ISSN 2007-9737

CMIR: A Corpus for Evaluation of Code Mixed Information Retrieval of Hindi-English Tweets

Kunal Chakma[1], Amitava Das[2]

[1] National Institute of Technology Agartala, Agartala, India

[2] Indian Institute of Information Technology, Sri City, Andhra Pradesh, India

kchakma@nita.ac.in, amitavadas@iiits.in

Abstract. Social media has become almost ubiquitous in present times. Such proliferation leads to automatic information processing need and has various challenges. The nature of social media content is mostly informal. Additionally while talking about Indian social media, users often prefer to use Roman transliterations of their native languages and English embedding. Therefore Information retrieval (IR) on such Indian social media data is a challenging and difficult task when the documents and the queries are a mixture of two or more languages written in either the native scripts and/or in the Roman transliterated form. Here in this paper we have emphasized issues related with Information Retrieval (IR) for *Code-Mixed* Indian social media texts, particularly texts from twitter. We describe a corpus collection process, reported limitations of available state-of-the-art IR systems on such data and formalize the problem of Code-Mixed Information Retrieval on informal texts.

Keywords. Code-mixed social media, code-mixing, code-mixed IR.

1 Introduction

The rapid growth of Internet over the last two decades instigated the proliferation of user generated content (UGC), added many new challenges for automatic text processing due to the informal and noisy nature of UGC, typically social media texts, chats conversations, and instant messages. Recent trends show that such informal texts contain a mixture of two or more languages depending on the geographical locations of the users and based on their proficiencies in the neighboring languages. Such mixing of languages is known as *Code-Mixing* [24]. *Code-Mixing* is found in abundance in the Indian subcontinent and is prevalent in Indian Social Media. People feel ease to type their own language using Roman script or Transliteration due to the unavailability of proper input methods for their languages.

India is a home to several hundred languages. Indeed Indians know and use English but they do mix Indian languages frequently in their social media posts. Hindi is the official language spoken almost by the half of the nation, and it is the 4^{th} most popular language world-wide based on first language speaker [1]. Here in this paper we concentrate on IR problem for English-Hindi *Code-Mixed* text. *Code-Mixed Information Retrieval* (CMIR) is challenging because queries written either in native or Roman scripts need to be matched to the documents written in either or both the scripts. However in this paper, we will focus only on Roman transliterated Hindi mixed with English *Code-Mixed* twitter data. CMIR has to address a non-trivial term matching solution for searching to match each Roman transliterated query terms with the desired word(s), but Roman transliterated form of a Hindi word has no standardization, could have several spelling variations. For example, the Hindi word

[1] https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers

"मैं" ("I" in English) can be written as *me*, *mei*, *mey*, *main*, *mai* and so on. When such a *Code-Mixed* transliterated query search is made, the problem of matching the exact query terms with the terms present in the documents increases due to the spelling variations. Thus, there is a significant effect on the query relevance.

There have been several studies [17, 18, 9, 25] on *Cross Lingual Information Retrieval* (CLIR) and [16, 15, 19] on *Multi Lingual Information Retrieval* (MLIR) including Indian languages. In a CLIR setup users are allowed to make queries in one language and retrieve documents in one or more other languages. MLIR deals with asking questions in one or more languages and retrieving documents in one or more different languages. In either case, the documents and the query are written in their native scripts. For example, a CLIR with English query for a Hindi document might be written in Roman and Devanagari scripts respectively for the query and the document.

Another case is when the query is in one language but written in different scripts. This is called *Mixed Script Information Retrieval* (MSIR) [8]. However, the scenario is different when the query is in Roman transliterated form but contains query terms of different languages. The retrieval task becomes more difficult when the search domain is the social media text such as twitter. This is due to the informal and terse nature of tweets that makes the retrieval task difficult. Moreover, tweets are less likely to be coherent with each other due to the 140 characters limitation. Recently, text mining from Social Media sites such as Facebook and Twitter has taken momentum due to the nature and volume of data being generated (UGC) on the web. The UGC generated on Facebook and Twitter provide information for several business analytics purposes such as mining user's opinion about some products, organizations, sporting events, political campaigns and so on. Organizations can collect such information and perform data analytics for business development. But all the information being generated may not be useful for a particular purpose. Therefore, technology should be able to retrieve only the relevant information being desired. However, Twitter texts being more restrictive in terms of character length, carry information in a very concise manner. The users try to convey the maximum information within the 140 characters limitation of tweets by using phonetic typing, abbreviations, emoticons etc. thus making the retrieval tasks much more difficult. The difficulty further increases when the tweets are written using two or more languages. This is because in the CMIR scenario, the query terms are written in mixed languages and when documents (tweets) are searched, the retrieval task will only search across the documents (tweets) containing the query search terms and rank them accordingly. Existing state-of-the-art technologies have not been designed to handle such type of *Code-Mixed* texts and thus fail to address the issue of CMIR. Therefore, all these issues have motivated us to conduct our studies on IR for such type of informal texts.

In this paper, for the first time, we formally introduce the problem of transliterated *Code-Mixed* query search from twitter for Hindi-English mixed query terms. Present state of the art systems are unable to process *Code-Mixed* transliterated queries due to the lack of resources such as transliterated dictionaries, machine translation systems. Semantic search for transliterated query is still an unsolved problem and it increases many fold when applied on twitter search. Adequate tools are not available to process queries having *Code-Mixed* query terms and existing state of the art systems do not perform well for ranking the documents based on the queries.

The major contributions of this paper are:

— To present the concept, formal definition of *Code-Mixed Information Retrieval* (CMIR) from twitter for Indian languages particularly Hindi-English bilingual texts.

— To create a corpus for Hindi-English *Code-Mixed* tweets.

— To demonstrate how difficult the problem is and where existing IR techniques fail when applied on such data.

The rest of the paper is organized as follows. In Section 2, we discuss about the related works. Section 3 introduces the notion of CMIR formally and outlines the possible applications scenarios and research challenges. Section 4 discusses corpus acquisition process and statistics. In Section 5 the experimental setup and results are presented along with extensive empirical analysis. Finally, in Section 6 we make the concluding remarks.

2 Related Work

Although several studies have been done on CLIR or MLIR, very little attention has been drawn on *Code-Mixed Information Retrieval* (CMIR) from the Social Media domain. The work presented in this paper is mainly motivated from [8] that do discuss the issue of MSIR from Roman transliterated query search for Hindi song lyrics. However, the MSIR setup in [8] did not focus on *Code-Mixed* Social Media texts. Song lyrics whether written in native script or Roman transliterated form, are mostly monolingual in nature whereas *Code-Mixed* tweets are multilingual (bilingual in our case).

The MSIR setup in [8] may not address the true complexity of CMIR scenario presented here since there are inherent difficulties in *Code-Mixed* texts such as identifying the sentence boundaries of text within a tweet. In *Code-Mixed* tweets, one part may be written in one language and another part in another language or there could be *Code-Mixing* at the word level also. Sentence boundary detection (SBD) is itself a difficult problem for such informal texts. Thus, matching the query terms to the languages in the documents is much more difficult. Such characteristics are not found in the cases of either CLIR or MLIR where the languages of query and the documents are implicitly known to the search engine. It is also likely that the problem of spelling variations or out of vocabulary words (OOV) may not be present in a significant manner in CLIR or MLIR. The reason for this argument is that in CLIR, the query and the documents are assumed to be written in their native scripts with spelling variations to be almost negligible.

In contrast, *Code-Mixed* transliterated twitter data is full of noise such as spelling variations, OOV words etc. Identification of Named Entities (NE) plays a crucial role in IR. This is again a difficult task when applied on transliterated tweets because of the spelling variations of the query terms and their presence in the documents. [1] show that due to the lack of standardization in the way a local language is mapped to the Roman script, there is a large variation in spellings which further compounds the problem of transliteration. In [5] they have proposed a query-suggestion system for a Bollywood Song Search system where they have stressed on the presence of valid variations in spelling Hindi words in Roman script. [6] have shown that 90 % of the queries are formulated using the Roman alphabet while only 8% use the Greek alphabet, and the reason for this is that out of every 3 Greek navigational queries, 1 fails due to the low level of indexing by the search engines of the Greek Web.

[7] describe a method to mine Hindi-English transliteration pairs from online Hindi song lyrics crawled from the web. [10] and [21] have used Edit-distance based approaches for the generation of language pairs for Tamil-English and for English-Telugu respectively. [12] developed a stemmer based method that deletes commonly used suffixes. [14] propose a method for normalization of transliterated text that combines two techniques: a stemmer based method that deletes commonly used suffixes with rules for mapping variants to a single canonical form. [20] have proposed a method that uses both stemming and grapheme-to-phoneme conversion for a multilingual search engine for 10 Indian languages. [22] have worked on transliterated search of Hindi song lyrics where they have converted the Roman transliterated words into Devanagari form. [23] use an English taxonomy system to classify non-English based queries which is heavily dependent on the availability of translation systems for the language pairs in question.

It is therefore, observed that though previous studies have attempted to develop transliterated IR systems for Indian Languages (IL) but none of them have addressed the problem with respect to *Code-Mixed* transliterated IR for twitter data. Therefore, the problem of IR with respect to transliterated *Code-Mixed* query search on social media still remains unexplored.

3 CMIR:Formalization

In this section, we formalize the concept of *Code Mixed Information Retrieval* with respect to *MSIR* introduced by [8].

3.1 Code Mixed IR

We assume L to represent the set of natural languages and S to be the set of scripts such that $L = \{l_1, l_2, \cdots, l_n\}$ and $S = \{s_1, s_2, \cdots, s_n\}$ respectively. There could either be a one-to-one or one-to-many mapping between the two sets. Let us assume that a word w written in a language with a particular script be denoted as $\langle l_i, s_j\rangle$. When $i = j$, we say that the word is written in its native script otherwise, it is in transliterated form.

Let q be a query given over a set of documents D where the IR task is to rank the documents in D so that the documents most relevant to q appear at the top. For a monolingual query q, we can assume that $q \in \langle l_1, s_1\rangle$. In MSIR,$q \in \langle l_1, s_j\rangle$ where j may or may not be equal to 1 and therefore, the document pool is denoted as

$$D = \bigcup_{k=1\cdots n} D_{1,k}, \qquad (1)$$

where $D_{1,k} = \{d_{1,k,1}, d_{1,k,2}, \cdots, d_{1,k,N}\}$ are documents in language l_1 written in script s_k, i.e., for all m, $d_{1,k,m} \in \langle l_1, s_k\rangle$. It suggests that in the MSIR setup, the query and the documents are all in the same language, say l_1, which are written in more than one different scripts. Therefore the IR task is to search across the scripts. However, in case of CMIR, query terms belong to different languages which may be either in their native scripts or transliterated form. Therefore, in this case, either $q \in \langle l_i \cup l_j, s_i \cup s_j\rangle$ or $q \in \langle l_i \cup l_j, s_j\rangle$ for bilingual documents. This suggests that the documents are written in either or both the languages l_1 and l_2 in their native scripts s_1 and s_2 respectively or both l_1 and l_2 written in the same script (either s_1 or s_2). It is very less likely that in a practical scenario a query will be written in multiple scripts.

The document pool thus becomes

$$D = \bigcup_{k=1\cdots n} D_{L,k}, \qquad (2)$$

where $L = \{l_1, l_2, \cdots l_N\}$,$D_{L,k} = \{d_{L,k,1}, d_{L,k,2}, \cdots, d_{L,k,N}\}$ are documents in languages L written in scripts s_k. This means for all m, $d_{L,k,m} \in \langle L, s_k\rangle$. Therefore, the IR task in this case is not only to search across the scripts but also across the languages of the query terms and the documents.

3.2 Difficulties and Challenges in CMIR for Social Media Texts

The IR task under the CMIR setup is to search for documents written in multiple languages and rank them according to the query given. CMIR on social media texts introduces several challenges for the IR task. First, it is a difficult task to identify the language of each query terms for a given *Code-Mixed* query across *Code-Mixed* tweets where the document length is short. The second challenge is the presence of spelling variations in transliterated query terms. For example, “मैं” in Hindi which stands for “I” or “me” in English, can be written in any one of the following Roman transliterated forms such as *mein,me,mei,main,mai* and so on. Therefore, for a query q which has one of the given Roman transliterated search terms, only those documents (tweets in our case) in the document collection D will be retrieved which matches with the exact terms. However, there may be documents in D with same query terms with different spellings. A retrieval system in such a scenario will ignore all documents that do not contain the query search term with the exact spellings. Third challenge is to identify words which are represented using mixing of non-alphabetic characters. Tweets are known to be terse in nature and sometimes words could be expressed with non-alphabetic characters. For instance, *“before”* is often written as *“b4”*. Therefore, existing state-of-the-art systems are not appropriate for handling documents of such nature as discussed above.

4 Corpus Creation

4.1 Data Collection

For this study, we have collected Hindi-English tweets where the Hindi terms are in Roman transliterated form. Fetching *Code-Mixed* Hindi-English tweets is itself a difficult task. Tweets were collected on topics related to the events trending at that time. However, it is difficult to decide what is trending. Therefore, we started searching for tweets for events that were happening and making news at the time of collection. For example, for a topic like *Delhi election* which was held in February 2015, we started collecting tweets from November 2014 to April 2015 that is the period from political campaigning to declaration of the election results. For such collection of tweets, the probable query search terms are the entities such as names of the political parties and the candidates contesting the election. Initially we attempted to collect tweets based on the Entities related to the topic. However, it was difficult to retrieve *Code-Mixed* Hindi-English tweets, therefore, we simply issued *bag of words* as the query containing Hindi terms in Roman transliterated form mixed with the topic under consideration. The Hindi search terms are mostly stop words. For example, for tweets related to the *Delhi election*, queries such as *"BJP aur Kejriwal"* (BJP and Kejriwal) were issued. In the example query, *"BJP"* is the name of a political party, *"Kejriwal"* is the name of a candidate and "*aur*" is a Hindi stop word meaning "and" in English. In another example query, "*aam aadmi party Delhi mein*" where "*aam aadmi party*" is the name of a political party and "*mein*" (*"in"*) is the Roman transliterated Hindi word. In similar fashion, tweets related to other trending topics were collected such as *Cricket World Cup 2015* which was held between February and March 2015, the *Bollywood*[2] *controversy* related to an Indian film actor. We also tried to collect tweets from other topics during the period from December 2015 to January 2016.Though several other topics were searched but we did not include them in the corpus because of their smaller presence (limited to 3 to 4 tweets only). We have used *Twitter4j* API [3] for collecting the tweets. Due to the API restriction, we could not fetch tweets beyond seven days thus resulting in limited number of tweets for some topics.

[2]http://https://en.wikipedia.org/wiki/Bollywood
[3]http://twitter4j.org

Tweets were collected based on different query search terms using AND and OR Boolean operators. For example, a search query like "AAP OR aadmi AND hai" which has three terms viz; "AAP", "aadmi" and "hai" was used to fetch tweets containing either "AAP" only or "aadmi" and "hai" both. A total of 9,715 tweets were collected based on different query search terms. It has been observed that there were initially redundant tweets due to the presence of re-tweets, retrieval of the same tweets from two different queries and same tweets with trailing different hash tags, URLs and emoticons. By redundant, we mean in cases where the maximum text of two or more tweets match. Therefore, we have considered them as duplicate tweets and have removed them by a measure known as Jaccard similarity coefficient [4].It is interesting to observe that certain tweets were retrieved which are written in Devanagari script for the Hindi words. In some cases an entire tweet has been found to be written in Devanagari script. We have not considered such tweets for our work and therefore we have manually removed them from our corpus. As our initial step, we have used CMU tokenizer [5] [13] for tokenizing the tweets. Finally, the corpus size was reduced to 9578 tweets out of which 6678 are *Code-Mixed* and the distribution is shown in Table 1

Table 1. Topic wise distribution

Topic	Initial tweets	Code-Mixed tweets
AAP	6096	3994
Delhi elections	851	587
dilli sarkar	120	87
Cricket World Cup 2015	64	51
kejriwal	604	414
national_herald	1083	855
netaji_files	69	51
nirbhaya	233	189
odd_even_delhi	32	32
Salman verdict	426	418
Total tweets	**9578**	**6678**

4.2 Language Identification

For any *Code-Mixed* corpus, it is of utmost importance to identify the languages at the word

[4]https://en.wikipedia.org/wiki/Jaccard_index
[5]http://www.ark.cs.cmu.edu/TweetNLP/

level. Automatic language identification from large multilingual corpora has great significance and therefore, we have used such a system [2] for our purpose. [2] gave an F1 score of 91.1% for automatic language identification on our corpus. The tokens were tagged as `hi,en,ne,acro,univ,undef`. Here `hi` and `en` signifies that a token is either a Hindi or an English word respectively. Likewise, `ne` signifies that a token is a Named Entity, `acro` is an Acronym, `univ` is either an emoticon or a punctuation symbol and `undef` for tokens which are neither `univ` nor `hi` or `en`. An example annotation is shown below:

Tweet:

jaldi delhi .... thoda time aur ... aapni sarkar ko chuno ..... this time 75% voting .... :)

Annotation:

*jaldi/**hi** delhi/**ne** ..../**univ** thoda/**hi** time/**en** aur/**hi** .../**univ** aapni/**hi** sarkar/**hi** ko/**hi** chuno/**hi** ...../**univ** this/**en** time/**en** 75%/**univ** voting/**en** ..../**univ** :)/**univ***

Meaning:

hurry delhi .... little more time ... choose your government ..... this time 75% voting .... :)

Tag wise distribution of the tokens from 6,678 *Code-Mixed* tweets is shown in Table 2. From Table 2 it is observed that Hindi word count is more than the English words in the corpus.

Table 2. Tag wise distribution of tokens in 6678 *Code-Mixed* tweets

Tag	Count	%
hi	72786	50.42
en	22314	15.46
ne	3159	2.19
acro	974	0.67
univ	45100	31.24
undef	29	0.02
Total	**144362**	

4.3 Code Mixing Types

After automatic language identification of the words using [2], we have measured the level of *Code-Mixing* by calculating the *Code Mixing Index* (CMI) introduced in [3] and [4].CMI has been calculated to measure the level of mixing between Hindi and English in our corpus. In our corpus both intra and inter-sentence level *Code-Mixing* are present. We have not measured their percentage distribution because categorizing them requires *Sentence Boundary Detection* (SBD) which is itself a very difficult task. We therefore only report the distribution of *Code-Mixed* data in our corpus. However, there is another type of *Code-Mixing* occurring at word level which we have not found in our corpus. Table 3 lists the distribution of monolingual vs. multilingual tweets. We observed that 69.72% of our corpus is *Code-Mixed* and the remaining 30.28% is monolingual.2% of the corpus has been found to be of other monolingual type.A tweet is considered as “other monolingual” if there are no `hi` or `en` words.

Table 3. Type of *Code-Mixing* in our corpus

Code-Mixed(CM) Type	Count	%
Total Multilingual (CM)	**6678**	**69.72**
Hindi Monolingual	2861	29.87
English Monolingual	37	0.38
Other Monolingual	2	0.02
Total tweets	**9578**	

5 Experiments and Results

5.1 Creation of Gold Standard Data

From the 6678 *Code-Mixed* tweets, we have taken 1959 tweets and conducted two annotation experiments. The reason for choosing the 1959 tweets for creating gold standard is that for our experiments, the chosen set of queries from few topics constitute only 1959 tweets. For the annotation purpose we could not deploy annotators from Amazon Mechanical Turk (AMT) due to financial constraints. Therefore, two in-house students studying Masters in Computer Science and Engineering, were selected for the annotation work who are well conversant with both Hindi and English languages. In our first experiment, each annotator was given a topic with query related to the topic and the list of tweets. Annotators were then requested to rank the tweets according to its relevance for the given query without consulting each other. The agreement score in the first experiment was very low with a Kappa[6] value of 0.1312 only. After carefully investigating the queries and the corresponding tweets, we have found that for some queries it was comparatively easier to rank them whereas for some other queries, the ranking based on relevance judgement was difficult. It was difficult because the collection of tweets for certain queries were semantically diverse from each other with respect to the query. Therefore, in our second experiment, both the annotators were instructed to understand the expected outcome of a given query and to discuss the relevance of a tweet accordingly. After conducting the second experiment, we achieved inter-annotator agreement with a Kappa value of 0.6762. This suggests that relevance judgement for tweets is itself a very difficult task due to their terse nature.

5.2 Searching and Ranking

For evaluation of existing IR systems on our corpus of 1959 gold data, we have used Lucene[7] for indexing, searching and retrieval. Lucene combines *Boolean model (BM)* with *Vector Space Model (VSM)* of Information Retrieval where each distinct index term is a dimension, and weights are tf-idf values and uses the *practical scoring function* [8]. We tested with 20 queries with an average query length of 2.67 terms and compared them against the gold data. For our experimental purposes, *Boolean queries* with AND and OR operators and queries as phrases were issued for searching and retrieving the tweets.

[6]https://en.wikipedia.org/wiki/Cohen%27s_kappa

[7]https://lucene.apache.org

[8]https://lucene.apache.org/core/3_6_0/api/all/org/apache/lucene /search/Similarity.html

We have configured Lucene to retrieve the top 20 tweets for each query. Each tweet has been considered as a document for indexing. Therefore, for 1,959 tweets there are 1,959 documents that were indexed. The Lucene Standard Analyzer is capable of preprocessing (tokenization, stop word removal) only English texts. A total of 11,565 terms were indexed with three fields (contents, file name, file path). The top ten terms that Lucene identified are shown in Table 4. It is interesting to observe that most of these terms are actually Hindi stop words such as *hai,ki,ko,ka*.The Standard Analyzer could not detect the presence of Hindi stop words in their Roman transliterated form thus affecting the ranking. Also noted in Table 4 is that the frequency of relevant indexed terms such as *kejriwal,sonia,khan* etc. are comparatively low. An indexed term is considered relevant if it is close to the topic. For example, terms like *kejriwal,bjp,modi* etc. are considered relevant for the tweets related to Delhi election.

Table 4. Top 10 indexed terms and other relevant terms among top 100 terms

Term	Frequency	Relevant Term	Frequency
hai	1165	national	288
ko	740	herald	284
ki	514	khan	155
soniarahulhazirho	474	court	129
jail	466	aap	116
kejriwal	440	congress	115
salman	420	bhai	114
ke	375	desh	106
ka	344	nirbhayarapistout	98
kya	316	sonia	95

5.3 Results

For evaluating the performance of Lucene on our gold corpus of 1,959 tweets, we initially noted the Mean Average Precision (MAP) of all the tweets ranked by Lucene for the queries against the gold standard. The MAP value we obtained is 4.254359863552588 x 10^{-5} which is very low in terms of performance. This suggests that relevance judgement for ranking informal texts such as tweets, is a difficult task because relevance judgement is idiosyncratic in nature and therefore, creating gold standard data for IR is biased. Therefore, MAP is more likely to be low if there is a huge disparity between the system rankings and the gold rankings. Moreover, due to the terse nature of tweets, it is difficult to judge whether a particular tweet is more relevant than the other for a query. Therefore, we instead took a different approach for evaluating Lucene on our corpus. In our alternate approach, we first issued exactly the same queries to Lucene that were used to collect the tweets. Again we conducted two rounds of annotations. In the first round, tweets that were ranked by Lucene were given to the annotators for manual re-ranking. The inter-annotator agreement was then observed to be 0.7458, which is higher in comparison to the earlier annotation agreement score of 0.6762. In the second round, both the annotators were requested to come to a consensus and create the gold standard. Finally, gold data was created after manually re-ranking the system (Lucene) retrieved tweets. Queries have differing number of relevant documents. Therefore, we cannot use one single cut-off level for all queries. This would not allow systems to achieve the theoretically possible maximal values in all conditions. We have therefore, measured the 11-point average precision [11] and Mean Average Precision (MAP) for the Vector Space Model based ranking mechanism. A total of 20 queries were issued and finally a MAP value of 0.186 was obtained which is reported in Table 5.

Table 5. 11-point Average Precision and MAP

Recall	Precision
0.025	0.1
0.275	0.179
0.4	0.202
0.625	0.205
0.65	0.175
0.65	0.153
0.75	0.133
0.75	0.121
1.0	0.125
1.0	0.118
1.0	0.112
MAP	**0.186**

6 Discussion and Conclusion

Previous works on multilingual IR such as [8] have addressed the issue with respect to retrieval of Hindi song lyrics written either in Devanagari script or Roman transliterated form of Hindi words or a mixture of both. But the work in [8] did not consider the case of documents where the text is *Code-Mixed*. In this paper we have addressed the problem of retrieving *Code-Mixed* Hindi-English documents from informal texts such as tweets where the nature of the text is unpredictable. It is unpredictable due to the informal nature of tweets where spelling variations, out of vocabulary words are often present. We have also stated the difficulties in creating gold standard data for CMIR corpus and established that relevance judgement for ranking tweets is a difficult task as inter-annotator agreement is very low. Without any modification to the original tweets in our corpus, we have indexed the *Code-Mixed* tweets in Lucene with the standard configurations, which Lucene ranked them using its *Practical Scoring Function*. We have adopted relevance feedback mechanism for our evaluations due to the fact that it assumes that the user issuing the queries has sufficient knowledge about the documents desired. This is due to the fact that tweets are very short in length with only 140 characters. Therefore, two tweets may carry the same information but written in different contexts thus making it difficult for human judges to score the relevance. For relevance judgement, sufficient knowledge about the documents is desired. Our experiments suggests that relevance judgement for CMIR does not improve the performance of the ranking mechanism. This is because relevance judgement is biased and it is not appropriate in certain cases where the data has “misspellings”, “spelling variations” and “Mismatch of searcher‘s vocabulary versus collection vocabulary”. Tweets are terse and noisy in nature. Therefore, misspellings and spelling variations are found in abundance in such informal texts and thus make the IR tasks more difficult. In future, we can extend our work with other evaluation measures such as Query Expansion, semantic search. Global relevance evaluation mechanisms such as Query Expansion could be applied on our corpus to measure the performance of searching and ranking. It is clear that state-of-the-art techniques do not perform well on CMIR and therefore, our stated problem opens new research challenges in the *Code-Mixed Social Media IR* domain.

Acknowledgments

We thank our Post Graduating students Mr. Bidipto Chakraborty and Ms. Priyadarshini Lamabam for helping us with the annotation and validation process.

References

1. **Ahmed, U. Z., Bali, K., Choudhury, M., & VB, S. (2011).** Challenges in designing input method editors for Indian languages: The role of word-origin and context. *Proceedings of WTIM*, pp. 1–9.
2. **Barman, U., Das, A., Wagner, J., & Foster, J. (2014).** Code mixing: A challenge for language identification in the language of social media. *Proceedings of The First Workshop on Computational Approaches to Code Switching*, pp. 13–23.
3. **Das, A. & Gämback, B. (2014).** Identifying languages at the word level in code-mixed indian social media text. *ICON 2014 Conference Proceedings*, pp. 84–89.
4. **Das, A. & Gämback, B. (2014).** On measuring the complexity of code-mixing. *Proceedings of the 1st Workshop on Language Technologies for Indian Social Media (SOCIAL-INDIA)*, pp. 1–8.
5. **Dua, N., Gupta, K., Choudhury, M., & Bali, K. (2011).** Query completion without query logs for song search. *Proceedings of WWW (Companion Volume)*, pp. 31–32.
6. **Efthimiadis, E. N., Malevris, N., Kousaridas, A., Lepeniotou, A., & Loutas, N. (2009).** Non-English web search: an evaluation of indexing and searching the Greek web. *Information Retrieval, Springer Link*, Vol. 12, No. 3, pp. 352–379.
7. **Gupta, K., Choudhury, M., & Bali, K. (2012).** Mining Hindi-English transliteration pairs from online Hindi lyrics. *Proceedings of LREC*, pp. 2459–2465.
8. **Gupta, P., Bali, K., Banchs, R. E., Choudhury, M., & Rosso, P. (2014).** Query expansion for mixed-script information retrieval. *SIGIR 2014 Conference Proceedings*, pp. 667–686.
9. **Jagarlamudi, J. & Kumaran, A. (2007).** Cross-lingual information retrieval system for indian languages. *Proceedings of 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007*, pp. 80–87.
10. **Janarthanam, S. C., Subramaniam, S., & Nallasamy, U. (2008).** Named entity transliteration for cross-language information retrieval using compressed word format mapping algorithm. *Proceedings of iNEWS*, pp. 33–38.
11. **Manning, C. D., Raghavan, P., & Schütze, H. (2008).** *Introduction to Information Retrieval*. Cambridge University Press, Cambridge.
12. **Oard, D. W., Levow, G.-A., & Cabezas, C. I. (2000).** Clef experiments at maryland: Statistical stemming and backoff translation. *Proceedings of CLEF*, pp. 176–187.
13. **Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., & Smith, N. A. (2013).** Improved part-of-speech tagging for online conversational text with word clusters. *Proceedings of NAACL 2013*, CMU, pp. 380–390.
14. **Pal, D., Majumder, P., Mitra, M., Mitra, S., & Sen, A. (2008).** Issues in searching for indian language web content. *Proceedings of iNEWS*, pp. 93–96.
15. **Peters, C. (2006).** The impact of evaluation on multilingual information retrieval system development. *Proceedings of Language Resources and Evaluation (LREC)*, pp. 675–678.
16. **Peters, C., Braschler, M., & Clough, P. (2012).** *Multilingual Information Retrieval: From Research To Practice*. Springer-Verlag Berlin Heidelberg.
17. **Pirkol, A., Hedlund, T., Keskustalo, H., & Järvelin, K. (2000).** Cross-lingual information retrieval problems: Methods and findings for three language pairs. *ProLISSa Progress in Library and Information Science in Southern Africa. First biannual DISSAnet Conference*, pp. 01–16.
18. **Pirkola, A., Hedlund, T., Keskustalo, H., & Järvelin, K. (2001).** Dictionary-based cross-language information retrieval: Problems, methods, and research findings. *Journal of Information Retrieval,Kluwer Academic Publishers*, Vol. 4, No. 3-4, pp. 209–230.
19. **Rahimi, R., Shakery, A., & King, I. (2015).** Multilingual information retrieval in the language modeling

framework. *Information Retrieval Journal,Springer*, Vol. 18, No. 3, pp. 246–281.

20. **Raj, A. A. & Maganti, H. (2009).** Transliteration based search engine for multilingual information access. *Proceedings of CLIAWS3*, pp. 12–20.
21. **Sowmya, V. B. & Varma, V. (2009).** Transliteration based text input methods for Telugu. *Proceedings of ICCPOL*, pp. 122–132.
22. **Verulkar, P., Balabantray, R. C., & Chakrapani, R. A. (2015).** Transliterated search of Hindi lyrics. *International Journal of Computer Applications*, Vol. 121, No. 1, pp. 32–37.
23. **Wang, X., Broder, A., Gabrilovich, E., Josifovski, V., & Pang, B. (2008).** Cross-lingual query classification: A preliminary study. *Proceedings of iNEWS*, pp. 101–104.
24. **Wikipedia (2015).** *Code-Mixing*. Wikipedia.
25. **Xu, J. & Weischedel, R. (2000).** Proceedings of 2000 joint sigdat conference on empirical methods in natural language processing and very large corpora. *Proceedings of 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pp. 95–103.

Kunal Chakma is Assistant Professor in Department of Computer Science and Engineering at National Institute of Technology Agartala, India.

Amitava Das is Assistant Professor in Department of Computer Science and Engineering at Indian Institute of Information Technology Sri City, Andhra Pradesh, India.

Article received on 15/02/2016; accepted on 04/03/2016.
Corresponding author is Kunal Chakma.