congress* and also words like *Parliament* and *government* in the target language English. Inclusion of such terms can harm the retrieval process as named entities play a critical role in Information Retrieval and so we decide to exclude them from the embeddings and use a transliteration scheme as described in Section 3.5

On further investigation, we find that there are 8 such queries for which no translation was available from the Dictionary. Table 5 shows some of these queries. For OOV words that are actually in English and have been written in Hindi orthographic format (e.g, 'housing', 'speaker' and 'cancer' in English have been written as '*haausiMga*', '*spiikara*' and '*kaiMsara*' in Hindi), word embeddings (WE) can easily retrieve translations like 'housing','society' and 'speaker','parliament' and 'cancer','disease' respectively using contextual cues. It is thus evident that the word embedding based method is robust, the translations being very close in meaning to the source language words.

Table 5: Example queries which could not find Translations in the Dictionary but could find Translations using the proposed WE method

Query in Hindi	Translation in English	Translations (WE)	MAP	P5	P10
aadarsha haausiMga sosaaiTii ghoTaale istiiphaa	Adarsh Housing Society scam resignation	Adarsh housing institution scam coterie	0.3	0.6	0.4
bhaaratiiya saMsada aataMkavaadii hamalaa	Indian Parliament attack	Indian Parliament constitutional terrorist assault	0.21	0.6	0.6
aaiiphona aaiipaiDa Dijaaina lokapriyataa lancha	Design Popularity iPhone iPad Launch	iPhone iPad popularity unveiled	0.65	1	1

Table 6: Example queries to illustrate the 'Max' and 'Sum' functions for SIM Vec

Query in Hindi	Translation in English	Translation Method	Translations	MAP	P5	P10
shriilaMkaaii raaShTriiya krikeTa Tiima para hamalaa	Sri Lankan national cricket team attack	Sum	Sri^1 Lankan^1 cricket^0.34 team^0.34 sport^0.32	0.3738	0.6	0.6
		Max	Sri^1 Lankan^1 team^0.35 assault^0.33 attack^0.32	0.51	0.8	0.9
iraaka kaa prathama chunaava	Iraq's first election	Sum	Iraqi^1 choice^0.37 unfashionable^0.32 predictable^0.31	0.08	0	0
		Max	Iraqi^1 elections^0.334 first^0.332 election^0.33	0.4	0.8	0.6
miga durghaTanaa pashchima baMgaala	MiG crash in West Bengal	Sum	MiG^1 West^1 Bengal^1 oriental^0.34 venomous^0.33 exotic^0.33	0.18	0.2	0.2
		Max	MiG^1 West^1 Bengal^1 accident^0.36 mishap^0.33 crash^0.31	0.4	0.8	0.5

When weights are assigned to the translated words, the performance is even better. The insight gained after observing the individual query results for the weighted version is, that it works better for long queries, distributing the weights as per the similarity values.

For SIM Vec, we experimented with both the 'Sum' and 'Max' functions. After doing an analysis on the queries returned by the sum function, we found that those words that are related to the meaning of the entire query come up, while in max, words that have high similarity with one of the query terms, come up in the translation. Table 6 illustrates some example queries from this method. For the first example, 'sum' could

not retrieve words like 'assault' and 'attack', because these were similar only to one query term, '*hamalaa*', but not the others.

While the SIM Vec with the 'Max' function performs the best among the proposed approaches, these results are still inferior to the monolingual baseline as well as Google Translate. Next, we use our proposed method with dictionary based approach as well as Google Translate in a hybrid model.

4.3 Experiments with Hybrid Models

For these experiments, we combine the dictionary based translations or those obtained from Google Translate with translations derived from the embedding based method. The following variations have been tried.

– **Hybrid Translations using Dictionary (WE+DT)**: In this technique of query translation, for each query term q_i, we take translations from the dictionary, if a translation exists. If not, we take its translation from the embedding based methods.
– **Hybrid Translations using Dictionary, weighted (WE+DT weighted, SIM Vec+DT Weighted)**: We assign weights to the dictionary and word embedding based translation words such that the weights for the translations for each of the query terms add up to 1. If a query term has its translation from both dictionary as well as embedding based method, then the dictionary terms are assigned a total weight of w and the rest $1 - w$ is divided proportionately according to similarity values from the embedding based methods. We give 80% importance to the word embedding based terms and 20% importance to the dictionary based terms ($w = 0.2$)[10]
– **Hybrid Translations using Google Translate**
(Google Translate+Sim Vec, Google Translate+Sim Vec+DT): We include query translations from Google, with the same weighting approach as described above.

Table 7 shows the results of the hybrid approaches with dictionary and Table 9 shows these results while using Google Translate with our embedding methods. In both the cases, the hybrid model improves upon the Dictionary / Google Traslate results, obtained when the word embeddings are not used. Specifically, Sim Vec with the Max function performs the best.

Results for some of the individual queries are shown in Table 8. We see that WE, when combined with DT, retrieves many relevant terms, which improve the performance.

From Table 9, we see that our proposed method not only improves upon the dictionary but also improves over Google Translate and English Monolingual. Table 10 summarizes the improvements of our approach over the baselines, to nearest integers. For DT and [3], improvements obtained by our method are shown, while for English Monolingual, we show the % of the E.M. results obtained by our method. We see that all the proposed approaches improve over DT and [3] consistently. Hybrid model with Google Translate improves even on the English monolingual.

---

[10] We experimented with other weightages like 70%-30% and 90%-10% but the 80%-20% division gives the best result. We also experimented with the unweighted version of SIM Vec, but results were better with the weighted version and hence we omit them for brevity.

Table 7: Performance Results when Queries are Translated using a Hybrid of Word Embeddings and Dictionary

Method	# Translations	2012 Dataset			2008 Dataset		
		MAP	P5	P10	MAP	P5	P10
Dictionary	-	0.1691	0.2048	0.2048	0.0804	0.1464	0.137
WE+DT	3words	0.2593	0.404	0.38	0.128	0.168	0.16
	5words	**0.2615**	**0.424**	**0.41**	**0.133**	**0.1835**	0.168
	7words	0.26	0.416	0.397	0.13	0.174	**0.169**
WE+DT weighted	3words	0.2623	0.358	0.35	0.1219	0.208	0.11
	5words	**0.2898**	**0.4920**	**0.49**	**0.147**	**0.22**	**0.218**
	7words	0.2718	0.391	0.39	0.136	0.19	0.18
SIM Vec+ DT weighted	Sum - 15words	0.2835	0.4604	0.457	0.1419	0.237	0.23
	Sum - 20words	**0.2850**	**0.4668**	**0.46**	**0.142**	**0.25**	**0.248**
	Sum - 25words	0.2824	0.4615	0.453	0.14	0.247	0.24
	Max - 15words	0.2965	0.495	0.49	0.148	0.234	0.228
	Max - 20words	**0.2975**	**0.508**	**0.4913**	**0.1486**	0.241	0.236
	Max - 25words	0.2967	0.497	0.485	0.139	**0.25**	**0.248**

5   Conclusion and Future Work

In this paper, we proposed a method based on word embeddings for query translation in the CLIR task. Extensive evaluations performed under various settings confirm that word embedding based method is a potential tool with which the language barrier in the CLIR task can be resolved. It alone performs well over the dictionary method and when combined with the dictionary and Google Translate in a hybrid model, it gives the best performance, improving even the target monolingual baseline by 15%. In future, we will like to repeat these experiments over other source-target language pairs to confirm that this is generalizable across many different language pairs and achieves similar performance gains. We will also study the effect of corpus size (on source and target) as well as the dictionary size on the performance of the system. Finally, we will also experiment using this method for tasks such as bilingual lexical induction.

6   Acknowledgments

Table 8: Example queries to illustrate the hybrid model with word Embeddings and Dictionary

Query in Hindi	Translation in English	Translation Method	Translations	MAP	P5	P10
gorakhaalaiMDa kii maaMga	Demand of Gorkhaland	DT	Gorkhaland	0.197	0.2	0.4
		WE + DT Weighted	Gorkhalandˆ1 demandˆ0.51 demandsˆ0.49	0.88	1	1
abhiyukta ajamala kasaaba	Accused Ajmal Kasab	DT	Ajmal Kasab accused	0.32	0.2	0.2
		WE + DT Weighted	Ajmalˆ1 Kasabˆ1 murderˆ0.26 criminalˆ0.25 murdererˆ0.25 complainantˆ0.24 accused0.2	0.66	0.8	0.8
2003 *aashiyaana kapa vijetaa*	2003 ASEAN Cup winner	DT	2003 ASEAN cup champion victor	0.24	0.4	0.3
		WE + DT Weighted	2003ˆ1 ASEANˆ1 tournamentˆ0.8 cupˆ0.2 winnersˆ0.52 winnerˆ0.48 championshipˆ0.1 victorˆ0.1	0.4	0.6	0.5

Table 9: Performance Results when Queries are Translated using a Hybrid of Word Embeddings, Google Translate and Dictionary

Method	# Translations	2012 Dataset			2008 Dataset		
		MAP	P5	P10	MAP	P5	P10
Google Translate	-	0.3566	0.576	0.522	0.178	0.255	0.24
Google Translate+Sim Vec	10words	0.3669	0.552	0.532	**0.184**	0.266	0.247
	13words	**0.3704**	0.548	0.536	0.1798	0.278	**0.249**
	15words	0.3694	0.532	0.536	0.1737	0.271	0.243
	20words	0.3691	0.552	**0.538**	0.173	0.276	0.235
	25words	0.3699	0.568	0.532	0.1729	**0.284**	0.232
	30words	0.3691	**0.58**	0.53	0.1719	0.28	0.232
Google Translate+ Sim Vec+ DT	10words	0.3682	0.556	0.526	0.1803	0.248	0.236
	15words	**0.3719**	0.56	0.532	**0.1854**	0.2506	0.2404
	20words	0.3699	0.568	0.532	0.1776	0.253	0.2458
	25words	0.3691	0.58	0.53	0.1727	0.2574	**0.2492**
	30words	0.368	**0.588**	**0.544**	0.1703	**0.2626**	0.249

Table 10: Comparison of Word Embedding based methods with Baselines. (DT stands for 'Dictionary' ; [3] refers to Chinnakotla et.al's method ; E.M. stands for 'English Monolingual' ; imp. is 'improvement')

Method		2012 Dataset			2008 Dataset		
		% imp. over DT	% imp. over [3]	% of E.M.	% imp. over DT	% imp. over [3]	% of E.M.
Simple	WE	52	15	80	54	17	80
	WE weighted	66	26	87	66	27	86
	SIM Vec - sum	52	15	80	53	17	80
	SIM Vec - max	**68**	**27**	**88**	**72**	**31**	**89**
Hybrid with Dictionary	WE + DT	55	17	81	58	21	83
	WE + DT weighted	72	30	90	75	34	91
	SIM Vec (sum)+ DT	69	27	89	69	29	89
	SIM Vec (max)+ DT	**76**	**33**	**92**	**77**	**35**	**92**
Hybrid with Dictionary and Google Translate	GT + SIM Vec (max)	119	66	115	119	66	114
	GT + SIM Vec (max) + DT	**119**	**66**	**115**	**120**	**69**	**115**

Bibliography

[1] Ballesteros, L., Croft, B.: Dictionary Methods for Cross-Lingual Information Retrieval. In: DEXA, Zurich, Switzerland (1996)

[2] Chew, P.A., Bader, B.W., Kolda, T.G., Abdelali, A.: Cross-language Information Retrieval Using PARAFAC2. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2007)

[3] Chinnakotla, M.K., Ranadive, S., Damani, O.P., Bhattacharyya, P.: Hindi to English and Marathi to English Cross-Language Information Retrieval Evaluation. In: CLEF 2007, Budapest, Hungary (2007)

[4] Faruqui, M., Dyer, C.: Improving Vector Space Word Representations Using Multilingual Correlation. In: Proceedings of EACL (2014)

[5] Gouws, S., Soggard, A.: Simple task-specific bilingual word embeddings. In: ACL

[6] Herbert, B., Szarvas, G., Gurevych, I.: Combining Query Translation Techniques to Improve Cross-Language Information Retrieval. In: ECIR, Dublin, Ireland (2011)

[7] Hull, D.A., Grefenstette, G.: Querying Across Languages: A Dictionary-based Approach to Multilingual Information Retrieval. In: ACM SIGIR (1996)

[8] Jagarlamudi, J., Kumaran, A.: Cross-Lingual Information Retrieval System for Indian Languages. In: CLEF 2007, Budapest, Hungary (2007)

[9] Levow, G.A., Oard, D.W., Resnik, P.: Dictionary-based Techniques for Cross-Language Information Retrieval. Information Processing and Management 41(3), 523–547 (2005)

[10] Littmana, M.L., Dumais, S.T., Landauer, T.K.: Automatic Cross-Language Information Retrieval Using Latent Semantic Indexing. In: Cross-Language Information Retrieval, 1998 - Springer (1998)

[11] M. Gardner and K. Huang and E.E.Papalexakis and X. Fu and P. P. Talukdar and C. Faloutsos and N. D. Sidiropoulos and T. Mitchell: Translation Invariant Word Embeddings. In: EMNLP, Lisbon, Portugal (2015)

[12] Marc Franco-Salvador, P.R., Navigli, R.: A Knowledge-based Representation for Cross-Language Document Retrieval and Categorization. In: EACL. pp. 414–423 (2014)

[13] Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. In: arXiv preprint arXiv:1301.3781 (2013)

[14] Mikolov, T., V.Le, Q.: Exploiting Similarities among Languages for Machine Translation. In: CoRR, abs/1309.4168, 2013b (2013)

[15] Padariya, N., Chinnakotla, M., Nagesh, A., Damani, O.P.: Evaluation of Hindi to English, Marathi to English and English to Hindi CLIR at FIRE 2008. In: Working Notes of Forum for Information Retrieval Evaluation (FIRE) (2008)

[16] Pennington, J., Socher, R., Manning, C.D.: GloVe: Global Vectors for Word Representation. In: EMNLP (2014)

[17] Pirkola, A.: The Effects of Query Structure and Dictionary Setups in Dictionary-based Cross-Language Information Retrieval. In: ACM SIGIR (1998)

[18] Prasath, R., Sarkar, S., Reilly, P.O..: Improving Cross Language Information Retrieval Using Corpus Based Query Suggestion Approach. In: Proceedings, Part II, 16th International Conference, CICLing, Cairo, Egypt (2015)

[19] Ryan Cotterell, H.S.: Morphological Word Embeddings. In: NAACL (HLT). pp. 1287–1292 (2015)

[20] Schamoni, S., Hieber, F., Sokolov, A., Riezler, S.: Learning Translational and Knowledge-based Similarities from Relevance Rankings for Cross-Language Retrieval. In: ACL 2014 , Baltimore (2014)

[21] Siyu Qiu, Qing Cui, J.B.B.G.T.Y.L.: Co-learning of Word Representations and Morpheme Representations. In: COLING. pp. 141–150 (2014)

[22] Sokolov, A., Hieber, F., Riezler, S.: Learning to Translate Queries for CLIR. In: ACM SIGIR (2014)

[23] Sorg, P., Riezler, S.: Cross-lingual Information Retrieval with Explicit Semantic Analysis. In: Working Notes for the CLEF 2008 (2008)

[24] Ture, F., Lin, J., Oard, D.W.: Combining Statistical Translation Techniques for Cross-Language Information Retrieval. In: COLING (2012)

[25] Ture, F., Lin, J., Oard, D.W.: Looking Inside the Box: Context-Sensitive Translation for Cross-Language Information Retrieval. In: ACM SIGIR (2012)

[26] Udupa, R., K, S., Bakalov, A., Bhole, A.: They Are Out There, If You Know Where to Look: Mining Transliterations of OOV Query Terms for Cross-Language Information Retrieval. In: ECIR 2009, Toulouse, France. pp. 437–448 (2009)

[27] Vahid, A.H., Arora, P., Liu, Q., Jones, G.J.F.: A Comparative Study of Online Translation Services for Cross Language Information Retrieval for Indian Languages. In: Proceedings of WWW '15 Companion Proceedings of the 24th International Conference on World Wide Web. pp. 859–864 (2015)

[28] Vulić, I., Moens, M.F.: Monolingual and Cross-Lingual Information Retrieval Models based on (Bilingual) Word Embeddings. In: ACM SIGIR. pp. 363–372 (2015)

[29] Wen-tau Yih, Kristina Toutanova, J.C.P., Meek, C.: Learning Discriminative Projections For Text Similarity Measures. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning. Association for Computational Linguistics. pp. 247–256 (2011)