Code Mixing: A Challenge for Language Identification in the Language of Social Media

Utsab Barman, Amitava Das[†], Joachim Wagner and Jennifer Foster
CNGL Centre for Global Intelligent Content, National Centre for Language Technology
School of Computing, Dublin City University, Dublin, Ireland
[†]Department of Computer Science and Engineering
University of North Texas, Denton, Texas, USA
{ubarman,jwagner,jfoster}@computing.dcu.ie
amitava.das@unt.edu

Abstract

In social media communication, multilingual speakers often switch between languages, and, in such an environment, automatic language identification becomes both a necessary and challenging task. In this paper, we describe our work in progress on the problem of automatic language identification for the language of social media. We describe a new dataset that we are in the process of creating, which contains Facebook posts and comments that exhibit code mixing between Bengali, English and Hindi. We also present some preliminary word-level language identification experiments using this dataset. Different techniques are employed, including a simple unsupervised dictionary-based approach, supervised word-level classification with and without contextual clues, and sequence labelling using Conditional Random Fields. We find that the dictionary-based approach is surpassed by supervised classification and sequence labelling, and that it is important to take contextual clues into consideration.

1 Introduction

Automatic processing and understanding of Social Media Content (SMC) is currently attracting much attention from the Natural Language Processing research community. Although English is still by far the most popular language in SMC, its dominance is receding. Hong et al. (2011), for example, applied an automatic language detection algorithm to over 62 million tweets to identify the top 10 most popular languages on Twitter. They found that only half of the tweets were in English. Moreover, mixing multiple languages together (*code mixing*) is a popular trend in social media users from language-dense areas (Cárdenas-Claros and Isharyanti, 2009; Shafie and Nayan, 2013). In a scenario where speakers switch between languages within a conversation, sentence or even word, the task of automatic language identification becomes increasingly important to facilitate further processing.

Speakers whose first language uses a non-Roman alphabet write using the Roman alphabet for convenience (phonetic typing) which increases the likelihood of code mixing with a Roman-alphabet language. This can be especially observed in South-East Asia and in the Indian subcontinent. The following is a code mixing comment taken from a Facebook group of Indian university students:

> Original: *Yaar tu to*, GOD *hain*. **tui** JU **te ki korchis**? Hail u man!
>
> Translation: Buddy you are GOD. What are you doing in JU? Hail u man!

This comment is written in three languages: English, Hindi (italics), and Bengali (boldface). For Bengali and Hindi, phonetic typing has been used.

We follow in the footsteps of recent work on language identification for SMC (Hughes et al., 2006; Baldwin and Lui, 2010; Bergsma et al., 2012), focusing specifically on the problem of *word-level* language identification for code mixing SMC. Our corpus for this task is collected from Facebook and contains instances of *Bengali(BN)-English(EN)-Hindi(HI)* code mixing.

The paper is organized as follows: in Section 2, we review related research in the area of code mixing and language identification; in Section 3, we describe our code mixing corpus, the data it-

self and the annotation process; in Section 4, we list the tools and resources which we use in our language identification experiments, described in Section 5. Finally, in Section 6, we conclude and provide suggestions for future research on this topic.

2 Background and Related Work

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 has been less work on *automatic language identification for multilingual code-mixed texts*. Before turning to that topic, we first briefly survey studies on the general characteristics of code mixing.

Code mixing is a normal, natural product of bilingual and multilingual language use. Significant studies of the phenomenon can be found in the linguistics literature (Milroy and Muysken, 1995; Alex, 2008; Auer, 2013). These works mainly discuss the sociological and conversational necessities behind code mixing as well as its linguistic nature. Scholars distinguish between *inter-sentence*, *intra-sentence* and *intra-word* code mixing.

Several researchers have investigated the reasons for and the types of code mixing. Initial studies on Chinese-English code mixing in Hong Kong (Li, 2000) and Macao (San, 2009) indicated that mainly linguistic motivations were triggering the code mixing in those highly bilingual societies. Hidayat (2012) showed that Facebook users tend to mainly use inter-sentential switching over intra-sentential, and report 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 mixing in social media text was also noted in the study by San (2009), which compared the mixing in blog posts to that in the spoken language in Macao. Dewaele (2010) claims that 'strong emotional arousal' increases the frequency of code mixing. Dey and Fung (2014) present a speech corpus of English-Hindi code mixing in student interviews and analyse the motivations for code mixing and in what grammatical contexts code mixing occurs.

Turning to the work on automatic analysis of code mixing, there have been some studies on detecting code mixing in speech (Solorio and Liu, 2008a; Weiner et al., 2012). Solorio and Liu (2008b) try to predict the points inside a set of spoken Spanish-English sentences where the speakers switch between the two languages. 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 (Farrugia, 2004; Rosner and Farrugia, 2007). Yamaguchi and Tanaka-Ishii (2012) perform language identification using artificial multilingual data, created by randomly sampling text segments from monolingual documents. King and Abney (2013) used weakly semi-supervised methods to perform word-level language identification. A dataset of 30 languages has been used in their work. They explore several language identification approaches, including a Naive Bayes classifier for individual word-level classification and sequence labelling with Conditional Random Fields trained with Generalized Expectation criteria (Mann and McCallum, 2008; Mann and McCallum, 2010), which achieved the highest scores. Another very recent work on this topic is (Nguyen and Doğruöz, 2013). They report on language identification experiments performed on Turkish and Dutch forum data. Experiments have been carried out using language models, dictionaries, logistic regression classification and Conditional Random Fields. They find that language models are more robust than dictionaries and that contextual information is helpful for the task.

3 Corpus Acquisition

Taking into account the claim that code mixing is frequent among speakers who are *multilingual* and *younger in age* (Cárdenas-Claros and Isharyanti, 2009), we choose an Indian student community between the 20-30 year age group as our data source. India is a country with 30 spoken languages, among which 22 are official. code mixing is very frequent in the Indian sub-continent because languages change within very short geo-distances and people generally have a basic knowledge of their neighboring languages.

A Facebook group[1] and 11 Facebook users (known to the authors) were selected to obtain publicly available posts and comments. The Facebook graph API explorer was used for data collection. Since these Facebook users are from West Bengal, the most dominant language is Bengali

[1] https://www.facebook.com/jumatrimonial

(Native Language), followed by English and then Hindi (National Language of India). The posts and comments in Bengali and Hindi script were discarded during data collection, resulting in 2335 posts and 9813 comments.

3.1 Annotation

Four annotators took part in the annotation task. Three were computer science students and the other was one of the authors. The annotators are proficient in all three languages of our corpus. A simple annotation tool was developed which enabled these annotators to identify and distinguish the different languages present in the content by tagging them. Annotators were supplied with 4 basic tags (viz. *sent*ence, *frag*ment, *incl*usion and *wlcm* (word-level code mixing)) to annotate different levels of code mixing. Under each tag, six attributes were provided, viz. *English (en), Bengali (bn), Hindi (hi), Mixed (mixd), Universal (univ) and Undefined (undef)*. The attribute *univ* is associated with symbols, numbers, emoticons and universal expressions (e.g. *hahaha*, *lol*). The attribute *undef* is specified for a sentence or a word for which no language tags can be attributed or cannot be categorized as *univ*. In addition, annotators were instructed to annotate named entities separately. What follows are descriptions of each of the annotation tags.

Sentence (sent): This tag refers to a sentence and can be used to mark *inter-sentential code mixing*. Annotators were instructed to identify a sentence with its base language (e.g. *en*, *bn*, *hi* and *mixd*) or with other types (e.g. *univ*, *undef*) as the first task of annotation. Only the attribute *mixd* is used to refer to a sentence which contains multiple languages in the same proportion. A sentence may contain any number of inclusions, fragments and word-level code mixing. A sentence can be attributed as *univ* if and only if it contains symbols, numbers, emoticons, chat acronyms and no other words (Hindi, English or Bengali). A sentence can be attributed as *undef* if it is not a sentence marked as *univ* and has words/tokens that can not be categorized as Hindi, English or Bengali. Some examples of sentence-level annotations are the following:

1. **English-Sentence:**
   [sent-lang="en"] *what a.....6 hrs long...but really nice tennis....* [/sent]
2. **Bengali-Sentence:**
   [sent-lang="bn"] *shubho nabo borsho.. :)* [/sent]
3. **Hindi Sentence:**
   [sent-lang="hi"] *karwa sachh ..... :(* [/sent]
4. **Mixed-Sentence:**
   [sent-lang="mixd"] [frag-lang="hi"] *oye hoye ..... angreji me kahte hai ke* [/frag] [frag-lang="en"] *I love u.. !!!* [/frag] [/sent]
5. **Univ-Sentence:**
   [sent-lang="univ"] *hahahahahahah....!!!!!* [/sent]
6. **Undef-Sentence:**
   [sent-lang="undef"] *Hablando de una triple amenaza.* [/sent]

Fragment (frag): This refers to a group of foreign words, grammatically related, in a sentence. The presence of this tag in a sentence conveys that *intra-sentential code mixing* has occurred within the sentence boundary. Identification of fragments (if present) in a sentence was the second task of annotation. A *sentence (sent)* with attribute *mixd* must contain multiple *fragments (frag)* with a specific language attribute. In the fourth example above, the sentence contains a Hindi fragment *oye hoye ..... angreji me kahte hai ke* and an English fragment *I love u.. !!!*, hence it is considered as a *mixd* sentence. A fragment can have any number of inclusions and word-level code mixing. In the first example below, *Jio* is a popular Bengali word appearing in the English fragment *Jio.. good joke*, hence tagged as a Bengali inclusion. One can argue that the word *Jio* could be a separate Bengali inclusion (i.e. can be tagged as a Bengali inclusion outside the English fragment). But looking at the syntactic pattern and the sense expressed by the comment, the annotator kept it as a single unit. In the second example below, an instance of word-level code mixing, *typer*, has been found in an English fragment (where the root English word *type* has the Bengali suffix *r*).

1. **Fragment with Inclusion:**
   [sent-lang="mixd"] [frag-lang="en"] [incl-lang="bn"] *Jio..* [/incl] *good joke* [/frag] [frag lang="bn"] *"amar Babin"* [/frag] [/sent]
2. **Fragment with Word-Level code mixing:**
   [sent-lang="mixd"] [frag-lang="en"] *" I will find u and marry you "* [/frag] [frag-lang="bn"] [wlcm-type="en-and-bn-suffix"] *typer* [/wlcm] *hoe glo to! :D* [/frag] [/sent]

Inclusion (incl): An inclusion is a foreign word or phrase in a sentence or in a fragment which is assimilated or used very frequently in native language. Identification of inclusions can be performed after annotating a sentence and fragment (if present in that sentence). An inclusion within a sentence or fragment also denotes *intra-sentential code mixing*. In the example below, *seriously* is an English inclusion which is assimilated in today's colloquial Bengali and Hindi. The only tag that an inclusion may contain is word-level code mixing.

1. **Sentence with Inclusion:**
   [sent-lang="bn"] *Na re* [incl-lang="en"] *seriously* [/incl] *ami khub kharap achi.* [/sent]

Word-Level code mixing (wlcm): This is the smallest unit of code mixing. This tag was introduced to capture *intra-word code mixing* and denotes cases where code mixing has occurred within a single word. Identifying word-level code mixing is the last task of annotation. Annotators were told to mention the type of word-level code mixing in the form of an attribute (Base Language + Second Language) format. Some examples are provided below. In the first example below, the root word *class* is English and *e* is an Bengali suffix that has been added. In the third example below, the opposite can be observed – the root word *Kando* is Bengali, and an English suffix *z* has been added. In the second example below, a named entity *suman* is present with a Bengali suffix *er*.

1. **Word-Level code mixing (EN-BN):**
   [wlcm-type="en-and-bn-suffix"] *classe* [/wlcm]
2. **Word-Level code mixing (NE-BN):**
   [wlcm-type="NE-and-bn-suffix"] *sumaner* [/wlcm]
3. **Word-Level code mixing (BN-EN):**
   [wlcm-type="bn-and-en-suffix"] *kandoz* [/wlcm]

3.1.1 Inter Annotator Agreement

We calculate word-level inter annotator agreement (Cohen's Kappa) on a subset of 100 comments (randomly selected) between two annotators. Two annotators are in agreement about a word if they both annotate the word with the same attribute (*en, bn, hi, univ, undef*), regardless of whether the word is inside an inclusion, fragment or sentence. Our observations that the word-level annotation process is not a very ambiguous task and that annotation instruction is also straightforward are confirmed in a high inter-annotator agreement (IAA) with a Kappa value of 0.884.

3.2 Data Characteristics

Tag-level and word-level statistics of annotated data that reveal the characteristics of our data set are described in Table 1 and in Table 2 respectively. More than 56% of total sentences and almost 40% of total tokens are in Bengali, which is the dominant language of this corpus. English is the second most dominant language covering almost 33% of total tokens and 35% of total sentences. The amount of Hindi data is substantially lower – nearly 1.75% of total tokens and 2% of total sentences. However, English inclusions (84% of total inclusions) are more prominent than Hindi or Bengali inclusions and there are a substantial number of English fragments (almost 52% of total fragments) present in our corpus. This means that English is the main language involved in the code mixing.

Statistics of Different Tags						
Tags	**En**	**Bn**	**Hi**	**Mixd**	**Univ**	**Undef**
sent	5,370	8,523	354	204	746	15
frag	288	213	40	0	6	0
incl	7,377	262	94	0	1,032	1
wlcm						477
Name Entity						3,602
Acronym						691

Table 1: Tag-level statistics

Word-Level Tag	**Count**
EN	66,298
BN	79,899
HI	3,440
WLCM	633
NE	5,233
ACRO	715
UNIV	39,291
UNDEF	61

Table 2: Word-level statistics

3.2.1 Code Mixing Types

In our corpus, inter- and intra-sentential code mixing are more prominent than word-level code mixing, which is similar to the findings of (Hidayat, 2012) . Our corpus contains every type of code mixing in English, Hindi and Bengali viz. inter/intra sentential and word-level as described in the previous section. Some examples of different types of code mixing in our corpus are presented below.

1. **Inter-Sentential:**
   [sent-lang="hi"] *Itna izzat diye aapne mujhe* !!! [/sent]
   [sent-lang="en"] *Tears of joy. :'(:'(* [/sent]
2. **Intra-Sentential:**
   [sent-lang="bn"] [incl-lang="en"] *by d way* [/incl] *ei* [frag-lang="en"] *my craving arms shall forever remain empty .. never hold u close ..* [/frag] *line ta baddo* [incl-lang="en"] *cheezy* [/incl] :P ;) [/sent]
3. **Word-Level:**
   [sent-lang="bn"] [incl-lang="en"] *1st yr* [/incl] *eo to ei* [wlcm-type="en+bnSuffix"] *tymer* [/wlcm] *modhye sobar jute jay ..* [/sent]

3.2.2 Ambiguous Words

Annotators were instructed to tag an English word as English irrespective of any influence of word borrowing or foreign inclusion but an inspection of the annotations revealed that English words were sometimes annotated as Bengali or Hindi. To understand this phenomenon we processed the list of language (EN,BN and HI) word types (total 26,475) and observed the percentage of types that were not always annotated with the one language throughout the corpus. The results are presented in Table 3. Almost 7% of total types are ambiguous (i.e. tagged in different languages during annotation). Among them, a substantial amount (5.58%) are English/Bengali.

Label(s)	Count	Percentage
EN	9,109	34.40
BN	14,345	54.18
HI	1,039	3.92
EN or BN	1,479	5.58
EN or HI	61	0.23
BN or HI	277	1.04
EN or BN or HI	165	0.62

Table 3: Statistics of ambiguous and monolingual word types

There are two reasons why this is happening:

Same Words Across Languages Some words are the same (e.g. *baba*, *maa*, *na*, *khali*) in Hindi and Bengali because both of the languages originated from a single language *Sanskrit* and share a good amount of common vocabulary. It also occurred in English-Hindi and English-Bengali as a result of *word borrowing*. Most of these are commonly used inclusions like *clg*, *dept*, *question*, *cigarette*, and *topic*. Sometimes the annotators were careful enough to tag such words as English and sometimes these words were tagged in the annotators' native languages. During cross checking of the annotated data the same error patterns were observed for multiple annotators, i.e. tagging commonly used foreign words into native language. It only demonstrates that these English words are highly assimilated in the conversational vocabulary of Bengali and Hindi.

Phonetic Similarity of Spellings Due to phonetic typing some words share the same surface form across two and sometimes across three languages. As an example, *to* is a word in the three languages: it has occurred 1209 times as English, 715 times as Bengali and 55 times as Hindi in our data. The meaning of these words (e.g. *to*, *bolo*, *die*) are different in different languages. This phenomenon is perhaps exacerbated by the trend towards short and noisy spelling in SMC.

4 Tools and Resources

We have used the following resources and tools in our experiment.

Dictionaries

1. **British National Corpus (BNC):** We compile a word frequency list from the BNC (Aston and Burnard, 1998).
2. **SEMEVAL 2013 Twitter Corpus (SemevalTwitter):** To cope with the language of social media we use the SEMEVAL 2013 (Nakov et al., 2013) training data for the Twitter sentiment analysis task. This data comes from a popular social media site and hence is likely to reflect the linguistic properties of SMC.
3. **Lexical Normalization List (LexNormList):** Spelling variation is a well-known phenomenon in SMC. We use a lexical normalization dictionary created by Han et al. (2012) to handle the different spelling variations in our data.

Machine Learning Toolkits

1. **WEKA:** We use the Weka toolkit (Hall et al., 2009) for our experiments in decision tree training.

2. **MALLET:** CRF learning is applied using the MALLET toolkit (McCallum, 2002).

3. **Liblinear:** We apply Support Vector Machine (SVM) learning with a linear kernel using the Liblinear package (Fan et al., 2008).

NLP Tools For data tokenization we used the CMU Tweet-Tokenizer (Owoputi et al., 2013).

5 Experiments

Since our training data is entirely labelled at the word-level by human annotators, we address the word-level language identification task in a fully supervised way.

Out of the total data, 15% is set aside as a blind test set, while the rest is employed in our experiments through a 5-fold cross-validation setup. There is a substantial amount of token overlap between the cross-validation data and the test set – 88% of total EN tokens, 86% of total Bengali tokens and 57% of total Hindi tokens of the test set are present in the cross-validation data.[2]

We address the problem of word-level in three different ways:

1. A simple heuristic-based approach which uses a combination of our dictionaries to classify the language of a word
2. Word-level classification using supervised machine learning with SVMs but no contextual information
3. Word-level classification using supervised machine learning with SVMs and sequence labelling using CRFs, both employing contextual information

Named entities and instances of word-level code mixing are excluded from evaluation. For systems which do not take the context of a word into account, i.e. the dictionary-based approach (Section 5.1) and the SVM approach without contextual clues (Section 5.2), named entities and instances of word-level code mixing can be safely excluded from training. For systems which do take context into account, the CRF system (Section 5.3.1) and the SVM system with contextual clues (Section 5.3.2), these are included in training, because to exclude them would result in unrealistic contexts. This means that these systems can classify a word to be a named entity or an instance of word-level code mixing. To avoid this, we implement a post-processor which backs off in these cases to a system which hasn't seen named entities or word-level code mixing in training (see Section 5.3).

[2]We found 25 comments and 17 posts common between the cross-validation data and the test set. The reason for this is that users of social media often express themselves in a concise way. Almost all of these common data consisted of 1 to 3 token(s). In most of the cases these tokens were emoticons, symbols or universal expressions such as *wow* and *lol*. As the percentage of these comments is low, we keep these comments as they are.

5.1 Dictionary-Based Detection

We start with dictionary-based language detection. Generally a dictionary-based language detector predicts the language of a word based on its frequency in multiple language dictionaries. In our data the Bengali and Hindi tokens are phonetically typed. As no such transliterated dictionary is, to our knowledge, available for Bengali and Hindi, we use the training set words as dictionaries. For words that have multiple annotations in training data (ambiguous words), we select the majority tag based on frequency, e.g. the word *to* will always be tagged as English.

Our English dictionaries are those described in Section 4 (*BNC, LexNormList, SemEvalTwitter*) and the training set words. For *LexNormList*, we have no frequency information, and so we consider it as a simple word list. To predict the language of a word, dictionaries with normalized frequency were considered first (BNC, SemEvalTwitter, Training Data), if not found, word list look-up was performed. The predicted language is chosen based on the dominant language(s) of the corpus if the word appears in multiple dictionaries with same frequency or if the word does not appear in any dictionary or list.

A simple rule-based method is applied to predict universal expressions. A token is considered as *univ* if any of the following conditions satisfies:

- All characters of the token are symbols or numbers.
- The token contains certain repetitions identified by regular expressions.(e.g. *hahaha*).
- The token is a hash-tag or an URL or mention-tags (e.g. *@Sumit*).
- Tokens (e.g. *lol*) identified by a word list compiled from the relevant 4/5th of the training data.

Table 4 shows the results of dictionary-based detection obtained from 5-fold cross-validation averaging. We try different combinations and frequency thresholds of the above dictionaries. We find that using a normalized frequency is helpful

and that a combination of *LexNormList* and *Training Data* dictionaries is suited best for our data. Hence, we consider this as our baseline language identification system.

Dictionary	Accuracy(%)
BNC	80.09
SemevalTwitter	77.61
LexNormList	79.86
Training Data	90.21
LexNormList+TrainingData (Baseline)	**93.12**

Table 4: Average cross-validation accuracy of dictionary-based detection

5.2 Word-Level Classification without Contextual Clues

The following feature types are employed:

1. **Char-n-grams (G):** We start with a character n-gram-based approach (Cavnar and Trenkle, 1994), which is most common and followed by many language identification researchers. Following the work of King and Abney (2013), we select character n-grams (n=1 to 5) and the word as the features in our experiments.
2. **Presence in Dictionaries (D):** We use presence in a dictionary as a features for all available dictionaries in previous experiments.
3. **Length of words (L):** Instead of using the raw length value as a feature, we follow our previous work (Rubino et al., 2013; Wagner et al., 2014) and create multiple features for length using a decision tree (J48). We use length as the only feature to train a decision tree for each fold and use the nodes obtained from the tree to create boolean features.
4. **Capitalization (C):** We use 3 boolean features to encode capitalization information: whether any letter in the word is capitalized, whether all letters in the word are capitalized and whether the first letter is capitalized.

We perform experiments with an SVM classifier (linear kernel) for different combination of these features.[3] Parameter optimizations (C range 2^{-15} to 2^{10}) for SVM are performed for each feature set and best cross-validation accuracy is found for the GDLC-based run (94.75%) at C=1 (see Table 5 and Fig. 1).

[3]According to (Hsu et al., 2010) the SVM linear kernel with parameter C optimization is good enough when dealing with a large number of features. Though an RBF kernel can be more effective than a linear one, it is possible only after proper optimization of C and γ parameters, which is computational expensive for such a large feature set.

Features	Accuracy	Features	Accuracy
G	94.62	GD	94.67
GL	94.62	GDL	94.73
GC	94.64	GDC	94.72
GLC	94.64	**GDLC**	**94.75**

Table 5: Average cross-validation accuracy for SVM word-level classification (without context), G = char-n-gram, L = binary length features, D = presence in dictionaries and C = capitalization features

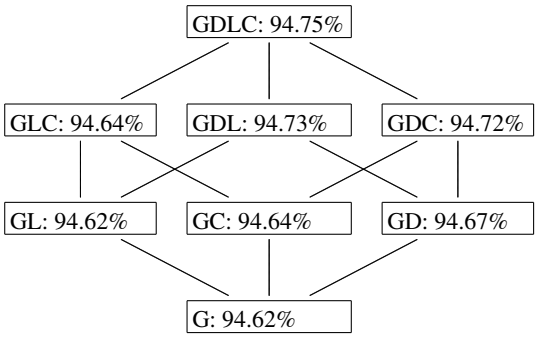

Figure 1: Average cross-validation accuracy for SVM word-level classification (without context), G = char-n-gram, L = binary length features, D = presence in dictionaries and C = capitalization features: cube visualization

We also investigate the use of a dictionary-to-char-n-gram back-off model – the idea is to apply the char-n-gram model SVM-GDLC for those words for which a majority-based decision is taken during dictionary-based detection. However, it does not outperform the SVM. Hence, we select SVM-GDLC for the next steps of our experiments as the best exemplar of our individual word-level classifier (without contextual clues).

5.3 Language Identification with Contextual Clues

Contextual clues can play a very important role in word-level language identification. As an example, a part of a comment is presented from cross-validation fold 1 that contains the word *die* which is wrongly classified by the SVM classifier. The frequency of *die* in the training set of fold 1 is 6 for English, 31 for Bengali and 0 for Hindi.

Gold Data: *..../univ the/en movie/en for/en which/en i/en can/en* ***die/en*** *for/en*

Features	Order-0	Order-1	Order-2
G	92.80	95.16	95.36
GD	93.42	95.59	95.98
GL	92.82	95.14	95.41
GDL	93.47	95.60	95.94
GC	92.07	94.60	95.05
GDC	93.47	95.62	**95.98**
GLC	92.36	94.53	95.02
GDLC	93.47	95.58	95.98

Table 6: Average cross-validation accuracy of CRF-based predictions where G = char-n-gram, L = length feature, D = single dictionary-based labels (baseline system) and C = capitalization features

...../univ

SVM Output: *..../univ the/en movie/en for/en which/en i/en can/en* ***die/bn*** *for/en ...../univ*

We now investigate whether contextual information can correct the mis-classified tags.

Although named entities and word-level code mixing are excluded from evaluation, when dealing with context it is important to consider named entity and word-level code mixing during training because these may contain some important information. We include these tokens in the training data for our context-based experiments, labelling them as *other*. The presence of this new label may affect the prediction for a language token during classification and sequence labelling. To avoid this situation, a 4-way (*bn, hi, en, univ*) backoff classifier is trained separately on English, Hindi, Bengali and universal tokens. During evaluation of any context-based system we discard named entity and word-level code mixing from the prediction of that system. If any of the remaining tokens is predicted as *other* we back off to the decision of the 4-way classifier for that token. For the CRF experiments (Section 5.3.1), the backoff classifier is a CRF system, and, for the SVM experiments (Section 5.3.2), the backoff classifier is an SVM system.

5.3.1 Conditional Random Fields (CRF)

As our goal is to apply contextual clues, we first employ Conditional Random Fields (CRF), an approach which takes history into account in predicting the optimal sequence of labels. We employ a linear chain CRF with an increasing order (Order-0, Order-1 and Order-2) with 200 iterations for different feature combinations (used in SVM-based runs). However, we observe that accuracy of CRF based runs decreases when binarized length features (see Section 5.2 and dictionary features (a feature for each dictionary) are involved. Hence, we use the dictionary-based predictions of the baseline system to generate a single dictionary feature for each token and only the raw length value of a token instead of binarized length features. The results are presented in Table 6 and the second order results are visualized in Fig. 2.

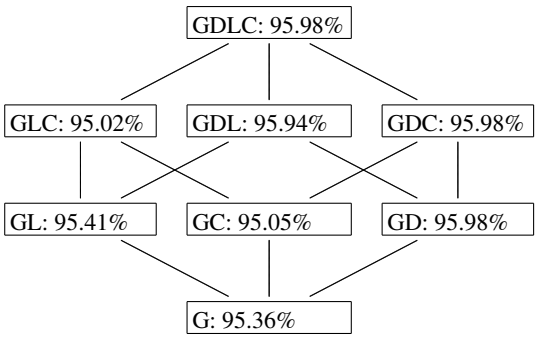

Figure 2: CRF Order-2 results: cube visualisation G = char-n-gram, L = binary length features, D = presence in dictionaries and C = capitalization features

Context	Accuracy (%)
GDLC + P_1	94.66
GDLC + P_2	94.55
GDLC + N_1	94.53
GDLC + N_2	94.37
GDLC + P_1N_1	**95.14**
GDLC + P_2N_2	94.55

Table 7: Average cross-validation accuracy of SVM (GDLC) context-based runs, where P-i = previous i word(s) , N-i = next i word(s)

As expected, the performance increases as the order increases from zero to one and two. The use of a single dictionary feature is also helpful. The results for GDC, GDLC, and GD based runs are almost similar (95.98%). However, we choose the GDC system because it performed slightly better (95.989%) than the GDLC (95.983%) and the GD (95.983%) systems.

5.3.2 SVM with Context

We also add contextual clues to our SVM classifier. To obtain contextual information we include the previous and next two words as features in the SVM-GDLC-based run.[4] All possible com-

[4]We also experimented with extracting all GDLC features for the context words but this did not help.

binations are considered during experiments (Table 7). After C parameter optimization, the best cross-validation accuracy is found for the P_1N_1 (one word previous and one word next) run with C=0.125 (95.14%).

5.4 Test Set Results

We apply our best dictionary-based system, our best SVM system (with and without context) and our best CRF system to the held-out test set. The results are shown in Table 8. Our best result is achieved using the CRF model (95.76%).

5.5 Error Analysis

Manual error analysis shows the limitations of these systems. The word-level classifier without contextual clues does not perform well with Hindi data. The number of Hindi tokens is quite low. Only 2.4% (4,658) of total tokens of the training data are Hindi, out of which 55.36% are bilingually ambiguous and 29.51% are tri-lingually ambiguous tokens. Individual word-level systems often fail to assign proper labels to ambiguous words, but adding context information helps to overcome this problem. Considering the previous example of *die*, both context-based SVM and CRF systems classify it properly. Though the final system CRF-GDC performs well, it also has some limitations, failing to identify the language for the tokens which appear very frequently in three languages (e.g. *are*, *na*, *pic*).

6 Conclusion

We have presented an initial study on automatic language identification with Indian language code mixing from social media communication. We described our dataset of Bengali-Hindi-English Facebook comments and we presented the results of our word-level classification experiments on this dataset. Our experimental results lead us to conclude that character n-gram features are useful for this task, contextual information is also important and that information from dictionaries can be effectively incorporated as features.

In the future we plan to apply the techniques and feature sets that we used in these experiments to other datasets. We have already started this by applying variants of the systems presented here to the Nepali-English and Spanish-English datasets which were introduced as part of the 2014 code mixing shared task (Solorio et al., 2014; Barman et al., 2014).

We did not include word-level code mixing in our experiments – in our future experiments we will explore ways to identify and segment this type of code mixing. It will be also important to find the best way to handle inclusions since there is a fine line between word borrowing and code mixing.

Acknowledgements

This research is supported by the Science Foundation Ireland (Grant 12/CE/I2267) as part of CNGL (www.cngl.ie) at Dublin City University. The authors wish to acknowledge the DJEI/DES/SFI/HEA for the provision of computational facilities and support. Our special thanks to Soumik Mandal from Jadavpur University, India for coordinating the annotation task. We also thank the administrator of JUMatrimonial and the 11 Facebook users who agreed that we can use their posts for their support and permission.

References

Beatrice Alex. 2008. *Automatic detection of English inclusions in mixed-lingual data with an application to parsing*. Ph.D. thesis, School of Informatics, The University of Edinburgh, Edinburgh, UK.

Guy Aston and Lou Burnard. 1998. *The BNC handbook: exploring the British National Corpus with SARA*. Capstone.

Peter Auer. 2013. *Code-Switching in Conversation: Language, Interaction and Identity*. Routledge.

Timothy Baldwin and Marco Lui. 2010. Language identification: The long and the short of the matter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 229–237. Association for Computational Linguistics.

Utsab Barman, Joachim Wagner, Grzegorz Chrupała, and Jennifer Foster. 2014. DCU-UVT: Word-level language classification with code-mixed data. In *Proceedings of the First Workshop on Computational Approaches to Code-Switching. EMNLP 2014, Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar. Association for Computational Linguistics.

Shane Bergsma, Paul McNamee, Mossaab Bagdouri, Clayton Fink, and Theresa Wilson. 2012. Language identification for creating language-specific Twitter collections. In *Proceedings of the Second Workshop on Language in Social Media*, pages 65–74. Association for Computational Linguistics.

System	Precision (%)				Recall (%)				Accuracy
	EN	*BN*	*HI*	*UNIV*	*EN*	*BN*	*HI*	*UNIV*	(%)
Baseline (Dictionary)	92.67	90.73	80.64	99.67	92.28	94.63	43.47	94.99	93.64
SVM-GDLC	92.49	94.89	80.31	99.34	96.23	94.28	44.92	97.07	95.21
SVM-P_1N_1	93.51	95.56	83.18	99.42	96.63	95.23	55.94	96.95	95.52
CRF-GDC	94.77	94.88	91.86	99.34	95.65	96.22	55.65	97.73	95.76

Table 8: Test set results for Baseline (Dictionary), SVM-GDLC, SVM-P1N1 and CRF-GDC

MS Cárdenas-Claros and N Isharyanti. 2009. Code-switching and code-mixing in internet chatting: Between'yes,"ya,'and'si'-a case study. *The Jalt Call Journal*, 5(3):67–78.

William B. Cavnar and John M. Trenkle. 1994. N-gram-based text categorization. In Theo Pavlidis, editor, *Proceedings of SDAIR-94, Third Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175.

Jean-Marc Dewaele. 2010. *Emotions in Multiple Languages*. Palgrave Macmillan.

Anik Dey and Pascale Fung. 2014. A Hindi-English code-switching corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2410–2413, Reykjavik, Iceland. European Language Resources Association (ELRA).

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Paulseph-John Farrugia. 2004. TTS pre-processing issues for mixed language support. In *Proceedings of CSAW'04, the second Computer Science Annual Workshop*, pages 36–41. Department of Computer Science & A.I., University of Malta.

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*, pages 611–614. Springer.

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

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. Association for Computational Linguistics.

Taofik Hidayat. 2012. An analysis of code switching used by facebookers: a case study in a social network site. Student essay for the study programme "Pendidikan Bahasa Inggris" (English Education) at STKIP Siliwangi Bandung, Indonesia, `http://publikasi.stkipsiliwangi.ac.id/files/2012/10/08220227-taofik-hidayat.pdf`.

Lichan Hong, Gregorio Convertino, and Ed H. Chi. 2011. Language matters in twitter: A large scale study. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM-11)*, pages 518–521, Barcelona, Spain. Association for the Advancement of Artificial Intelligence.

Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. 2010. A practical guide to support vector classification. Technical report. Department of Computer Science, National Taiwan University, Taiwan, `https://www.cs.sfu.ca/people/Faculty/teaching/726/spring11/svmguide.pdf`.

Baden Hughes, Timothy Baldwin, Steven Bird, Jeremy Nicholson, and Andrew MacKinlay. 2006. Reconsidering language identification for written language resources. In *Proc. of the 5th edition of the International Conference on Language Resources and Evaluation (LREC 2006)*, pages 485–488, Genoa, Italy.

Aravind K. Joshi. 1982. Processing of sentences with intra-sentential code-switching. In J. Horecký, editor, *Proceedings of the 9th conference on Computational linguistics - Volume 1 (COLING'82)*, pages 145–150. Academia Praha, North-Holland Publishing Company.

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. Association for Computational Linguistics.

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

Gideon S. Mann and Andrew McCallum. 2008. Generalized expectation criteria for semi-supervised learning of conditional random fields. In *Proceedings of ACL-08: HLT*, pages 870–878, Columbus, Ohio. Association for Computational Linguistics.

Gideon S. Mann and Andrew McCallum. 2010. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *The Journal of Machine Learning Research*, 11:955–984.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Lesley Milroy and Pieter Muysken, editors. 1995. *One speaker, two languages: Cross-disciplinary perspectives on code-switching*. Cambridge University Press.

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. Semeval-2013 task 2: Sentiment analysis in twitter. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 312–320, Atlanta, Georgia, USA. Association for Computational Linguistics.

Dong Nguyen and A. Seza Doğruöz. 2013. Word level language identification in online multilingual communication. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 857–862, Seattle, Washington, USA. Association for Computational Linguistics.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2013)*, pages 380–390, Atlanta, Georgia. Association for Computational Linguistics.

Mike Rosner and Paulseph-John Farrugia. 2007. A tagging algorithm for mixed language identification in a noisy domain. In *INTERSPEECH-2007, 8th Annual Conference of the International Speech Communication Association*, pages 190–193. ISCA Archive.

Raphael Rubino, Joachim Wagner, Jennifer Foster, Johann Roturier, Rasoul Samad Zadeh Kaljahi, and Fred Hollowood. 2013. DCU-Symantec at the WMT 2013 quality estimation shared task. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 392–397, Sofia, Bulgaria. Association for Computational Linguistics.

Hong Ka San. 2009. Chinese-English code-switching in blogs by Macao young people. Master's thesis, The University of Edinburgh, Edinburgh, UK. `http://hdl.handle.net/1842/3626`.

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. `http://www.scholink.org/ojs/index.php/selt`.

Thamar Solorio and Yang Liu. 2008a. Learning to predict code-switching points. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 973–981. Association for Computational Linguistics.

Thamar Solorio and Yang Liu. 2008b. 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. Association for Computational Linguistics.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steve Bethard, Mona Diab, Mahmoud Gonheim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirshberg, Alison Chang, and Pascale Fung. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of the First Workshop on Computational Approaches to Code-Switching. EMNLP 2014, Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar. Association for Computational Linguistics.

Joachim Wagner, Piyush Arora, Santiago Cortes, Utsab Barman, Dasha Bogdanova, Jennifer Foster, and Lamia Tounsi. 2014. DCU: Aspect-based polarity classification for SemEval task 4. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval-2014)*, pages 392–397, Dublin, Ireland. Association for Computational Linguistics.

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 (SLTU'12)*, Cape Town, South Africa. International Research Center MICA.

Hiroshi Yamaguchi and Kumiko Tanaka-Ishii. 2012. Text segmentation by language using minimum description length. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 969–978. Association for Computational Linguistics.