Retrievability of Code Mixed Microblogs

Debasis Ganguly
ADAPT Centre, School of Computing
Dublin City University
Dublin, Ireland
dganguly@computing.dcu.ie

Ayan Bandyopadhyay
CVPR Unit
Indian Statistical Institute
Kolkata, India
bandyopadhyay.ayan@gmail.com

Mandar Mitra
CVPR Unit
Indian Statistical Institute
Kolkata, India
mandar@isical.ac.in

Gareth J.F. Jones
ADAPT Centre, School of Computing
Dublin City University
Dublin, Ireland
gjones@computing.dcu.ie

ABSTRACT

Mixing multiple languages within the same document, a phenomenon called (linguistic) code mixing or code switching, is a frequent trend among multilingual users of social media. In the context of information retrieval (IR), code mixing may affect retrieval effectiveness due to the mixing of different vocabularies with different collection statistics within a single collection of documents. In this paper, we investigate the indexing and retrieval strategies for a mixed collection of documents, comprising of code-mixed and the monolingual documents. In particular, we address three alternative modes of indexing, namely (a) a single index for the two sub-collections; (b) a separate index for each sub-collection; and (c) a clustered index with two individual sub-collection statistics coupled with the overall one. We make use of the expected *retrievability* scores of the two classes of documents to empirically show that indexing strategies (a) and (b) mostly retrieve the monolingual documents at top ranks with standard retrieval approaches. Our experiments show that, by contrast, the clustered index (c) is able to alleviate this problem by improving the retrievability of the code-mixed documents.

CCS Concepts

•**Information systems** → **Content analysis and feature selection;**

Keywords

Microblog Retrieval; Code Mixing; Retrievability; Fusion

1. INTRODUCTION

Mixing multiple languages within the same document, a phenomenon called (linguistic) code mixing or code switching, is a popular trend among multilingual social media users [11]. Although, code mixing can involve a switch between languages within a document, sentence or even a word [3], and hence can be classified into further fine-grained categories, in our study we treat all these cases equivalently. As an example of code mixing from Twitter[1], consider the following tweet, where for illustration purposes, we have annotated each word with its source language, namely EN (English), BN (Bengali), and NE (a named entity), along with a candidate English translation.

> **Tweet:** Not/EN so/EN soon/EN re/BN arko/NE dayitto/BN nebe/BN I/EN guess/EN
> **English translation:** Hey, not so soon. Arko will take responsibility, I guess.

We see that Bengali words, such as 'dayitto' (responsibility) and 'nebe' (will take), appear in transliterated form interspersed within a primarily English sentence. The prevalence of code mixed content in social media, such as Twitter, has spurred considerable natural language processing (NLP) research activity, such as language identification [10] and part-of-speech (POS) tagging [13]. To the best of our knowledge, no previous research has investigated the effect of code mixed content on information retrieval (IR).

In the context of IR, the code mixing effect can change retrieval effectiveness due to the introduction of a different/diverse vocabulary which can lead to different collection statistics (e.g. document frequency of terms), which in turn can lead to different similarity scores between code-mixed and mono-lingual documents with respect to a given query. Another research challenge arises from the fact that although a significant amount of code mixing can be seen on Twitter, the search queries executed on Twitter are usually not code mixed, i.e., searchers typically do not use code mixing when formulating their queries. This in turn leads to the difficulty of retrieving relevant code mixed tweets at the top ranks due to word mismatch issues. For example, the sample code mixed tweet shown above is less likely to appear within top ranks for a query such as 'Arko responsibility', due to the presence of the Bengali word 'dayitto' in the tweet instead of its English counterpart.

To investigate the effects of code mixing in IR, we seek to address the following research questions:

RQ-1: How frequently can a code mixed document be retrieved within top ranks for a *monolingual* query?

RQ-2: How can the average retrieval rank of relevant code mixed documents be improved?

[1]https://twitter.com/

SIGIR '16, July 17-21, 2016, Pisa, Italy

DOI: http://dx.doi.org/10.1145/2911451.2914727

2. EXPERIMENT SETUP

In this section, we describe the construction process for our experimental dataset of code-mixed and monolingual tweets. This is followed by a description of three alternative indexing approaches on which we execute queries. We then compare the retrievability values [4] of tweets obtained for each of these approaches.

2.1 Dataset Construction

In order to investigate retrievability of code-mixed documents, a document collection comprising of a mixture of monolingual and code mixed documents needs to be constructed. This section elaborates the data collection process.

Geo-tagged Twitter Streaming. Since social media is a prevalent source of code mixed content, we use Twitter as our source of code mixed documents. Since it is more likely to find instances of code mixing in multilingual language dense areas [11], we collected data from India which is predominantly a multi-lingual country with a majority of the population being trilingual. We therefore collected tweets with two types of code mixing, one for Hindi and the other for Bengali.

We used the streaming API of Twitter to collect tweets from three geo-tagged locations referring to three major cities of India, namely Delhi, Mumbai and Kolkata. The streaming API collects all tweets from a given region. Since we are only interested in a mixture of code mixed content and monolingual English content, we filtered out tweets which do not use the Roman script, because such tweets are likely to be written in one of the official languages of India (including Hindi) and hence is less likely to be code mixed. The streaming API was executed for a period of about 6 months, after which a total of about 2.68 million tweets had been accumulated.

Dictionary based Classification. To conduct retrievability experiments, the next step is to determine which documents from this collection are code mixed and which ones are monolingual. Note that we do not need to identify the language of each constituent word to detect the exact code switching points for which supervised approaches such as SVM (bag-of-words based) and CRF (word sequence based) have been reported to work well [5]. Instead, we rather need to detect whether any code mixing exists in a tweet, which is rather a simpler problem. Since the effectiveness of code mixing detection is not the core focus of this paper, we employ a simple dictionary based classification approach to classify the tweets of our collection into one of the two classes, i.e. code mixed or monolingual.

The dictionary that we use for classification is comprised of a list of 154 commonly used words in Hindi and Bengali in the transliterated form. If a tweet contains any of these 154 words, it is classified as a code mixed tweet. We use keywords from these two languages for two reasons. Firstly, these two are the most spoken among Indian languages (Hindi ranks 4^{th} in the world with 310M speakers whereas Bengali ranks 7^{th} with 205M speakers [1]). The second reason is due to the location of the tweets themselves. Bengali is the native language of Kolkata and there are a vast majority of Bengali speaking people in both Delhi and Mumbai, whereas Hindi is spoken with almost native fluency in all the three cities. Consequently, it is more likely that the tweets from these locations will borrow words from the vocabulary of one or both of these languages.

We evaluated the accuracy of code mixing detection on a subset of 100 tweets drawn randomly from each of the two classes, i.e. monolingual and code mixed. We found out that 89 and 95 out of the 100 in each class were truly code mixed and mono-lingual respectively. This shows that the dictionary based method works satisfactorily in the context of our experiments.

Dataset Statistics. Table 1 summarizes the dataset characteristics, from which we can make the following observations. Firstly, it can be seen that about 7.25% of the tweets are code-mixed, i.e. contains a transliterated Bengali or Hindi function word. The percentage of code-mixed tweets is thus significant, and hence the very question on the retrievability of this type of documents is a relevant one. Secondly, the vocabulary of the code mixed documents is more diverse in comparison to the monolingual one. Although the absolute value of the vocabulary size for the code mixed documents is smaller, the average document frequency (DF), computed as the ratio of the number of unique terms to the number of documents, is higher for the code mixed type. Another measure which indicates vocabulary diversity is the TTR (Type-token ratio), which is also higher for the code mixed documents. This indicates that the collection statistics, e.g. DF, of the non-English terms, e.g. hashtags, named entities or transliterated Bengali (BN) or Hindi (HN) words of the code mixed documents, can be significantly different in comparison to monolingual English (EN) documents.

Document type	#Docs	#Vocab	Av len	Av DF	TTR
Monolingual (EN)	2,502,343	8,335,003	10.69	3.30	0.311
Code mixed (BN/HN)	181,282	1,330,727	10.78	7.34	0.681

Table 1: Characteristics of the dataset constructed by streaming tweets from 3 major cities of India.

Query set construction. To execute large scale retrievability experiments, one needs to construct a set of sample queries. To compute average retrievability of documents, [4] uses frequently occurring unigrams and bigrams from the collection vocabulary as as sample queries. In contrast to [4] of selecting all word unigrams and bigrams as the set of queries, in our experiments we restrict this set of queries to all unigrams and bigrams belonging to the set of intersection between the two vocabularies, the query terms thus comprising hashtags, named entities and English words. This is because code mixing, being a linguistic phenomenon, is likely to be present in well-formed English (or non-English) sentences and usually absent in keyword-based user queries.

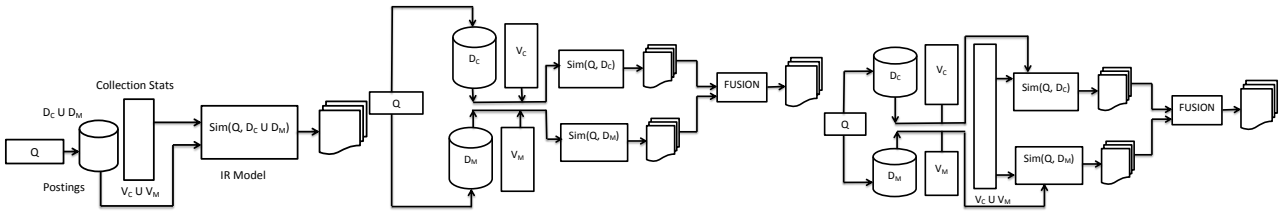
This way of selecting sample queries ensures that no Bengali or Hindi words can appear as a part of the query set because these words do not belong to the monolingual (EN) sub-collection. Note that English words exist in the query samples since due to the code mixing property, these words are a part of the code mixed documents vocabulary. A similar argument also applies for neutral words such as named entities and hashtags because they are a part of both monolingual EN and code-mixed vocabulary.

From the intersection set of the vocabularies of the two sub-collections of Table 1, we selected all unigrams and bigrams having document frequencies higher than 20 (similar to the settings of [4]) as the set of queries. In total, the number of queries used for our retrievability experiments is $36,422$. Similar to [4], we set the prior likelihood factor of a query to the value of 1.

2.2 Indexing and Retrieval

In order to explore the retrievability of a code mixed document, from among a collection of monolingual and other code mixed documents, we compare three different approaches to indexing and retrieval of the mixed dataset described in Section 2.1.

A standard IR model uses collection statistics, e.g. the DF of a term, to estimate the relative importance of a match between a document word with a given query term [6]. Since collection statistics can play an important role in scoring documents against a query, we investigate three different indexing approaches that use the collection statistics in three different ways, described as follows.



(a) Combined Index. (b) Separate Indexes. (c) Clustered Index.

Figure 1: Three different indexing and retrieval strategies for the mixture collection of monolingual and code-mixed documents.

Single Index. This indexing scheme does not require a code-mixing detection classifier. Both types of documents, namely the monolingual and the code-mixed, are stored in a single index without any distinction. Obviously, there is a single collection statistics table for both the vocabulary types, say V_M (monolingual) and V_C (code-mixed). A schematic diagram is shown in Figure 1a. The similarity score between a query Q and a document D is given by a function of the term frequency of a query term t in D, $tf(t, D)$, and the collection statistics of t computed over the set $V_M \cup V_C$, which we denote by $df_{M\cup C}(t)$ in Equation 1.

$$sim(Q, D) = f(tf(t, D), df_{M\cup C}(t)) \tag{1}$$

Split Index. In this indexing scheme, a code-mixing detection approach, e.g. a dictionary based approach, is required to construct separate indexes for each type of document, i.e. EN or code-mixed. Each index maintains its own collection statistics. A query is executed separately on these two indexes and the final result-list can be obtained with a standard fusion technique, e.g. COMBSUM [7]. The schematic diagram of this approach is shown in Figure 1b. The similarity score between a query Q and a document D is given by Equation 2, where t is a query term, D is a document which is either monolingual or code-mixed, i.e., $D \in \{D_M, D_C\}$, and g is a fusion function.

$$sim(Q, D) = g\Big(f\big(tf(t, D_M), df_M(t)\big), f\big(tf(t, D_C), df_C(t)\big)\Big) \tag{2}$$

Clustered Index. In this indexing scheme, schematically shown in Figure 1c, separate indexes are maintained for the two different document classes, similar to the *split index* approach. However, in addition to the local collection statistics, the similarity score also depends on the global collection statistics, $df_{M\cup C}(t)$, similar to the *single index* approach. This is shown in Equation 3, where it can be seen that the similarity score additionally depends on a linear combination of the local (per document type) and the overall collection statistics, parameterized by a number $\alpha \in [0, 1]$.

$$sim(Q, D) = g\Big(f\big(tf(t, D_M), \alpha \cdot df_{M\cup C}(t) + (1-\alpha) \cdot df_M(t)\big), f\big(tf(t, D_C), \alpha \cdot df_{M\cup C}(t) + (1-\alpha) \cdot df_C(t)\big)\Big) \tag{3}$$

Equation 3 is similar in principle to the clustered language model [9]. However, any other retrieval model can be applied in the general scheme of Equation 3 and Figure 1c.

Retrieval Models. In our experiments, for the abstract function f shown in Equations 1, 2 and 3, we apply standard retrieval models, namely, a) BM25 with $k = 1.2$ and $b = 0.75$ [12]; b) LM with $\lambda = 0.4$ [8]; c) DFR with Bose-Einstein model and H_1 normalization [2]); and d) a COMBSUM [7] fusion of all these models.

Evaluation Measures. To measure the average ranks at which monolingual documents are retrieved in comparison to code-mixed ones, we compute the retrievability score of each class of documents as defined in Equation 4, where c is a rank threshold parameter, set to 10 and 20 in our experiments, as suggested in [4].

$$r(d) = \sum_{q \in Q} \mathbb{I}[rank(d, q) \leq c] \tag{4}$$

The expected retrievability, $\hat{r}$, for a particular class of document, i.e. M or C, is then computed by averaging over the computed $r(d)$ values for documents of that class. The Gini coefficient of the $r(d)$ values measures the inequality of the $r(d)$ score distribution, i.e., lower the value, the more uniform the distribution is. In the context of our experiments, higher Gini scores would give the code mixed documents more chance to be retrieved at top ranks.

3. RESULTS

The retrievability scores, $\hat{r}$ values for the two document classes along with the Gini coefficients, G, are reported in Table 2. The highest expected retrievability values for each document class in each index type are shown with bold face, e.g., the DFR model results in highest retrievability of the monolingual documents with the combined indexing approach (Equation 1).

It can be seen that the combined indexing approach results in lower retrievability of the code-mixed documents. On assuming uniform likelihood of a document to be retrieved from any class, the number of possible ways of selecting the top c documents for the monolingual case is much higher than the code-mixed case (from Table 1 it can be seen that the number of monolingual English documents is more than 10 times that of the number of code-mixed ones), which means that a code-mixed document is likely to lose the 'competition' of getting a berth into the top ranks to its monolingual counterpart. With respect to the research question **RQ-1**, it can be commented that standard IR models (and their fusion) with a combined index of monolingual and code-mixed documents is not prone to retrieving the code-mixed documents at top ranks.

It can also be observed from Table 2 that the retrievability scores of the code-mixed documents increase with the use of separate indexes and fusing the results. The best results are obtained with BM25. However, the retrievability scores of the code-mixed documents are still lower compared to the mono-lingual documents. This shows that late fusion alone does not suffice to alleviate the bias towards retrieving more number of monolingual documents at top ranks.

Table 2 shows that with the clustered index and the use of a combination of collection statistics (Equation 3 with α set to 0.4), the retrievability scores of the code-mixed documents can be increased significantly by the LM retrieval model and the fusion approach. The expected retrievability of the code-mixed documents obtained with LM is in fact higher than that of monolingual documents. However, the BM25 and DFR models continue to favour

Index	Model	Type	c = 10 $\hat{r}$	c = 10 Gini	c = 20 $\hat{r}$	c = 20 Gini
Single	BM25	M	1.2298	0.1649	1.3258	0.2048
		C	1.1790	0.1399	1.2354	0.1656
	LM	M	1.2259	0.1620	1.3150	0.2001
		C	**1.1975**	0.1510	**1.2521**	0.1785
	DFR	M	**1.2484**	0.1732	1.3545	0.2159
		C	1.1758	0.1357	1.2342	0.1633
	FUSION	M	1.2305	0.1645	**1.3944**	0.2352
		C	1.1873	0.1437	1.2482	0.1723
Split	BM25	M	1.2456	0.1725	1.3385	0.2105
		C	**1.2036**	0.1574	**1.2604**	0.1822
	LM	M	1.2493	0.1735	1.3423	0.2112
		C	1.1829	0.1454	1.2455	0.1779
	DFR	M	**1.2559**	0.1768	**1.3527**	0.2150
		C	1.1766	0.1410	1.2416	0.1721
	FUSION	M	1.2431	0.1710	1.3355	0.2089
		C	1.1810	0.1451	1.2477	0.1762
Cluster	BM25	M	1.2796	0.1940	1.3804	0.2319
		C	1.2345	0.2002	1.1450	0.1345
	LM	M	1.2680	0.1878	1.3578	0.2232
		C	**1.4105**	0.2480	**1.5101**	0.2801
	DFR	M	**1.2974**	0.2016	**1.4063**	0.2415
		C	1.1960	0.1739	1.1252	0.1119
	FUSION	M	1.2798	0.1933	1.3769	0.2298
		C	1.3021	0.2319	1.1867	0.1610

Table 2: Expected Retrievability scores of the monolingual (M) and the code-mixed (C) documents along with the Gini coefficients. Cluster results were obtained with $\alpha = 0.4$ (Eqn. 3).

the monolingual documents, which shows that these models do not naturally extended to the clustered case as is the case with clustered LM [9]. A bias-variance trade-off is achieved with a fusion of the respective models on the clustered index, where we observe that bias towards any particular class of documents decreases. Going back to research question **RQ-2**, i.e. how can the retrievability of code-mixed documents be improved, it can be commented that the clustered mode of indexing and retrieval seems to be the best choice for retrieval of code-mixed content from within a collection of both code-mixed and monolingual documents.

Figure 2 shows how α, the linear combination parameter of Equation 3, affects the expected retrievability scores of the code-mixed documents. An interesting observation is that the average retrievability scores obtained with LM increase when more importance is given to the collection statistics obtained from the individual clusters ($1 - \alpha = 0.8$) than the global collection statistics ($\alpha = 0.2$).

4. CONCLUSIONS AND FUTURE WORK

In this paper, we studied how code-mixing can affect IR effectiveness. To conduct this study, we constructed a collection of about 2.68M tweets (written using the Roman script), out of which about 181K are code-mixed and the rest are English tweets without code-mixing. We investigated three alternative strategies to index and retrieve content from this collection, a) a single index which combines both types of documents; b) separate indexes for each type with the ranked lists obtained from each being fused; and c) a clustered index (each cluster having its own local collection statistics) coupled with a combined collection statistics table and a fusion of

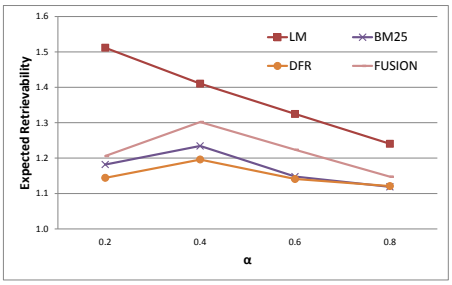


Figure 2: α vs retrievability of code-mixed documents for the clustered index.

the result-lists. We found that the clustered index leads to maximum information access for the code-mixed content with the LM retrieval model. A fusion of LM with BM25 and DFR on the clustered index results in a balanced information access for the two information types, namely monolingual and code-mixed.

In future, we would like to develop query sets with depth pooled relevance judgements on this collection of code-mixed tweets. We would also like to conduct user studies on task driven code-mixed content search to better understand user search behaviour.

Acknowledgements

This research is supported by Science Foundation Ireland (SFI) as a part of the ADAPT Centre at DCU (Grant No: 13/RC/2106) and by a grant under the SFI ISCA India consortium.

5. REFERENCES

[1] List of languages by number of native speakers. https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers. Accessed: 2016-02-09.

[2] G. Amati and C. J. Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389, 2002.

[3] P. Auer. *Code-switching in conversation: language, interaction and identity.* Taylor and Francis, 2002.

[4] L. Azzopardi and V. Vinay. Retrievability: an evaluation measure for higher order information access tasks. In *Proceedings of CIKM*, pages 561–570, 2008.

[5] U. Barman, A. Das, J. Wagner, and J. Foster. Code mixing: A challenge for language identification in the language of social media. In *Proceedings of First Workshop on Computational Approaches to Code Switching*, pages 13–23, 2014.

[6] H. Fang and C. Zhai. An exploration of axiomatic approaches to information retrieval. In *Proceedings of SIGIR '05*, pages 480–487, 2005.

[7] E. A. Fox and J. A. Shaw. Combination of multiple searches. In *Proc of TREC-2*, pages 243–252, 1994.

[8] D. Hiemstra. *Using Language Models for Information Retrieval*. PhD thesis, Center of Telematics and Information Technology, AE Enschede, 2000.

[9] X. Liu and W. B. Croft. Cluster-based retrieval using language models. In *Proc of SIGIR '04*, pages 186–193, 2004.

[10] D. Nguyen and A. S. Dogruöz. Word level language identification in online multilingual communication. In *Proceedings of EMNLP '13*, pages 857–862, 2013.

[11] C. J. Paolillo. "conversational" codeswitching on usenet and internet relay chat. *Language@Internet*, 8(3), 2011.

[12] S. E. Robertson, S. Walker, S. Jones, and M. Hancock-Beaulieu. Okapi at TREC-3. In *Proceedings of the Third Text REtrieval Conference (TREC 1994)*. NIST, 1994.

[13] Y. Vyas, S. Gella, J. Sharma, K. Bali, and M. Choudhury. POS tagging of english-hindi code-mixed social media content. In *Proceedings of EMNLP '14*, pages 974–979, 2014.