Unraveling the English-Bengali Code-Mixing Phenomenon

Arunavha Chanda
a.chanda@columbia.edu
Columbia University

Dipankar Das
dipankar.dipnil2005@gmail.com
Jadavpur University

Chandan Mazumdar
chandanm@cse.jdvu.ac.in
Jadavpur University

Abstract

Code-mixing is a prevalent phenomenon in modern day communication. Though several systems enjoy success in identifying a single language, identifying languages of words in code-mixed texts is a herculean task, more so in a social media context. This paper explores the English-Bengali code-mixing phenomenon and presents algorithms capable of identifying the language of every word to a reasonable accuracy in specific cases and the general case. We create and test a predictor-corrector model, develop a new code-mixed corpus from Facebook chat (made available for future research) and test and compare the efficiency of various machine learning algorithms (J48, IBk, Random Forest). The paper also seeks to remove the ambiguities in the token identification process.

1 Introduction

Code-mixing is a phenomenon in linguistics which is exhibited by multi-lingual people. Essentially, an utterance in which the speaker makes use of the grammar and lexicon of more than one language is said to have undergone code-mixing or code-switching (Appel and Muysken, 2005). Though some linguists draw a distinction between the terms "code-mixing" and "code-switching", we shall refer to both phenomena as "code-mixing" in general and draw distinctions regarding the context of switching when required (Muysken, 2000).

With English as the primary language on the internet, one would intuitively expect English to be the major language of use in social media as well. However, it comes as a bit of a surprise that around half of the messages on Twitter are in non-English languages (Schroeder, 2010). For multilingual people, we notice a tendency to communicate in all/several of the languages that they know. This arises from the fact that some multilingual speakers feel a higher level of comfort in their native language than in English. Apart from this, some conversational topics are more fluid in a particular language and some expressions convey the message properly only in one's native language.

In social media, code-mixing between languages such as English and Spanish (both of which employ a Roman script) is much simpler to analyze (apart from potential spelling mistakes) since the words used in normal Spanish and used in social media are spelled and used in almost entirely the same way and using the same script. However, when we consider languages that employ different scripts, most people do not have patience to switch between scripts while writing. Thus, people convert words from their language into the Roman script when mixing with English. In our analysis, we consider one such language, which is extremely fertile when it comes to code-mixing with English: Bengali. We wish to explore some procedures to identify languages in social media and internet search contexts in code-mixed English-Bengali texts.

The rest of the sections are as follows. Section 2 introduces the background of English-Bengali code-mixed data, especially in social media. Section 3 discusses the related work in the field of exploration. Section 4 speaks of the difficulties and hurdles in

Proceedings of the Second Workshop on Computational Approaches to Code Switching, pages 80–89,
Austin, TX, November 1, 2016.

this language identification task. In Section 5, we talk about the nature of code-switching instances that we have found in our corpora. Section 6 lists the techniques and tools we use in our experiments and Section 7 shares the details, results and observations from those experiments. In Section 8, we have a general discussion of our observations and we close in Section 9 with conclusions and future goals.

2 Background of English-Bengali Code-Mixing in Social Media

India is a linguistic area with one of the longest histories of contact, influence, use, teaching and learning of English-in-diaspora in the world (Kachru and Nelson, 2006). English is the *de facto lingua franca* in India and also an official language of the country (Guha, 2011). Thus, a huge number of Indians active on the internet are able in English communication to some degree. India also enjoys huge diversity in language. Apart from Hindi, it has several regional languages that are the primary tongue of people native to the region. Thus, a very vibrant trilingualism exists in large parts of the nation- This is seen more strongly in regions where the regional language is very widely spoken, like Bengali, which had an estimated 207 million speakers worldwide in 2007 (Microsoft Encarta, 2007). As a result, these languages are very likely to have a strong influence not only on daily speech, but also on internet and social media activities.

A significant feature of Indian multilingualism is that it is complementary. People speak in different languages in different scenarios. For example, an upper-middle class Bengali student would speak in Bengali at home, communicate in public in Hindi and in college in English. This is an integral part of the Indian composite culture (Sharma, 2001).

Due to this usage of multiple languages, code-mixing becomes inevitable and naturally spills over to social media and internet as well. In case of Bengali, since the language is very strongly tied to the daily lives of people, people have a very strong tendency to use it in written text as well. Bengali not only has a different script, but a vastly different alphabet as well. This heralds several new problems, which we discuss in the next section.

3 Related Work

Language identification and patterns in code-mixed data by itself is a field that has been explored for a very long time. Early linguistic research includes the work done in Spanish-English verbal communication by John Lipski, where he identifies the levels of linguistic competence in bilingual people as one of the root causes of code-switching (Lipski, 1978). Aravind Joshi presents a similar argument as well, in the case of Marathi-English code-switching (Joshi, 1982).

While that certainly is a big factor, there are other factors as well that we have spoken of earlier. The varying levels of diversity at the national, regional and local levels also influence different varieties of code-switching and between different languages as well (Singh and Gorla, 2007).

The Bengali-English code-switching phenomenon has historically been explored very little. A few groups have recently begun exploring the possibilities of language identification in these code-mixed situations. However, there are a few fundamental differences between their works and our work.

In other research works, some ambiguity is left with regard to the words that are present in both English and Bengali either by removing them (Das and Gambäck, 2013) or by classifying them as mixed (Depending on suffixes or word-level mixing) (Barman et al., 2014). However, such ambiguity needs to be removed, if we are required to utilize such type of data for further analysis or use them for building models of sentiment and/or predictive analysis, since people generally use mixed or ambiguous words in some single language context as well, which is why they code-mix in the first place.

In both of the other research works mentioned, the groups composed their own corpus from a Facebook group and the posts and comments by members (Das and Gambäck, 2013; Barman et al., 2014). Both of the groups also use N-gram pruning and dictionary checks.

Das and Gambäck (2013) also utilize SVM, while Barman et al (2014) use a word-length feature along with capitalization checks. We discuss relative accuracy in section 8.

4 Difficulties and Challenges in Language Identification

We shall introduce some of the problems and hurdles we faced during the system development and discuss them here.

4.1 Scarcity of uniform transliteration rules

During conversion of scripts, different people use different rules to decide the spellings of words. Suppose for instance, the Bengali word for "less", which contains the letters: "k"+"short a"+"m"+"short a". The short "a" vowel is a silent vowel in Bengali and indicates the lingering consonant sound at the end. In this case, this word would be transcribed in the IPA (International Phonetic Alphabet) as: IPA:[kɔm].

Due to letterwise transliteration, this word would be spelled in a dictionary conversion as "kama" owing to it's construction in Bengali language and the presence of silent/transformed "a" vowels.

However, interestingly, people tend to think of pronunciation rather than construction when converting the script, and the most socially acceptable spelling of this word is "kom". Such instances dominate a large portion of Bengali writing in social media.

Apart from this, there exist several occurrences of spelling variations. The word for "good" in Bengali is pronounced IPA:[b^{h}alo]. Since the [v] sound doesn't exist in Bengali, some assign the [b^h] sound to the letter "V" and spell good as "valo" rather than the standard spelling, "bhalo". This type of spelling variation exists in innumerable other words, but all versions are easily recognized by readers.

4.2 English slang terms

There is also the problem of modern English chat terms and colloquial terms like "lol" for "laughing out loud" and "gr8" for "great".

4.3 Bengali word forms

Bengali morphology relies heavily on "sandhi" and root transformation. For example, the word for "song" in Bengali is "*gaan*" and "singing" is "*gaowa*". However, "sang" would be "*geyechhilam*"/ "*geyechhilo*"/ "*geyechhilen*"/ "*geyechhile*" based on the subject and the honorific used. But, none of these match with the basic words "gaan" or "gaowa". This is very common in morphology and causes serious problems, since these are not noted in a dictionary. This would make us lose a huge number of words that come in sentences, because several words in such sentences are not in raw form, but transformed.

4.4 Ambiguous words

Generally, in many other studies, we have many words which could be both English and Bengali owing to their spelling. For instance, "hole" exists in English (IPA:[hoʊl]) and in Bengali (IPA:[hole] meaning "if {something} happens"). Individually, there is no way of knowing whether the word written in the English "hole" or the Bengali "hole", since they are spelled similarly. However, since we deal with sentences, we take into consideration the language that the surrounding words are in to get a sense of the context of the sentence. The English "hole" is likelier to be surrounded by more English words while the Bengali "hole" is likelier to be surrounded by more Bengali words. Thus, we have this context information that helps us.

4.5 Lack of Tagged Datasets

Being a less explored language, there is a dearth of English-Bengali code-mixed tagged datasets. For this reason, we use one corpus from the FIRE 2013 conference (hosted by the Information Retrieval Society of India), that consists of phrases such as search terms, and one corpus that we have built and tagged from social media messages exchanged on Facebook by college students and adults from West Bengal, India.

4.6 Imperfect English dictionary composition in WordNet

In testing for English words, we initially used WordNet[1]online (Princeton University, 2010). However, we faced some major obstacles with WordNet:

1. A lot of elementary English parts of speech including pronouns, such as "him", "her"; articles like "the"; conjunctions like "and", "because", "for"; prepositions such as "into", "onto", "to" are missing.

[1] http://wordnet.princeton.edu

Characteristics	FIRE 2013	Facebook chat
Text units	Phrases	Sentences
Expected source	Search terms	Chat messages
No. of lines	100	81
No. of tokens	539	518

Table 1: Details and statistics on test corpora used.

2. On the other hand, the WordNet repository contains a lot of foreign words and proper nouns, which aren't English words. Since we analyze and separate English from a foreign language, this causes problems, as a word like "Bharat" (native name for India) or "kobe" (Bengali word for "when") should ideally be classified as Bengali, but WordNet makes the system biased towards English.

4.7 Inappropriateness of Bengali dictionary

For checking the Bengali words, we use the Samsad Bengali-English dictionary's digital version (Biswas, 2004). This contains the words in Bengali text as well as a character-wise transliterated versions of the words. However, these English versions once again present to us the problem of "kama" vs "kom" as discussed in Section 4.1.

5 Descriptions of Datasets

The two test corpora we used:

1. **FIRE 2013 corpus**: The FIRE (Forum for Information Retrieval Evaluation) 2013 shared task had an annotated dataset corpus for Development, which we have used as one of our test corpora. It consists of short code-mixed phrases that resemble search queries on the internet (Information Retrieval Society of India, 2013).

2. **Facebook chat corpus**: We have composed our own Facebook chat corpus from the Facebook chats of various people of multiple age ranges, genders, geographic location and topic of discussion.

We also provide the statistics for our test corpora in Table 1.

The Facebook corpus is composed from chat messages, because we felt that a chat was more likely to have code-mixing, since one converses in a more informal setting there, while public posts are less likely to be influenced by code-mixing.

In the composition of our Facebook chat corpus (which has been made publicly available for future research[2]), we wanted to get a variety of styles of texting and mixing. For that reason, we collected text message conversations between Bengali college students, who are acquaintances, college students who were childhood friends, school friends and middle-aged women who are family friends. The annotation was done by an author.

The two corpora vary in their content types. As we see in Table 2, the FIRE 2013 corpus has the Bengali words heavily outweigh the number of English words, whereas in the Facebook chat corpus, we see an extremely level mix of words from both languages.

The "ambiguous words" row notes the number of words (already considered in the Bengali and English counts) that could belong to the other language too (based on our dictionary test). For example, the Bengali word, "more" IPA:[more] in the FIRE 2013 corpus could be the English "more" IPA:[mor] as well. The ambiguous words make up 29.68% of the FIRE 2013 corpus and 41.31% of the Facebook chat corpus. We divide these into two categories:

1. Bengali words that exist in the English dictionary: **Bengali (can be En)** in the table. "more" discussed above is an example of this category.

2. English words that find a match our search in the Bengali dictionary: **English (can be Bn)** in the table. For example, "to" IPA:[tu] in our Facebook chat corpus is an English word. However, a Bengali word, "to" IPA:[θo] also exists.

We notice there are a lot more **Bengali (can be En)** words than **English (can be Bn)** words in the FIRE 2013 corpus, while it is the exact opposite in the Facebook corpus. In fact, an interesting statistic is that out of all the English words in the Facebook chat corpus, 66.93% of the words register a match in the Bengali dictionary, and thus have a much higher likelihood of being wrongly classified. We can attribute this in part to the "Minimum Edit

[2] https://github.com/ArunavhaChanda/Facebook-Code-Mixed-Corpus

Characteristics	FIRE 2013	Facebook chat
Total words	**539**	**518**
Bengali words	364 (67.53%)	261 (50.39%)
English words	175 (32.47%)	257 (49.61%)
Ambiguous words	**160**	**214**
Bengali (can be En)	115 (71.88%)	42 (19.63%)
English (can be Bn)	45 (28.13%)	172 (80.37%)

Table 2: Details and statistics on test corpora used.

Distance" check in our Bengali search described in section 6.1.1.

6 Techniques and approaches

We have performed several versions of experiments and used different techniques. Our experiments can be divided into two major halves:

- **The first half** consisted of creating our own algorithms and processes that predicted and guessed the language of a word independently, without any machine learning used.

- **The second half** of our experiments were picking and choosing out features for every word and then using various machine learning algorithms on them to classify each word.

6.1 Resources used

6.1.1 Lexicons used

The dictionaries we use are the following:

1. **English dictionary**: We use the Python Enchant library[3] for checking English words and their existence in the dictionary. The good thing about Enchant is that the words included are not only in lemma form, but all kinds of morphological transformations are also included, making checks a lot easier. We also create a slang dictionary of our own containing colloquial English text words such as "LOL" and "gr8". We draw from the works of researchers at the University of Melbourne and University of Texas at Dallas (Han et al., 2012; Liu et al., 2011; Liu et al., 2012). We use this in both halves.

2. **Bengali dictionary**: The Samsada Bengali-English dictionary's digital edition[4] was used for the English transliterations of Bengali words and for a dictionary lookup (Biswas, 2004). What we do here to deal with the transliteration problem, is:

   - If a word in the dictionary ends in "a", we remove the "a" first (to account for the rarely used "short a" in social typing).
   - We then check for a Minimum Edit Distance of 1 character to the test word to indicate a match.

   For instance, when we check for "kom" typed in by someone, and our system is checking against "kama" (which is supposed to be the matching word), we have the "a" stripped from the end and obtain "kam" first. Then, we check for the Minimum Edit Distance between "kom" and "kam" and get 1. Thus, it is declared a match. This also accounts for some of the varied spellings discussed earlier. The Minimum Edit Distance we use incorporates the Levenshtein Distance (Levenshtein, 1966). We use this in both halves.

3. **Bengali suffix list**: We composed a list of common Bengali suffixes to check for transformed words. As discussed earlier, this helps us to deal with Bengali morphology, which is rather different from English morphology. We use this in both halves.

6.1.2 Training corpora used

These were two training corpora we used:

1. **Brown corpus**: We use the Brown corpus provided in the nltk[5] library in Python for a list of English words for creating an English language n-gram profile (Francis and Kučera, 1979). We use this in both halves.

2. **SHRUTI corpus**: We use the SHRUTI Bengali Continuous ASR Speech Corpus[6] to obtain

[3] https://pypi.python.org/pypi/pyenchant/

[4] http://dsal.uchicago.edu/dictionaries/biswas-bengali/

[5] http://www.nltk.org

[6] http://cse.iitkgp.ac.in/~pabitra/shruti_corpus.html

Bengali words. We use the same technique of removing "a"s from the ends of words, removing all punctuation, and lowering the case of all the words. Then, we create the n-gram profile for Bengali n-gram profile (Indian Institute of Technology, Kharagpur, 2011). We use this in both halves.

6.2 Algorithms and frameworks used

6.2.1 N-gram text categorization

We have developed an algorithm to using N-gram categorization for individual words. We use the two training corpora and extract every individual word and build language 4-gram profiles with them (since 4-grams give the best results) for Bengali and English. The profile consists of a sorted dictionary of the top 400 most frequently occurring 4-grams. When testing for a word, we find all the 4-grams for that word and then compute the distance of each 4-gram to it's counterpart in both the language profiles. If not found, we just assign it the length of the sorted language profile. We do this for each 4-gram in the word and then find the distances for both languages. The language that it is closer to is assigned as it's n-gram match. We use this in both halves.

6.2.2 WEKA

We use the WEKA (Waikato Environment for Knowledge Analysis) 3 tool for running our Machine Learning Algorithms and finding the success of our future algorithms (Hall et al., 2009). We use this in the second half.

6.2.3 Predictor-corrector algorithm

We will explain this algorithm with an example sentence from the FIRE 2013 corpus, which is entirely in Bengali:

Tormuj	*noon*	*die*	*khete*	*kemon*	*lage*
Watermelon	salt	with	to eat	how	taste

Meaning, "How does watermelon taste with salt?"

- Prediction: Our algorithm goes through the sentence identifying the language of each word by checking if it is English first and if not, if it is Bengali. We also perform this the other way around (Bengali, then English) depending on the order we determined. If it is neither, we find the n-gram match. This way, every word in the sentence gets tagged. In our example sentence, though all words are in Bengali, the words "noon" and "die" are tagged as English, since they are spelled the same way as words in English.

- Correction: In the sentence, the language that has more (predicted) words is declared the default language. After this, for each tagged word, we check if it existed in the non-tagged language's dictionary. If it does, we check the neighborhood (between 2-4 words depending on position in the sentence) of the word, and if the majority/half (depending on the default language) of the words are of the other language, it is corrected to the other language. In our sentence, the default language is Bengali and the words are checked one-by-one. "noon" is a word that can exist in Bengali and is surrounded by 67% Bengali words (2/3). So, it is corrected to Bengali. "die" also can exist in Bengali and is surrounded by 100% Bengali words (4/4 after correcting "noon") and is also corrected to Bengali.

  This way, our predictor-corrector method helps achieve better accuracy for ambiguous words. We use this in the first half.

7 Experiments and results

7.1 First half experiments

The following tests were performed in the first half:

1. Regular dictionary search (predictor), in two variants: (i) Check if word is Bengali first, then English. (ii)The reverse. Check for English first and then Bengali.

2. Predictor-corrector method (i) Check Bengali first. (ii) Check English first.

In the first part, whose results are mentioned in table 3, we note that the order in which we checked the language of words played a huge role in our accuracy (We measure accuracy as words correctly classified out of all the words). However, the interesting thing is that this gets reversed for both the corpora. For the Facebook chat corpus, we get a higher accuracy checking English first, while for the FIRE

Tools/Algorithm used	FIRE 2013	Facebook
Predictor (EB)	73.28%	88.22%
Predictor-corrector (EB)	82.56%	88.61%
Predictor (BE)	86.27%	64.86%
Predictor-corrector (BE)	86.09%	75.48%

Table 3: Results of first part. (BE: Bengali, then English; EB: English, then Bengali.)

Process/step	Accuracy
Regular dictionary	72.54%
+ Bengali suffixes	75.51%
+ "a" removals	77.37%
+ n-gram	86.27%

Table 4: Progressive results with features

corpus, we get a higher accuracy checking Bengali first. This knowledge fits in with the statistics we found previously.

From table 2, of the ambiguous words, we noted 71.88% of those in the FIRE corpus were Bengali and 80.37% of those in the Facebook chat corpus were in English. This shows that there is a greater chance of the Bengali words in the FIRE corpus being wrongly classified if we check for them being English first and the reverse for the Facebook chat corpus. However, if we check for words being Bengali first, the FIRE corpus' ambiguous Bengali words have a much higher chance of being correctly classified. This explains the varying effect order of checking has on corpora depending on the composition. Thus, it is not really a viable method unless we know details of word composition.

Another phenomenon we noted was the effect the inclusion of each feature had on the accuracy (Table 4). Before using the n-gram feature in the predictor for the first half experiments, we observed the following:

- Initially, we had only used regular dictionaries (the Samsada dictionary and Python Enchant) on the FIRE 2013 corpus and achieved an accuracy of 72.54% only (Bengali checks first). This was our baseline accuracy.
- After including our Bengali suffix repository, our accuracy increases to 75.51%.
- We include the checks by removing "a" from the ends of words in the Bengali dictionary to increment the accuracy to 77.37%.
- Finally, the accuracy increased with the n-gram feature to 86.27%.

The corrector method is also more useful with the less accurate predictor. This is likely because several of the ambiguous words, which were wrongly classified in the predictor, get accurately classified by the corrector and there are a lot more such words when we check words with the less ambiguous language first (English for FIRE and Bengali for Facebook; since words from the more ambiguous one get wrongly classified more). Thus, the corrector does a good job in such cases, giving accuracy boosts of +9.28% for the FIRE corpus and +10.62% for the Facebook corpus. However, it only marginally improves or reduces the accuracy in the reverse cases (-0.18% for FIRE and +0.39% for Facebook).

7.2 Second half experiments

The second half experiments were performed using WEKA and its machine learning algorithms. We used the following features of words as vectors:

1. Presence in Bengali dictionary (**B** in the table)
2. Presence in English dictionary (**E** in the table)
3. N-gram profile language match (**N** in the table)
4. Percentage of surrounding words that are "predicted" as Bengali: We predict using presence in one dictionary. If the word is in both or neither, we use the n-gram match as the predicted language (**S** in the table). We also use a version of this using majority language in the neighborhood, rather than percentage ((**s**) in the table)

Once we had these, we created arff (Attribute-Relation File Format) files of the words with these features as vectors and then used various algorithms in WEKA to classify the words.

The key results are summarized in table 5

We have listed results using three different classifiers and then with no surrounding data (only dictionaries and n-gram), using binary surrounding data, and using only the N-gram language categorizer.

In terms of classifiers, IBk performs the best, giving us accuracies of 91.65% and 90.54% as seen in the table.

	FIRE 2013				Facebook chat			
Classifier+Features	*Precision*	*Recall*	***F-score***	Accuracy	*Precision*	*Recall*	***F-score***	Accuracy
J48; **BENS**	0.895	0.896	0.895	89.61%	0.902	0.902	0.902	90.15%
IBk; **BENS**	**0.918**	**0.917**	**0.915**	**91.65%**	**0.905**	**0.905**	**0.905**	**90.54%**
RF; **BENS**	0.909	0.909	0.907	90.91%	0.893	0.892	0.892	89.19%
J48; **BEN**	0.900	0.887	0.880	88.68%	0.897	0.892	0.892	89.19%
J48; **BEN(s)**	0.909	0.905	0.902	90.54%	0.886	0.884	0.884	88.42%
J48; **N**	0.830	0.790	0.797	79.04%	0.812	0.805	0.804	80.50%

Table 5: Results of second part (Machine Learning). Key: RF=Random Forest

From the sixth row, it is evident that the N-gram categorization is very helpful, but not helpful enough. The fourth row shows us the result from using the dictionaries and breaking ties (word present in both or neither) using the N-gram decision is rather successful, though adding the surrounding data definitely helps the accuracy.

The results of the majority language in the neighborhood as a feature are in the fifth row. Interestingly, this technique caused a drop in accuracy for the Facebook chat corpus, but increased the accuracy for the FIRE 2013 corpus.

We did not use only the dictionaries as sole features, because the classification would have been 2-dimensional and much less useful than the dictionaries with N-gram.

8 Discussion

Comparing to similar tasks (on different corpora) by other groups, our system enjoys a fair amount of success. Compared to Das and Gambäck, our system has enjoyed greater success. They had a best accuracy of 76.37% compared to our 91.65% (Das and Gambäck, 2014). Their system employed SVM implementation, while ours did not. On the other hand, Barman et al achieved 95.98%, which was higher than ours (Barman et al., 2014) . The features used by them have been discussed earlier in section 3. Both these groups used their own corpora composed from similar Facebook groups used by college students. Our corpus has more variety in terms of age group and topics discussed. However, their corpora were larger in size.

One of the major points of discussion that we obtain from our research and experiments is the variation of dominant language in text. In past research, English (the language of the script) is considered the dominant/default language and the mixed language (Bengali, in our case) is considered secondary. However, this is not always the case. The Roman script is not chosen because English is the primary language. It is rather chosen out of convenience.

Hence, we assess the text first at the word level and then at the sentence level. In our corrector method, we assign the language with most words in a sentence as the default language for that sentence and then use our corrector method. It is likelier for a word surrounded by a lot of Bengali words in an English sentence to be Bengali, since code-switching happens mostly in a structured way. Language switch happens for a part of the sentence where the non-dominant language is more suitable. For example (with each 'E' and 'B' representing an English or Bengali word respectively), a code-mixed sentence is much likelier to be EEEEBBBBEEEE or BBBEEEBBBBEB, than EBEBBBEEBEBE. Considering this, we use the default language information to find the likelihood of a word needing to be corrected.

However, sometimes this correction in context causes a problem. It takes into account that sentences have certain points of code-switching and the switching is not random. This does give us an increase in accuracy for the Facebook chat corpus. However, the FIRE 2013 corpus consists of phrases and search terms and thus have random code-switching points rather than a particular pattern. For example:

She	*toh*	*elo*	*na*	**song**	**lyrics**
He/She	-	come	no		

(Bengali in italics; English in bold). This refers to the lyrics of a song called "*She toh elo na*". The way this sentence begins, one doesn't expect the last two words to be in English. However, since this is not a syntactical sentence, but a search phrase, we have "song lyrics" appear at the end. When our system

	Predictor		Corrector		IBk	
R\C	E	B	E	B	E	B
English	164	11	143	32	139	36
Bengali	132	232	62	302	9	355

Table 6: Confusion matrix, FIRE(EB) {R:Real, C:Classified}

checks for ”song”, it locates a Bengali word in the dictionary spelled as ”cong” IPA:[tʃoŋ]. Since the MED (Minimum Edit Distance) from it is 1, it registers ”song” as a potential Bengali word as well. And since it is heavily surrounded by Bengali words, it incorrectly re-tags ”song” as a Bengali word. However, the edit check is extremely necessary. A supporting example in our Facebook chat corpus itself exists in the word IPA: [aʃolɛ] (meaning ”actually” or ”really”). It is spelled in two different ways: ”asole” and ”ashole”. Such variations in spelling exist largely among different people and we need the MED test to verify.

To avoid such errors, in our second half, we do not correct as a post-processing step, but rather choose to include percentage of surrounding Bengali words as a feature in our vector.

Table 6 shows the progressive improvement (and errors) in classification. The number of Bengali words wrongly classified vastly decreases, but the number of English words being wrongly classified slightly increases: a trade-off that seems worth it.

We have also discussed the variations in spelling of Bengali words, and that remains a factor of concern that we have mitigated slightly, but not enough. For instance, people also tend to contract spellings even in Bengali. The word for ”good” (”bhalo” [b^halo]) is also spelled by some people as ”valo”, which is further contracted to ”vlo”. There are also acronyms, which can be Bengali or English, and we have no way of identifying them without a repository or a collection of common acronyms in both. Such circumstances are still difficult to deal with and need to be considered in future work.

Ambiguous words being assigned determined languages was one of the major decisions we took early on in our research, since even ambiguous words are written with a certain sentiment and language in mind.

There are problems of using our Bengali suffix list as well. For example, one of the most important suffixes we have here is ”te” IPA:[θe], used in ”shute” (to sleep), ”khete” (to eat), ”jete” (to go). One use of this is to make the verb an object verb. For example:

O	khete	gechhe
He/She	to eat	has gone

This sentence means ”He has gone to eat”. It is necessary to have the suffixes to identify ”khete” as a Bengali word and we need to check for the suffix ”te” to achieve that. However, for this reason, English words like ”cute”, ”mute” and ”forte” are also marked off as existing in the Bengali lexicon. The trade-off is a small one and considering the suffixes helps a lot in identifying Bengali words, it seems justified.

9 Conclusion and Future Work

If we are to carry out sentiment analysis in the future, it will be crucial to identify language and context for every word to understand the sentiment it conveys. Another tool required for sentiment analysis of English-Bengali code-mixed data is a Part-of-Speech tagger for Bengali. The way this will help even in code-mixed context is that once we know the default language of a sentence, it is likely to follow the grammatical/syntactical structure of that language. This helps put all the words in the sentence into context and makes it easier to (a) correct languages, if necessary and (b) helps understand the context of a word and by extension, its sentiment better.

Since Bengali is still a relatively unexplored language in Natural Language Processing, apart from a Part-of-Speech tagger, another hurdle is proper nouns: Names of people, places, landmarks etc are not found in any dictionary. Though the N-gram categorization routine helps a lot in this regard, it would still help to be able to identify these names.

In conclusion, we have given ways to improve language identification in code-mixed English-Bengali data to a large extent. If we are equipped with the POS tagger and the other points mentioned here, we will be at a stage to confidently work on the sentiment analysis of code-mixed data as well.

References

René Appel and Pieter Muysken, 2005. *Code Switching and Code Mixing*. Amsterdam University Press.

Utsab Barman, Amitava Das, Joachim Wagner, and Jennifer Foster. 2014. Code Mixing: A Challenge for Language Identification in the Language of Social Media. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*.

Sailendra Biswas. 2004. *Samsada Bangala abhidhana*. Sahitya Samsad, 7th edition.

Amitava Das and Björn Gambäck. 2013. Code-Mixing in Social Media Text. *Traitement Automatique des Langues*, 54.

Amitava Das and Björn Gambäck. 2014. Identifying Languages at the Word Level in Code-Mixed Indian Social Media Text. In *Proceedings of the 11th International Conference on Natural Language Processing*.

W.N. Francis and H Kučera. 1979. A Standard Corpus of Present-Day Edited American English, for use with Digital Computers (Brown). `http://www.helsinki.fi/varieng/CoRD/corpora/BROWN/`.

Ramachandra Guha. 2011. *India After Gandhi: The History of the World's Largest Democracy*. Pan Macmillan.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11.

Bo Han, Paul Cook, and Timothy Baldwin. 2012. Automatically Constructing a Normalisation Dictionary for Microblogs. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 421–432, Jeju Island, Korea, July. Association for Computational Linguistics.

Indian Institute of Technology, Kharagpur. 2011. SHRUTI Bengali Continuous ASR Speech Corpus. `http://cse.iitkgp.ac.in/~pabitra/shruti_corpus.html`.

Information Retrieval Society of India. 2013. Datasets for FIRE 2013 Track on Transliterated Search. `http://www.isical.ac.in/~fire/2013/`.

Aravind K. Joshi. 1982. Processing of sentences with intrasentential code-switching. *COLING-82*, pages 145–150.

Yamuna Kachru and Cecil L. Nelson. 2006. *World Englishes in Asian Contexts*. Hong Kong University Press.

V.I. Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707–710.

John M. Lipski. 1978. Code-switching and the problem of bilingual competence. *Aspects of bilingualism*, pages 250–264.

Fei Liu, Fuliang Weng, Bingqing Wang, and Yang Liu. 2011. Insertion, Deletion, or Substitution? Normalizing Text Messages without Pre-categorization nor Supervision. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*, pages 71–76.

Fei Liu, Fuliang Weng, and Xiao Jiang. 2012. A Broad-Coverage Normalization System for Social Media Language. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*, pages 1035–1044.

Microsoft Encarta. 2007. Languages Spoken by More Than 10 Million People. `http://web.archive.org/web/20071203134724/http://encarta.msn.com/media_701500404/Languages_Spoken_by_More_Than_10_Million_People.html`.

Pieter Muysken. 2000. *Bilingual Speech: A Typology of Code-Mixing*. Cambridge University Press.

Princeton University. 2010. About WordNet. `http://wordnet.princeton.edu`.

Stan Schroeder. 2010. Half of Messages on Twitter Aren't in English [STATS]. `http://mashable.com/2010/02/24/half-messages-twitter-english/`.

J.C. Sharma. 2001. Multilingualism in India. *Language in India*, 1.

Anil Kumar Singh and Jagadeesh Gorla. 2007. Identification of languages and encodings in a multilingual document. In *Proceedings of ACL-SIGWAC's Web As Corpus3*.