Identifying Languages at the Word Level in Code-Mixed Indian Social Media Text

Amitava Das
University of North Texas
Denton, Texas, USA
amitava.santu@gmail.com

Björn Gambäck
Norwegian University of Science and Technology
Trondheim, Norway
gamback@idi.ntnu.no

Abstract

Language identification at the document level has been considered an almost solved problem in some application areas, but language detectors fail in the social media context due to phenomena such as utterance internal code-switching, lexical borrowings, and phonetic typing; all implying that language identification in social media has to be carried out at the word level. The paper reports a study to detect language boundaries at the word level in chat message corpora in mixed English-Bengali and English-Hindi. We introduce a code-mixing index to evaluate the level of blending in the corpora and describe the performance of a system developed to separate multiple languages.

1 Introduction

An essential prerequisite for any kind of automatic text processing is to be able to identify the language in which a specific segment is written. Here we will in particular address the problem of word level language identification in social media texts. Available language detectors fail for these texts due to the style of writing and the brevity of the texts, despite a common belief that language identification is an almost solved problem (McNamee, 2005). Previous work has concentrated on identifying the (single) overall language of full documents or the proportions of different languages appearing in mixed-language documents. Longer documents tend to have fewer code-switching points, caused by loan words or author shifts. The code-mixing addressed here is more difficult and novel: we only have access to short (Facebook chat) utterances.

Looking at code-mixing in social media text is also overall a new research strand. These texts are characterised by having a high percentage of spelling errors and containing creative spellings (*gr8* for *'great'*), phonetic typing, word play (*goooood* for *'good'*), and abbreviations (*OMG* for *'Oh my God!'*). Non-English speakers do not always use Unicode to write social media text in their own language, frequently insert English elements (through code-mixing and Anglicisms), and often mix multiple languages to express their thoughts, making automatic language detection in social media texts a very challenging task, which only recently has started to attract attention.

Different types of language mixing phenomena have, however, been discussed and defined by several linguists, with some making clear distinctions between phenomena based on certain criteria, while others use 'code-mixing' or 'code-switching' as umbrella terms to include any type of language mixing — see, e.g., Muysken (2000) or Gafaranga and Torras (2002) — as it is not always very clear where word loaning stop and code-mixing begins (Alex, 2008). In the present paper, we will take 'code-mixing' as referring to the cases where the language changes occur inside a sentence (which also sometimes is called intra-sentential code-switching), while we will refer to 'code-switching' as the more general term and in particular use it for inter-sentential phenomena.

The next section discusses the concept of code-switching and some previous studies on code-mixing in social media text. Section 3 then introduces the data sets that have been used for investigating code-mixing between English and Hindi as well as between English and Bengali. Section 4 describes the methods used for word level language detection, based on character n-grams, dictionaries, and support vector machines, respectively. The language detection experiments are reported in Section 5, while Section 6 discusses the results. Finally, Section 7 sums up the discussion and points to some areas of future research.

2 Background and Related Work

In the 1940s and 1950s, code-switching was often considered a sub-standard use of language. However, since the 1980s it has generally been recognised as a natural part of bi- and multilingual language use. Linguistic efforts in the field have mainly concentrated on the sociological and conversational necessity behind code-switching (Auer, 1984) and its nature (Muysken, 1995), for example, on whether it is an act of identity in a group or competence-related (i.e., a consequence of a lack of competence in one of the languages), and on dividing switching into sub-categories such as *inter-* vs *intra-sentential* (if it occurs outside or inside sentence / clause boundaries); *intra-word* vs *tag* (if switching occurs inside a word, e.g., at a morpheme boundary, or by inserting a tag phrase or word from one language into another).

Following are examples of each type of code-switching between English and Bengali. Bengali segments are in boldface and each example has its corresponding English gloss on a new line.

Inter-sentential switching

Fear cuts deeper than sword ....
Fear cuts deeper than a sword ....

bukta fete jachche ... :(
it seems my heart will blow up ... :(

Intra-sentential switching

dakho sune 2mar kharap lagte pare
You might feel bad hearing this

but it is true that u r confused.
but it is true that you are confused.

Tag switching

ami majhe majhe *fb* ***te on hole ei***
While I get on facebook I do visit

confession page ***tite aasi.***
(the) confession page very often.

Intra-word switching

tomar osonkkhho *admirer* ***der***
Among your numerous admirer-s,

modhhe ami ekjon nogonno manush
I'm the inconsiderable one

In the intra-word case, the plural suffix of *admirer* has been Bengalified to *der*.

2.1 Characteristics of Code-Mixing

Several researchers have investigated the reasons for why code-mixing appear. Studies on Chinese-English code-mixing in Hong Kong (Li, 2000) and Macao (San, 2009) indicated that mainly linguistic motivations were triggering switches in those highly bilingual societies, with social motivations being less salient. However, this contrasts with studies on other language pairs and user groups in various social media, indicating that code-mixing often takes place at the beginning of messages or through simple insertions, and mainly to mark in-group membership: in short text messages (Sotillo, 2012), chat messages (Bock, 2013), Facebook comments (Shafie and Nayan, 2013), and emails (Negrón Goldbarg, 2009)

Other studies have investigated the types and frequency of code-mixing in social media. Hidayat (2012) showed that facebookers tend to mainly use inter-sentential switching (59%) over intra-sentential (33%) and tag switching (8%), and reports that 45% of the switching was instigated by real lexical needs, 40% was used for talking about a particular topic, and 5% for content clarification. The predominance of inter-sentential code-switching in social media text was also noted by San (2009), who compared the switching in blog posts to that in the spoken language in Macao.

2.2 Automatic Analysis of Code-Switching

The problem of language identification has been investigated for half a century (Gold, 1967) and that of computational analysis of code switching for several decades (Joshi, 1982), but there is little previous work on automatic language identification for multilingual code-mixed texts, although there have been some related studies on identifying code-switching points in speech (Chan et al., 2009; Solorio et al., 2011; Weiner et al., 2012). Notably, this work has mainly been on artificially generated speech data, with the simplification of only having 1–2 code-switching points per utterance. The spoken Spanish-English corpus used by Solorio and Liu (2008) is a small exception, with 129 intra-sentential language switches. They looked at part-of-speech tagging for this type of data in part by utilising a language identifier as a pre-processing step, but with no significant improvement in tagging accuracy.

Previous work on text has mainly been on identifying the (one) language (from several possible) of documents or the proportion of a text written in a language, often restricted to 1–2 known languages; so even when evidence is collected at word level, evaluation is at document level (Prager, 1997; Singh and Gorla, 2007; Yamaguchi and Tanaka-Ishii, 2012; Rodrigues, 2012; King and Abney, 2013; Lui et al., 2014). Other studies have looked at code-mixing in different types of short texts, such as information retrieval queries (Gottron and Lipka, 2010) and SMS messages (Rosner and Farrugia, 2007), or aimed to utilize code-mixed corpora to learn topic models (Peng et al., 2014) or user profiles (Khapra et al., 2013).

Most closely related to the present work are the efforts by Carter (2012), Lignos and Marcus (2013), Nguyen and Doğruöz (2013), and Voss et al. (2014). Carter collected Twitter messages (tweets) in five different European languages and manually inspected the multilingual micro-blogs for determining which language was the dominant one in a specific tweet. He then performed post level language identification, experimenting with a range of different models and a character n-gram distance metric, reporting a best overall classification accuracy of 92.4%. Evaluation at post level is reasonable for tweets, as Lui and Baldwin (2014) note that users that mix languages in their writing still tend to avoid code-switching within a tweet. However, for this is not the case for the chat messages that we address in the present paper.

Nguyen and Doğruöz (2013) investigated language identification at the utterance level on randomly sampled mixed Turkish-Dutch posts from an online chat forum, mainly annotated by a single annotator, but with 100 random posts annotated by a second annotator. They compared dictionary-based methods to language models, and adding logistic regression and linear-chain Conditional Random Fields. Their best system reached a high word accuracy (97.6%), but with a substantially lower accuracy on post level (89.5%), even though 83% of the posts actually were monolingual. Similarily, Lignos and Marcus (2013) also only addressed the bi-lingual case, looking at Spanish-English tweets. The strategy chosen by Lignos & Marcus is interesting in its simplicity: they only use the ratio of the word probability as information source and still obtain good results, the best being 96.9% accuracy at the word-level. However, their corpora are almost monolingual, so that result was obtained with a baseline as high as 92.3%. Voss et al. (2014) on the other hand worked on quite code-mixed tweets (20.2% of their test and development sets consisted of tweets in more than one language). They aimed to separate Romanized Moroccan Arabic (Darija), English and French tweets using a Maximum Entropy classifier, achieving F-scores of .928 and .892 for English and French, but only .846 for Darija due to low precision.

Number	EN-BN1	EN-HN1
Utterances	2,309	1,743
Words	71,207	66,494
Unique tokens	15,184	10,314

Table 1: Details of corpus collection

3 Code-Mixed Corpora

Most research on social media texts has so far concentrated on English, whereas the majority of these texts now are in non-English languages (Schroeder, 2010). Fischer (2011) provides an interesting insight on Twitter language usages in different geographical regions. Europe and South-East Asia are the most language-diverse areas of the ones currently exhibiting high Twitter usage. It is likely that code-mixing is frequent in those regions, where languages change over short geo-spatial distances and people generally have basic knowledge of the neighbouring languages.

Here we will concentrate on India, a nation with close to 500 spoken languages (or over 1600, depending on what is counted as a language) and with some 30 languages having more than 1 million speakers. India has no national language, but 22 languages carry official status in at least parts of the country, while English and Hindi are used for nation-wide communication. Language diversity and dialect changes instigate frequent code-mixing in India. Hence, Indians are multi-lingual by adaptation and necessity, and frequently change and mix languages in social media contexts.

3.1 Data Acquisition

English-Hindi and English-Bengali language mixing were selected for the present study. These language combinations were chosen as Hindi and Bengali are the two largest languages in India in terms of first-language speakers (and 4th and 7th worldwide). In our study, we include corpora collected both by ourselves for this study and by Utsab Barman (Burman et al., 2014), hereforth called EN-BN1 and EN-HN1 resp. EN-BN2 and EN-HN2. Various campus Facebook groups

Tag	Description	Tag	Description
en	English word	en+bn_suffix	English word + Bengali suffix
		en+hi_suffix	English word + Hindi suffix
bn	Bengali word	bn+en_suffix	Bengali word + English suffix
hi	Hindi word	hi+en_suffix	Hindi word + English suffix
ne	Named Entity (NE)	acro	Acronym
ne+en_suffix	NE + English suffix	acro+en_suffix	Acronym + English suffix
ne+bn_suffix	NE + Bengali suffix	acro+bn_suffix	Acronym + Bengali suffix
ne+hi_suffix	NE + Hindi suffix	acro+hi_suffix	Acronym + Hindi suffix
univ	Universal	undef	Undefined / Others

Table 2: Word level code-mixing annotation tagset

Corpus	Code Switching Types			Total
	Intra	Inter	Word	
EN-BN2	60.23%	32.20%	7.58%	56.51%
EN-HN2	54.71%	37.33%	7.96%	28.51%

Table 3: Code-switching categorisation

were used for data acquisition. In both cases, the English-Bengali data came from Jadavpur University in Eastern India where the native language of most students is Bengali. For English-Hindi, the data came from the Indian Institute of Technology Bombay in the West of the country where Hindi is the most common language. Table 1 presents statistics for our corpora for both language pairs.

The languages of the corpora were tagged at the word-level with the tag-set displayed in Table 2. The univ tag stands for emoticons (:), :(, etc.) and characters such as ", ', >, !, and @, while undef is for tokens that are hard to categorise.

3.2 Types of Code-Switching

The distribution of code-switching types is reported in Table 3, under the hypothesis that the base language is English with the non-English words (i.e., Hindi/Bengali) having been mixed in. Named entities and acronyms were treated as language independent, but assigned the language for multilingual categories based on suffixes.

The figures for inter- and intra-sentential code-switching were calculated automatically and are based on the total code-switching found in the corpus: if the language of a sentence was fully tagged either as Bengali or Hindi, then that sentence was considered as a type of inter-sentential code-switching, and all words in that sentence contribute to the inter-sentential code-switching percentage. For the word-internal code-mixing identification, only the “⋆ + ⋆ suffix” tags were considered. Tag-mixing was not considered or annotated as it either is a semantic category or can be further described as a subcategory of intra-sentential code-switching.

The ‘total’ percentage in Table 3 has been calculated at the word level, that is, as

$$\frac{\textit{total number of words found in non-English}}{\textit{total number of words in the corpus}}$$

The distributions of different mixing types were then calculated based on the total code-mixing found for the particular language pair.

3.3 A Code-Mixing Index

A typical inter-sentential code-switching example:

Yaar tu to, GOD *hain.*
Dude you are GOD.
tui JU te ki korchis? Hail u man!
What you are doing in JU? Hail u man!

This comment was written in three languages: English, Hindi (italics), and Bengali (boldface italics; JU is an abbreviation for Jadavpur University, but we treated named entities as language independent). The excerpt stems from the “JU Confession” Facebook group, which in general is English-Bengali; however, it has some 3–4% Hindi words mixed in (since Hindi is India’s primary nation-wide language it has strong influence on the other languages of the country). The example shows how closely languages co-exist in social media text, making language detection for these types of text a very complex task.

When comparing different code-mixed corpora to each other, it is desirable to have a measurement of the level of mixing between languages. To this end we introduce a *code-mixing index*, CMI. At the utterance level, this amounts to finding the most frequent language in the utterance and then counting the frequency of the words belonging to all other languages present, that is,

$$\text{CMI} = \frac{\sum_{i=1}^{N}(w_i) - max\{w_i\}}{n-u} \qquad (1)$$

Tag	EN-BN1
CMI mixed	24.48
CMI all	5.15
Non-mixed (%)	78.95
Mixed (%)	21.05

Table 4: Level of code-mixing in a corpus

where $\sum_1^N(w_i)$ is the sum over all N languages present in the utterance of their respective number of words, $max\{w_i\}$ is the highest number of words present from any language (regardless of if more than one language has the same highest word count), n is the total number of tokens, and u is the number of tokens given language independent tags (in our case that means tokens tagged as "universal", as abbreviations, and as named entities),

If an utterance only contains language independent tokens, we define its index to be zero. For others, we multiply by 100 to get digits in the range 0 : 100. Further, since $\sum_1^N(w_i)$ in fact is equivalent to $n - u$, Equation 1 can be rewritten as

$$\text{CMI} = \begin{cases} 100 \times [1 - \frac{max\{w_i\}}{n-u}] & : n > u \\ 0 & : n = u \end{cases} \tag{2}$$

where w_i is the words tagged with each language tag ($w_{en}, w_{hi}, w_{bn}, ...$; hence excluding items tagged `univ`, `acro`, `ne`, `undef`, etc., while including those with language tags and language-based suffix tags) and $max\{w_i\}$ thus is the number of words of the most prominent language (so for mono-lingual utterances, we will get CMI $= 0$, since then $max\{w_i\} = n - u$).

As an example, we utilize this index to evaluate the level of code-mixing in our English-Bengali corpus both on average over all utterances and on average over the utterances having a non-zero CMI, that is, over the utterances that contain some code-mixing, as shown in Table 4. It is also important to give the fraction of such utterances in a corpus, so we include those numbers as well.

4 Word-Level Language Detection

The task of detecting the language of a text segment in mixed-lingual text remains beyond the capabilities of classical automatic language identification techniques, e.g., Cavnar and Trenkle (1994) or Damashek (1995). We have tested some of the state-of-the-art systems on our corpora and found that they in general fail to separate language-specific segments from code-switched texts.

Instead we experimented with a system based on well-studied techniques, namely character n-gram distance measures, dictionary-based information, and classification with support vector machines, as further described in this section. The actual experiments and results with this system are reported in Section 5, where we also discuss ways to improve the system by adding post-processing.

4.1 N-gram Language Profiling and Pruning

The probably most well-known language detection system is *TextCat* (Cavnar and Trenkle, 1994), which utilises character-based n-gram models. The method generates language specific *n-gram* profiles from the training corpus sorted by their frequency. A similar text profile is created from the text to be classified, and a cumulative "out-of-place" measure between the text profile and each language profile is calculated. The measure determines how far an n-gram in one profile is from its place in the other profile. Based on that distance value, a threshold is calculated automatically to decide the language of a given text. Since the work of Beesley (1988), this approach has been widely used and is well established in language identification (Dunning, 1994; Teahan, 2000; Ahmed, 2005). Andersen (2012) also investigated n-gram based models, both in isolation and combined with the dictionary-based detection described below, as well as with a rule-based method utilising manually constructed regular expressions.

An *n-gram* model was adopted for the present task, too, but with a pruning technique to exclude uninformative n-grams during profile building. Common (high-frequency) n-grams for language pairs are excluded, as they are ambiguous and less discriminative. So is, for example, the bigram "*to*" very common in all the three languages, so less discriminative. To achieve this, a weight w_i^a is calculated for each n-gram g_i in language l_a by the formula $w_i^a = f_i^a / m_a$ where f_i^a is the frequency of the n-gram g_i in language l_a and m_a the total number of n-grams in language l_a. A particular n-gram g_i is excluded if its discriminative power when comparing languages l_a and l_b is lower than an experimentally chosen threshold value θ, i.e., if the condition $|w_i^a - w_i^b| \leq \theta$ is true.

There are various trade-offs to consider when choosing between character n-grams and word n-grams, and when deciding on the values of n and θ, i.e., the size of the n-grams and the discrimination threshold. Using Romanization for the Hindi and Bengali, and converting all text to lower-case, the alphabet of English is limited to 26 characters, so the set of possible character n-grams remains manageable for small values of n. White-spaces between words were kept for n-gram creation, in order to distinctly mark word boundaries, but multiple white-spaces were removed.

We carried out experiments on the training data for $n = \{1, 2, 3, 4, 5, 6, 7\}$, and found 3-grams and 4-grams to be the optimum choices after performance testing through 10-fold cross validation, with $\theta = 0.2$. The value of θ was not varied: n-grams with the same presence in multiple languages are less discriminating. The presence ratio should be $> 2\%$, so that value was selected for θ. N-gram pruning helps reduce the time it takes the system to converge by a factor 5 and also marginally increases performance (by 0.5).

4.2 Dictionary-Based Detection

Use of most-frequent-word dictionaries is another established method in language identification (Alex, 2008; Řehůřek and Kolkus, 2009). We incorporated a dictionary-based language detection technique for the present task, but were faced with some challenges for the dictionary preparation, in particular since social media text is full of noise. A fully edited electronic dictionary may not have all such distorted word forms as are used in these texts (e.g., '*gr8*' rather than '*great*'). Therefore a lexical normalisation dictionary (Han et al., 2012) prepared for Twitter was used for English.

Unfortunately, no such dictionary is available for Hindi or Bengali, so we used the Samsad English-Bengali dictionary (Biśvās, 2000). The Bengali part of the Samsad dictionary is written in Unicode, but in our corpus the Bengali texts are written in transliterated/phonetic (Romanized) form. Therefore the Bengali lexicon was transliterated into Romanized text using the Modified-Joint-Source-Channel model (Das et al., 2010). The same approach was taken when creating the Hindi dictionary, using Hindi WordNet (Narayan et al., 2002). In order to capture all the distorted word-forms for Hindi and Bengali, an edit distance (Levenshtein, 1966) method was adopted. A Minimum Edit Distance (MED) of ±3 was used as a threshold (chosen experimentally).

The general trend in dictionary-based methods is to keep only high-frequency words, but that is for longer texts, and surely not for code-mixing situations. Our language detection solution is targeted at the word level and for short texts, so we cannot only rely on the most-frequent word lists and have thus instead used the full-length dictionaries. Again, words common in any of the language pairs were excluded. For example, the word "*gun*" (ENG: weapon, BNG: multiplication, HND: character/properties/competence/talent) was deleted from all three dictionaries as it is common and thus ambiguous. Another example is the word "*din*" which is common in English (loud) and Hindi (day) dictionaries, and therefore removed. The Hindi-Bengali dictionary pair was not analysed because there are huge numbers of lexical overlaps between these two languages. Words that cannot be found in any of these dictionaries are labelled as **undef** and passed for labelling to the SVM module (described next), which can consider language tags of the contextual words.

4.3 SVM-based Word-Language Detection

Word level language detection from code-mixed text can be defined as a classification problem. Support Vector Machines (SVM) were chosen for the experiment (Joachims, 1999). The reason for choosing SVM is that it currently is the best performing machine learning technique across multiple domains and for many tasks, including language identification (Baldwin and Lui, 2010). Another possibility would be to treat language detection as sequence labelling and train the word level language tag sequences using the best performing sequence labelling techniques, e.g., Hidden Markov Models (HMM) or Conditional Random Fields (CRF). For the present system, the SVM implementation in Weka v3.6.10 (Hall et al., 2009) was used with default parameters. This linear kernel SVM was trained on the following features:

N-gram with weights: Implemented using the bag-of-words principle. If there after pruning are n unique n-grams for a language pair; there are n unique features. Assume, for example, that "*in*" is the i^{th} bigram in the list. In a given word w (e.g., *painting*), a particular n-gram occurs k times (twice for "*in*" in *painting*). If the pre-calculated weight of "*in*" is t_w^i, the feature vector is $1, 2, ..., (t_w^i * k), .., (n-2), (n-1), n$. Weights are set to 0 for absent n-grams. Weighting gives 3–4% better performance than binary features.

Dictionary-based: One binary feature for each of the three dictionaries (ENG, BNG, HND).

MED-based weight: Triggered if a word is out-of-vocabulary (absent in all dictionaries). The Minimum Edit Distance is calculated for each language, choosing the lowest MED as feature value. To simplify the search, radix sort, binary search and hash map techniques were incorporated.

Word context information: A 7-word window feature (including ±3 words) was used to incorporate contextual information. Surface-word forms for the previous three words and their language tags along with the following three words were considered as binary features. For each word there is a unique word dictionary pre-compiled from all corpora for both language pairs, and only three features were added for language tags.

5 Experiments and Performance

A simple dictionary-based method was used as baseline, hypothesising that each text is bilingual with English as base language. An English dictionary was used to identify each word in the text and the undefined words were marked either as Hindi or Bengali based on the corpus. In a real-world setting, location information could be extracted from the social media and the second language could be assumed to be the local language. For both languages, the base performance is below 40% (38.0% and 35.5% F-score for ENG-HND and ENG-BNG, resp.), which gives a clear indication of the difficulty.

To understand the effect of each feature and module, experiments were carried out at various levels. The n-gram pruning and dictionary modules were evaluated separately, and those features were used in the SVM classification. The performance at the word level on the test set is reported in Table 5. In addition, we run 10-fold cross-validation on the training set using SVM on both the language pairs and calculated the performance. The results then were quite a lot higher (around 98% for ENG-HND and 96% for ENG-BNG), but as can be seen in the table, evaluation on the test set made performance drop significantly. Hence, though using 10-fold cross-validation, the SVM certainly overfits the training data, which could be addressed by regularization and further feature selection. The N-gram pruning was an attempt at feature selection, but adding other features or filtering techniques is definitely possible.

Looking at the system mistakes made on the development data, a post-processing module was designed for error correction. The most prominent errors were caused by language in continuation: Suppose that the language of the words w_n and w_{n+2} is marked by the system as l_a and that the language of the word w_{n+1} is marked as $\neg l_a$, then the post-processor's role is to restore this language to l_a. This is definitely not a linguistically correct assumption, but while working with word-level code-mixed text, this straightforward change gives a performance boost of approximately 2–5% for both language pairs, as can be seen in the last line of Table 5.

There are also a few errors on language boundary detection, but to post-fix those we would need to add language-specific orthographic knowledge.

6 Discussion

Social media text code-mixing in Eurasian languages is a new problem, and needs more efforts to be fully understood and solved. This linguistic phenomenon has many peculiar characteristics, for example: *addaing*, *jugading*, and *frustu* (meaning: being frustated). It is hard to define the language of these words, but they could be described as being examples of "*Engali*" resp. "*Engdi*", along the lines of Benglish and Hinglish, (see Section 3.1), i.e., the root forms are English, but with suffixes coming from Bengali and Hindi.

Another difficult situation is reduplication, which is very frequent in South-East Asian languages. The social media users are influenced by the languages in their own geo-spaces, so reduplication is quite common in South-East Asian code-mixed text. The users in these regions are also very generative in terms of reduplication and give birth to new reduplication situations that are not common (or even valid) in any of the local languages or in English; for example: *affair taffair*.

It is also difficult to compare the results reported here to those obtained in other media and for other types of data: While previous work on speech mainly has been on artificially generated data, previous work on text has mainly been on language identification at the document level, even when evidence is collected at word level. Longer documents tend to have fewer code-switching points.

The code-mixing addressed here is more difficult and novel, and the few closely related efforts cannot be directly compared to either: the multi-lingual Twitter-setting addressed by Voss et al. (2014) might be closest to our work, but their

System		Precision		Recall		F_1-Score	
		HND	BNG	HND	BNG	HND	BNG
N-Gram Pruning		70.12%	69.51%	48.32%	46.01%	57.21%	55.37%
	+ Dictionary	82.37%	77.69%	51.03%	52.21%	63.02%	62.45%
SVM	Word Context	72.01%	74.33%	50.80%	48.55%	59.57%	58.74%
	+ N-Gram Weight	89.36%	86.83%	58.01%	56.03%	70.35%	68.11%
	+ Dictionary + MED	90.84%	87.14%	65.37%	60.22%	76.03%	74.35%
Post Processing		94.37%	91.92%	68.04%	65.32%	79.07%	76.37%

Table 5: System word-level performance for language detection from code-mixed text

results were hurt by very low precision for Morrocan Arabic, possibly since they only used a Maximum Entropy classifier to identify languages. The solution used by Carter (2012) is based on Twitter-specific priors, while the approach by Nguyen and Doğruöz (2013) utilises language specific dictionaries (just as our approach does), making a comparison across languages somewhat unfair. The idea introduced by Lignos and Marcus (2013), to only using the ratio of the word probability, would potentially be easier to compare across languages.

Our work also substantially differs from Nguyen and Doğruöz (2013) and Lignos and Marcus (2013) in that we address a multilingual setting, while their work is strictly bilingual (with the first authors making the assumption that words from other languages — English — appearing in the messages could be assumed to belong to the dominating language, i.e., Dutch in their case). Further, even though they also work on chat data, Nguyen and Doğruöz (2013) mainly investigated utterance (post) level classification, and hence give no actual word-level baseline, just stating that 83% of the posts are monolingual. 2.71% of their unique tokens are multilingual, while in our case it is 8.25%. Nguyen & Dogruoz have gratefully made their data available and testing our system on it gives a slightly increased accuracy compared to their results (by 1.0%).

7 Conclusion and Future Work

The social media revolution has added a new dimension to language processing, with the borders of society fading, and the mixing of languages and cultures increasing. The paper has presented an initial study on the detection of code-mixing in the context of social media texts. This is a quite complex language identification task which has to be carried out at the word level, since each message and each single sentence can contain text and words in several languages.

The experiments described in here have focused on code-mixing only in Facebook posts written in the language pairs English-Hindi and English-Bengali. In the future, it would be reasonable to experiment with other languages and other types of social media text, such as tweets. Although Facebook posts tend to be short, they are commonly not as short as tweets, which have a strict length limitation (to 140 characters). It would be interesting to investigate whether this restriction induces more or less code-mixing in tweets (as compared to Facebook posts), and whether the reduced size of the context makes language identification even harder.

Furthermore, the present work has concentrated on code-mixing in romanized Indian social media texts, but there are other possible code-mixing cases such as Unicode and romanized Indian language text plus English, or with English words transliterated in Unicode. The following examples are collected from Twitter.

kishe poren ?
In which department you are studying ?
আমি – mechanical engineering তে ।
I am in mechanical engineering .

আউটিঙ এ এসেও এত বোরিং কাটবে সময়!
I am getting bored even in outing!

The language identification system described here mainly uses standard techniques such as character n-grams, dictionaries and SVM-classifiers. Incorporating other techniques and information sources are obvious targets for future work. For example, to use a sequence learning method such as Conditional Random Fields to capture patterns of sequences containing code switching, or combinations (ensembles) of different types of learners.

Acknowledgements

Thanks to Utsab Barman, Dublin City University, Ireland, for helping us in corpus collection-annotation, and to Dong Nguyen and Seza Doğruöz (University of Twente resp. Tilburg University, The Netherlands) for making their data set available.

References

Bashir U. Ahmed. 2005. *Detection of Foreign Words and Names in Written Text*. PhD Thesis, School of Computer Science and Information Systems, Pace University, New York, USA.

Beatrice Alex. 2008. *Automatic Detection of English Inclusions in Mixed-lingual Data with an Application to Parsing*. PhD Thesis, School of Informatics, University of Edinburgh, Edinburgh, UK.

Gisle Andersen. 2012. Semi-automatic approaches to Anglicism detection in Norwegian corpus data. In Cristiano Furiassi, Virginia Pulcini, and Félix Rodríguez González, editors, *The Anglicization of European lexis*, pages 111–130. John Benjamins.

Peter Auer. 1984. *Bilingual Conversation*. John Benjamins.

Timothy Baldwin and Marco Lui. 2010. Language identification: The long and the short of the matter. In *Proceedings of the 2010 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 229–237, Los Angeles, California, June. ACL.

Kenneth R. Beesley. 1988. Language identifier: A computer program for automatic natural-language identification of on-line text. In *Proceedings of the 29th Annual Conference of the American Translators Association*, pages 47–54, Medford, New Jersey.

Śailendra Biśvās. 2000. *Samsad Bengali-English dictionary*. Sahitya Samsad, Calcutta, India, 3 edition.

Zannie Bock. 2013. Cyber socialising: Emerging genres and registers of intimacy among young South African students. *Language Matters: Studies in the Languages of Africa*, 44(2):68–91.

Utsab Burman, 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 2014 EMNLP*, pages 13–23, Doha, Qatar, October. ACL. 1st Workshop on Computational Approaches to Code Switching.

Simon Carter. 2012. *Exploration and Exploitation of Multilingual Data for Statistical Machine Translation*. PhD Thesis, University of Amsterdam, Informatics Institute, Amsterdam, The Netherlands, December.

William D. Cavnar and John M. Trenkle. 1994. N-gram-based text categorization. In *Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, Nevada, April. UNLV Publications/Reprographics.

Joyce YC Chan, Houwei Cao, PC Ching, and Tan Lee. 2009. Automatic recognition of Cantonese-English code-mixing speech. *International Journal of Computational Linguistics and Chinese Language Processing*, 14(3):281–304.

Marc Damashek. 1995. Gauging similarity with n-grams: Language-independent categorization of text. *Science*, 267(5199):843–848.

Amitava Das, Tanik Saikh, Tapabrata Mondal, Asif Ekbal, and Sivaji Bandyopadhyay. 2010. English to Indian languages machine transliteration system at NEWS 2010. In *Proceedings of the 48th ACL*, pages 71–75, Uppsala, Sweden, July. ACL. 2nd Named Entities Workshop.

Ted Dunning. 1994. Statistical identification of language. Technical report, Computing Research Laboratory, New Mexico State University, Las Cruces, New Mexico, March.

Eric Fischer. 2011. Language communities of Twitter, October. `http://www.flickr.com/photos/walkingsf/6277163176/in/photostream/`.

Joseph Gafaranga and Maria-Carme Torras. 2002. Interactional otherness: Towards a redefinition of codeswitching. *International Journal of Bilingualism*, 6(1):1–22.

E. Mark Gold. 1967. Language identification in the limit. *Information and Control*, 10(5):447–474.

Thomas Gottron and Nedim Lipka. 2010. A comparison of language identification approaches on short, query-style texts. In *Advances in Information Retrieval: 32nd European Conference on IR Research, Proceedings*, pages 611–614, Milton Keynes, UK, March. Springer.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, November.

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. ACL.

Taofik Hidayat. 2012. An analysis of code switching used by facebookers (a case study in a social network site). BA Thesis, English Education Study Program, College of Teaching and Education (STKIP), Bandung, Indonesia, October.

Thorsten Joachims. 1999. Making large-scale support vector machine learning practical. In Bernhard Schölkopf, Christopher J.C. Burges, and Alexander J. Smola, editors, *Advances in Kernel Methods: Support Vector Learning*, chapter 11, pages 169–184. MIT Press, Cambridge, Massachusetts.

Aravind K. Joshi. 1982. Processing of sentences with intra-sentential code-switching. In *Proceedings of the 9th International Conference on Computational Linguistics*, pages 145–150, Prague, Czechoslovakia, July. ACL.

Mitesh M. Khapra, Salil Joshi, Ananthakrishnan Ramanathan, and Karthik Visweswariah. 2013. Offering language based services on social media by identifying user's preferred language(s) from romanized text. In *Proceedings of the 22nd International WWW*, volume Companion, pages 71–72, Rio de Janeiro, Brazil, May.

Ben King and Steven Abney. 2013. Labeling the languages of words in mixed-language documents using weakly supervised methods. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1110–1119, Atlanta, Georgia, June. ACL.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710, February.

David C. S. Li. 2000. Cantonese-English code-switching research in Hong Kong: a Y2K review. *World Englishes*, 19(3):305–322, November.

Constantine Lignos and Mitch Marcus. 2013. Toward web-scale analysis of codeswitching. In *87th Annual Meeting of the Linguistic Society of America*, Boston, Massachusetts, January. poster.

Marco Lui and Timothy Baldwin. 2014. Accurate language identification of twitter messages. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 17–25, Göteborg, Sweden, April. ACL. 5th Workshop on Language Analysis for Social Media.

Marco Lui, Jey Han Lau, and Timothy Baldwin. 2014. Automatic detection and language identification of multilingual documents. *Transactions of the Association for Computational Linguistics*, 2:27–40, February.

Paul McNamee. 2005. Language identification: A solved problem suitable for undergraduate instruction. *Journal of Computing Sciences in Colleges*, 20(3):94–101, February.

Pieter Muysken. 1995. Code-switching and grammatical theory. In Lesley Milroy and Pieter Muysken, editors, *One speaker, two languages: Cross-disciplinary perspectives on code-switching*, pages 177–198. Cambridge University Press, Cambridge, England.

Pieter Muysken. 2000. *Bilingual speech: A typology of code-mixing*. Cambridge University Press, Cambridge, England.

Dipak Narayan, Debasri Chakrabarti, Prabhakar Pande, and Pushpak Bhattacharyya. 2002. An experience in building the Indo WordNet — a WordNet for Hindi. In *Proceedings of the 1st International Conference on Global WordNet*, Mysore, India, January.

Rosalyn Negrón Goldbarg. 2009. Spanish-English codeswitching in email communication. *Language@Internet*, 6:article 3, February.

Dong Nguyen and A Seza Doğruöz. 2013. Word level language identification in online multilingual communication. In *Proceedings of the 2013 EMNLP*, pages 857–862, Seattle, Washington, October. ACL.

Nanyun Peng, Yiming Wang, and Mark Dredze. 2014. Learning polylingual topic models from code-switched social media documents. In *Proceedings of the 52nd ACL*, volume 2, short papers, pages 674–679, Baltimore, Maryland, June. ACL.

John M. Prager. 1997. Linguini: Language identification for multilingual documents. In *Proceedings of the 32nd Hawaii International Conference on Systems Sciences*, pages 1–11, Maui, Hawaii, January. IEEE.

Radim Řehůřek and Milan Kolkus. 2009. Language identification on the web: Extending the dictionary method. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing: Proceedings of the 10th International Conference*, number 5449 in Lecture Notes in Computer Science, pages 357–368, Mexico City, Mexico, March. Springer-Verlag.

Paul Rodrigues. 2012. *Processing Highly Variant Language Using Incremental Model Selection*. PhD Thesis, Indiana University, Dept. of Linguistics, Bloomington, Indiana, February.

Mike Rosner and Paulseph-John Farrugia. 2007. A tagging algorithm for mixed language identification in a noisy domain. In *Proceedings of the 8th Annual INTERSPEECH Conference*, volume 3, pages 1941–1944, Antwerp, Belgium, August. ISCA.

Hong Ka San. 2009. Chinese-English code-switching in blogs by Macao young people. MSc Thesis, Applied Linguistics, University of Edinburgh, Edinburgh, Scotland, August.

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

Latisha Asmaak Shafie and Surina Nayan. 2013. Languages, code-switching practice and primary functions of Facebook among university students. *Study in English Language Teaching*, 1(1):187–199, February.

Anil Kumar Singh and Jagadeesh Gorla. 2007. Identification of languages and encodings in a multilingual document. In *Proceedings of the 3rd Workshop on Building and Exploring Web Corpora*, pages 95–108, Louvain-la-Neuve, Belgium, September. Presses universitaires de Louvain.

Thamar Solorio and Yang Liu. 2008. Part-of-speech tagging for English-Spanish code-switched text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1051–1060, Honolulu, Hawaii, October. ACL.

Thamar Solorio, Melissa Sherman, Yang Liu, Lisa M. Bedore, Elisabeth D. Peña, and Aquiles Iglesias. 2011. Analyzing language samples of Spanish-English bilingual children for the automated prediction of language dominance. *Natural Language Engineering*, 17(3):367–395, July.

Susanna Sotillo. 2012. *Ehhhh utede hacen plane sin mi???:@ im feeling left out:(* form, function and type of code switching in SMS texting. In *ICAME 33 Corpora at the centre and crossroads of English linguistics*, pages 309–310, Leuven, Belgium, June. Katholieke Universiteit Leuven.

William John Teahan. 2000. Text classification and segmentation using minimum cross-entropy. In *Proceedings of the 6th International Conference on Computer-Assisted Information Retrieval (Recherche d'Information Assistée par Ordinateur, RIAO 2000)*, pages 943–961, Paris, France, April.

Clare Voss, Stephen Tratz, Jamal Laoudi, and Douglas Briesch. 2014. Finding romanized Arabic dialect in code-mixed tweets. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pages 188–199, Reykjavík, Iceland, May. ELRA.

Jochen Weiner, Ngoc Thang Vu, Dominic Telaar, Florian Metze, Tanja Schultz, Dau-Cheng Lyu, Eng-Siong Chng, and Haizhou Li. 2012. Integration of language identification into a recognition system for spoken conversations containing code-switches. In *Proceedings of the 3rd Workshop on Spoken Language Technologies for Under-resourced Languages*, Cape Town, South Africa, May.

Hiroshi Yamaguchi and Kumiko Tanaka-Ishii. 2012. Text segmentation by language using minimum description length. In *Proceedings of the 50th ACL*, volume 1, pages 969–978, Jeju, Korea, July. ACL.