Analyzing Roles of Classifiers and Code-Mixed factors for Sentiment Identification

Soumil Mandal¹ and Dipankar Das²

¹ Department of Computer Science & Engineering, SRM University, Chennai, India
² Department of Computer Science & Engineering, Jadavpur University, Kolkata, India
{soumil.mandal¹, dipankar.dipnil2005²}@gmail.com

Abstract. Multilingual speakers often switch between languages to express themselves on social communication platforms. Sometimes, the original script of the language is preserved, while using a common script for all the languages is quite popular as well due to convenience. On such occasions, multiple languages are being mixed with different rules of grammar, using the same script which makes it a challenging task for natural language processing even in case of accurate sentiment identification. In this paper, we report results of various experiments carried out on movie reviews dataset having this code-mixing property of two languages like English and Bengali, both typed in Roman script. We have tested various machine learning algorithms trained only on English features on our code-mixed data and have achieved a maximum accuracy of 59.00% using a Naïve Bayes (NB) model. We have also tested various models trained on code-mixed data, as well as English features and the highest accuracy of 72.50% was obtained using a Support Vector Machine (SVM) model. Finally, we have analyzed the misclassified snippets and have discussed the challenges needed to be resolved for better accuracy.

Keywords: Code-Mixing, Sentiment Classification, Bilingual Sentiment Analysis, English-Bengali Code-Mixing

1 Introduction

English is by far the most popular language in the Web 2.0 but on Social Media, its dominance is receding. An automated language detection algorithm was applied to over 62 million tweets to identify the top 10 most popular languages on Twitter [1]. It was found that about half of the tweets were in English while the other half were in other languages. It is also a popular trend, and growing with the rise in middle class in several countries to mix multiple languages (code-mixing) for expressing their thoughts on Social Media.

Users whose first language belongs to non-Roman alphabets mix the Roman alphabet for convenience. Such usage increases the likelihood of code-mixing with a language of Roman alphabets. This case is quite clearly observed in South Asia and especially in the Indian subcontinent. Majority of schools in the urban areas of India uses English as the primary language of teaching and communication and leads to
rapid increase in code-mixed data that isn’t being utilized to its potential due to lack of resources and systems which can deal this effectively. For our experiments, we used data comprising of two languages, English (the most popular language for international communication purposes, spoken by 5.52%\(^1\) of the world population as of 2010) and Bengali (the dominating language in the region of West Bengal and Bangladesh, spoken by 3.05%\(^1\) of the world population as of 2010).

Sentiment analysis which is also known as opinion mining is rapidly growing field in the world of Natural Language Processing. In cases such as these where the user mixes several languages, the task of sentiment analysis becomes harder and accuracy decreases rapidly. The task of sentiment classification of multilingual text has been attempted by various researchers before.

Most of the works have been carried out on data where the original script of the languages was used whereas there has been quite a few works exist in the same script. One commonly used method is to convert a whole document into a single language and then polarity is determined [2]. This method is not quite accurate due to the fact that machine translation in itself is a big challenge and many a times several classes of information is lost in the process. A method where classifiers trained on the languages has been explored as well [3]. More complex methods have been tested like language identification followed by POS tagging and finally polarity identification [4]. This process is relatively ineffective in cases like linguistic code switching where loss of context is a big issue. Experiments on cross-lingual sentiment analysis have been tried as well [5]. On the other hand, a language independent model, relying only on emoticons which outperformed a Naïve Bayes model trained on bag of words is described in [6]. A method which does not rely only on emoticons, but also character and punctuation repetitions and consider language independent features is mentioned in [7]. To the best of our knowledge, there hasn’t been any work on sentiment identification of English-Bengali code-mixed data yet where both the languages are in Roman Script. Many challenges can be seen in this scenario, the most common ones are present in grammatical structure as well as ambiguity.

In our paper, we aim to see how different supervised machine learning algorithms trained on English features perform on English-Bengali code-mixed data as well as improve the accuracy of the same by including code-mixed features in the training set. Both the English and code-mix data are on the same topic in our case, which is movie/film reviews. On a whole, we have performed three experiments. In the first experiment, we have trained our classifiers on English features and have tested the same on English movie reviews dataset where LSVC (SVM with linear kernel) obtained the best accuracy with a score of 84.45%. For the second experiment, we have again used the previously trained classifiers and have tested them on our code-mixed dataset. Here, MNB (Multinomial Naïve Bayes) was the winner with an accuracy of 59.00%. Finally, in our third experiment, we have extracted features from our code-mixed training dataset and have tested them on code-mix data and again LSVC performed the best with an accuracy of 72.50%. Lastly, important evaluating parameters of the best performing classifiers from each experiment were calculated and the mis-

\(^1\) https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers
classified snippets were analyzed. We have also discussed the possible steps needed to be taken to improve accuracy in future.

The paper is organized as follows: In Section 2, we have described the datasets used for experimentation. In Section 3, we have described the machine learning algorithms used for our work and in Section 4 we have discussed the features used to train these algorithms. The experimental setup is described in Section 5 along with results obtained. In Section 6, we have analyzed the misclassified snippets and have discussed the probable reasons. Finally, in Section 7, we concluded the paper and have discussed the work needed to be done in the future for better results.

2 Data

On a whole, two datasets were used for conducting our experiments. One was in English while the other one was in English-Bengali Code-Mix.

2.1 English Data

The Cornell polarity dataset v1.0\(^2\) consisting of movie reviews has been used in our present experiments. It contains 5331 positive snippets and 5331 negative snippets. Training dataset was made by randomly picking 4000 snippets from each of the two sets. The test dataset contains randomly picked 1200 snippets from the 1331 remaining snippets from each of the sets. The two datasets had no snippet in common.

| Data     | Positive | Negative | Total |
|----------|----------|----------|-------|
| Training | 4000     | 4000     | 8000  |
| Testing  | 1200     | 1200     | 2400  |

Table 1. English data description based on usage.

Examples

1. A real movie, about real people, that gives us a rare glimpse into a culture most of us don’t know.
2. Intriguing and beautiful film, but those of you who read the book are likely to be disappointed.

2.2 Code-Mix Data

Our Code-Mix data comprises of two languages, namely English and Bengali. The data was collected using the Twitter API\(^3\) and the Facebook Graph API\(^4\). We collected data from June 15, 2017 to Dec 1, 2017 (duration of 5 months). Reviews were hand-picked based on relevance to the topic (movie/film/show). A total of 800 positive and 800 negative snippets were collected. For training purpose, 600 positive and 600 neg-

\(^2\) https://www.cs.cornell.edu/people/pabo/movie-review-data/

\(^3\) https://dev.twitter.com/overview/api

\(^4\) https://developers.facebook.com/docs/graph-api
ative snippets were selected using a random function. For testing, the remaining 200 positive and 200 negative snippets were used. The data used for training and test had no snippet in common. Our Code-Mix data is relatively quite small due to the fact that it’s extremely tedious and time consuming effort to collect snippets consisting of solely English and Bengali (Roman Script) words since to the best of our knowledge, none of the mentioned APIs provide such facility.

```
| Data     | Positive | Negative | Total |
|----------|----------|----------|-------|
| Testing  | 600      | 600      | 1200  |
| Training | 200      | 200      | 400   |
```

Table 2. Code-Mix data description based on usage.

**Examples**

1. *(Etotoi kharap je)Bengali (critics)English (der khub koshto kore dekhte hoyeche)Bengali (film)English (taah)Bengali .

   Translation: So pathetic that the critics had a tough time watching the film.

2. *(Bondhuder sathe dekhar jonho ekdom thik thak)Bengali (movie)English (taah)Bengali .

   *(Shei)Bengali (school days)English (er kotha mone pore gelo)Bengali .

   Translation: The movie is ideal for watching with friends, reminds me of my school days.

3. *(Proshongsha chara r kichui nei amar mukhe)Bengali , (Bhaggish)Bengali (first day first show)English (tei gechilam nahole ekgada)Bengali (spoilers)English (shunte hoto)Bengali .

   Translation: I have nothing but appreciation, fortunately watched first day first show otherwise would have had to hear a lot of spoilers.

### 3 Machine Learning Classifiers Used

In our present set up, we have not implemented any unsupervised system because the identification of sentiment using unsupervised methods has not produced satisfactory results [17]. Moreover, to deal with special code-mix features, we have employed several supervised classifiers.

#### 3.1 Naïve Bayes (NB)

These methods are a set of supervised learning algorithms based on Bayes Theorem with the naïve assumption of independence between every pair of features. The different Naïve Bayes classifiers differ mainly by the assumptions they make regarding Prob(X/Y). For our experiments, we have used *Gaussian Naïve Bayes (GNB), Bernoulli Naïve Bayes (BNB)* and *Multinomial Naïve Bayes (MNB)* Classifiers which are present under Naïve Bayes classifiers in the *scikit learn* package. The parameters of the modules were set to default values.

5 [http://scikit-learn.org/stable/modules/naive_bayes.html](http://scikit-learn.org/stable/modules/naive_bayes.html)
3.2 Linear Model (LM)

Linear Models describe a continuous response variable as a function of one or more predictor variables. We have used two such models for our experiments. The first one is Logistic Regression (LRC) whereas the second classifier is called Stochastic Gradient Descent (SGDC) which is based on “partial fit” method. The Linear Models present in the scikit learn package was used by tuning the parameters to default values.

3.3 Support Vector Machine (SVM)

Support Vector Machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. They are large-margin rather than probabilistic classifiers in contrast to Naïve Bayes and Linear Models. They are found to be highly effective for text classification and generally outperform Naïve Bayes models. For our experiments, we have used the Linear Support Vector Machine (LSVC) which is based on “one vs the rest” and NuSVC which is based on “one against one”. We have used the Support Vector Machines under scikit learn for our experiments. The parameters of the modules were set to default values.

4 Features

4.1 Part Of Speech (POS)

A Part of Speech tagger is a piece of software that reads text and assigns part of speech tag to each word based on context. We have used the NLTK POS tagger for our experiments. Additionally, we have conducted the preprocessing step by removing the punctuation marks using NLTK regex tokenizer.

4.2 N-Grams

N-Gram refers to contiguous sequence of n items from a given sequence of text or speech. For our experiments, we have used unigrams, bigrams and trigrams (n = 1 to 3). We have generated the n-grams with the help of NLTK n-gram module.

Sentence Example: \(\text{(Movie)English (ta khub bhalo)Bengali} \). 
Translation: The movie is very good.

unigrams – \{\{Movie\}, \{ta\}, \{khub\}, \{bhalo\}\}, bigrams – \{\{Movie ta\}, \{ta khub\}, \{khub bhalo\}\}, trigrams – \{\{Movie ta khub\}, \{ta khub bhalo\}\} 

---

6 http://scikit-learn.org/stable/modules/linear_model.html
7 http://scikit-learn.org/stable/modules/svm.html
8 http://www.nltk.org/book/ch05.html
9 http://www.nltk.org/_modules/nltk/tokenize/regexp.html
10 http://www.nltk.org/_modules/nltk/model/ngram.html
4.3 SentiWordNet 3.0

A word appearing in the SentiWordNet [8] generally contains emotion. For word level emotion classification, it is necessary to disambiguate emotion and non-emotion words properly. This feature helps the classifier to clearly define emotion and non-emotion words. We have used SentiWordNet 3.0\(^{11}\) present in the NLTK package for our experiments.

4.4 SO-CAL

SO-Calculator\(^{12}\) [9] is an application used for calculating Semantic Orientation of text documents. It was mainly designed for online product reviews. It has a total of five dictionaries, namely adjective, adverb, noun, verb and intensification.

4.5 NRC Emotion Lexicon

The NRC Emotion Lexicon\(^{13}\) [10] is a list of English words (about 14,000) and their association with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy and disgust) and two sentiments (negative and positive).

(Note: Feature *presence* as opposed to feature *count* was used which speeds up the overall processing)

5 Experimental Setup and Result Analysis

5.1 English Data

*Exp1*: For our unigram model, we have done experiment using adjectives only (*fe01*) as well as adjectives, nouns, adverbs and verbs (*fe02*). After extraction of n-grams, we discarded bigrams with ≥1 stop-words and trigrams with ≥2 stop-words. The top 1000\(^{14}\) bigrams (*fe03*) and top 500\(^{14}\) trigrams (*fe04*) were chosen based on frequency. They were used in a *bag of words* fashion. The lexicons used for extracting features from the training set were SentiWordNet (*fe05*), SOCAL (*fe06*), and NRC emotion lexicon (*fe07*). For NRC emotion lexicon, words without a polarity (i.e positive 0, negative 0) weren’t used. Features used for training are described in FSet1. Results are shown in Table 3.

---

\(^{11}\) http://sentiwordnet.isti.cnr.it/

\(^{12}\) http://www.sfu.ca/~mtaboada/research/nserc-project.html

\(^{13}\) http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm

\(^{14}\) The number of bigrams and trigrams to include as features was done empirically
Feature set acronyms (FSet1):

- **fe01** – unigrams: all adjectives,
- **fe02** – unigrams: all adjectives, nouns, adverbs, verbs
- **fe03** – fe02 + top 1000 bigrams
- **fe04** – fe03 + top 500 trigrams
- **fe05** – fe04 + SWN
- **fe06** – fe05 + SOCAL
- **fe07** – fe06 + NRC emotion lexicon

### Summary

From Table 3, we can clearly see a jump in the accuracy from **fe01** (max acc. 72.80%) to **fe02** (max acc. 78.16%). Introducing bigrams **fe03** increased the accuracy for all the classifiers (average 1.65%) and introducing trigrams **fe04** increased the accuracy as well (average 0.37%), though not as much as bigrams. Introduction of **SentiWordNet** (**fe05**) again shows a significant improvement in the accuracy (average 2.49%). There on introduction of **SOCAL** (**fe06**) and **NRC emotion lexicon** (**fe07**) showed little improvement in the accuracy.

### Classifiers

It is clear from Table 3 that **Support Vector Machines** (**LSVC and NuSVC**) performed better than the rest [11]. Among these two, **LSVC** (confusion matrix shown in Table 7) got the edge with an accuracy of 84.45% whereas **NuSVC** got 84.37%. In **Linear Models**, **LRC** performed slightly better with an accuracy of 80.08% while **SGDC** performed significantly poorer with an accuracy of 76.75%. **Nave Bayes** models performed comparatively well with **MNB** getting the highest accuracy of 80.58%, **BNB** with an accuracy of 79.50% and **GNB** got the overall least accuracy with a score of 71.91%. For our case (English Train – English Test), based on performance, we can infer that **SVM > NB > LM**.

#### 5.2 Code-Mix Data

On Code-Mix testing dataset, we have done two experiments, namely **English Train – Code-Mix Test** (**Exp2**) and **Code-Mix Train – Code-Mix Test** (**Exp3**).
Exp2: Here we have ran the same classifiers used in Exp1 i.e classifiers trained on English based on fe01, fe02, fe03, fe04, fe05, fe06 and fe07 (FSet1) features and tested on Code-Mix data. The performances of the different classifiers are shown in Table 4.

| Feature Combinations | Accuracy in %          |
|----------------------|------------------------|
|                      | NB | LM | SVM |
| uni (adj)            | 45.25 | 50.75 | 51.50 |
| uni (adj + adv + vrb + nou) | 50.00 | 53.75 | 54.50 |
| uni + bi             | 50.75 | 54.75 | 55.25 |
| uni + bi + tri + SWN | 51.00 | 55.25 | 55.50 |
| uni + bi + tri + SWN + SOCAL | 54.25 | 57.75 | 58.25 |
| uni + bi + tri + SWN + SOCAL + NRC | 54.50 | 58.25 | 59.00 |

Table 4. Accuracies of classifiers (English Train – Code-Mix Test). Boldface: Best performance shown in the experiment. Underline: Decrease in accuracy with inclusion of bigrams and trigrams.

Summary: In Exp2, we can see a significant downfall in the accuracies of the classifiers as compared to Exp1 (from 84.45% to 59.00%). In this experiment, we can again see a significant rise in the accuracies after adding adverbs, verbs and nouns along with adjectives (average 3.67%). Introduction of bigrams (fe03) shows very small improvement (average 0.35%) and even smaller (average 0.07%) or no improvement (LSVC) in case of trigrams (fe04). For Linear Models (i.e LRC and SGDC), we can see a drop in the accuracy after introduction of bigrams and trigrams [12]. Introduction of SentiWordNet again shows a bit improvement in the accuracy (average 2.14%). Adding SOCAL (fe06) and NRC emotion lexicon (fe07) doesn’t show much improvement.

Classifiers: We can see from Table 4 that Naïve Bayes (MNB and BNB) performed better than the rest. Among these two, MNB (confusion matrix shown in Table 8) performed better with an accuracy of 59.00% whereas BNB got 58.25%. For Linear Models, both LRC and SGDC had varying changes in the accuracy as different features were introduced but at the end, for fe07, both of them got the same accuracy of 55.00%. Among Support Vector Machines, NuSVC performed a bit better with an accuracy of 57.25% while LSVC got 56.75%. In this case (English Train – Code-Mix Test), based on performance, we can infer that NB > SVM > LM.

Exp3: Here, we did one more preprocessing steps 1) removal of emoticons and hashtags. Next, we collected top 1000 unigrams (fe08) based on frequency. Unigrams which were either English stop-words were not chosen. Also, Bengali unigrams

---

15 Negative impact on accuracy of LM and SVM classifiers while little impact on NB classifiers
16 Out of scope (also to keep parity with English data)
(e.g., er, e, je)\textsuperscript{17} which doesn’t carry any sentiment were ignored as well. After extracting bigrams, bigrams containing ≥1 English stop words or Bengali non-sentiment\textsuperscript{17} words were removed [13]. For trigrams, similar experiment was done except trigrams containing ≥2 English stop words or Bengali non-sentiment\textsuperscript{17} words were removed [13]. For bigrams (fe09), top 200\textsuperscript{14} were chosen and trigrams top 100\textsuperscript{16} (fe10) were chosen based on frequency. They were used in a bag of words fashion. Lexicons used for feature extraction from Code-Mix training set were SentiWordNet (fe11), SOCAL (fe12) and NRC emotion lexicon (fe13). For NRC emotion lexicon, words without a polarity (i.e positive 0, negative 0) weren’t used. Features used for training are described in FSet2.

Feature set acronyms (FSet2):

- fe08 – unigrams top 1000
- fe09 – fe08 + bigrams top 200
- fe10 – fe09 + trigrams top 100
- fe11 – fe10 + SWN
- fe12 – fe11 + SOCAL
- fe13 – fe12 + NRC emotion lexicon

| Feature Combinations          | Accuracy in % |       |       |       |       |       |
|------------------------------|---------------|-------|-------|-------|-------|-------|
|                              | NB            | LM    | SVM   |       |       |       |
|                              | GNB           | BNB   | MNB   | LRC   | SGDC  | LSVC  | NuSVC |
| uni                          | 51.50         | 53.50 | 56.00 | 55.25 | 53.75 | 59.00 | 58.75 |
| uni + bi                     | 52.50         | 54.25 | 56.75 | 56.00 | 54.75 | 60.25 | 60.00 |
| uni + bi + tri               | 52.75         | 54.50 | 57.25 | 56.25 | 54.75 | 61.00 | 60.75 |
| uni + bi + tri + SWN         | 58.25         | 64.75 | 66.50 | 66.00 | 63.50 | 72.25 | 71.50 |
| uni + bi + tri + SWN + SOCAL | 58.75         | 65.50 | 67.00 | 66.25 | 64.25 | 72.50 | 71.75 |
| uni + bi + tri + SWN + SOCAL + NRC | 58.75      | 65.75 | 67.75 | 66.75 | 64.75 | 72.50 | 72.00 |

Table 5. Accuracies of classifiers (Code-Mix Train – Code-Mix Test). Boldface: Best performance shown in the experiment. Underline: Significant increase in accuracy with inclusion of SWN.

Summary: For unigrams (fe08), we can see quite a bit rise (average 2.14%) in the accuracy as compared to unigrams (fe02) shown in Table 5. Introduction of bigrams (fe09) and trigrams (fe10) doesn’t improve the accuracy much. Interestingly, introduction of SentiWordNet (fe11) shows a big improvement (average 9.35%) in the accuracy for all the classifiers. This is due to the fact that our Code-Mix dataset contained a lot of sentiment carrying English words. Also, this improvement is not shown in Table 4 in case of fe05 because fe05 only contains English features while fe11 contains both English as well as Bengali features (fe08, fe09, fe10). Again, like Exp1 and

\textsuperscript{17}A list Bengali words which won’t contribute to any polarity was made manually from the code-mixed training dataset.
Exp2, introducing SOCAL (fe12) and NRC emotion lexicon (fe13) shows slight improvement.

Classifiers: It can be seen from Table 7 that Support Vector Machine performed better than the rest by a big margin. Among SVMs, LSVC (confusion matrix shown in Table 9) got the better score with an accuracy of 72.50% (rise in 13.5% accuracy from Exp2) while NuSVC got 72.00%. LRC scored better than SGDC in Linear Models with an accuracy of 66.75% while SGDC got 64.75%. Naïve Bayes didn’t disappoint much either, MNB got the highest with an accuracy of 67.75% followed by BNB with an accuracy of 65.75% and trailed by GNB with an accuracy of 58.75%. In this experiment, we can see a similar performance pattern with Exp1. Here based on performance we can infer that SVM > NB > LM.

6 Error Analysis

In this section, we discuss the drawbacks of our systems. We have done error analysis for results on both English data as well as Code-Mix data. Analysis on Code-Mix data is done more extensively as compared to English data since English data analysis has been covered by a lot researchers before. The best performing systems are evaluated with the help of confusion matrix along with important parameters like Accuracy, Precision, Recall, F1-Score, G-Measure and Matthews Correlation Coefficient (MCC). The parameters calculated along with the formula used is shown in Table 6.

English Train – English Test: LSVC performed the best with an accuracy of 84.45% (shown in Table 3). Othe parameter values are shown in Table 7.

English Train – Code-Mix Test: MNB performed the best with an accuracy of 59.00% (shown in Table 4). Othe parameter values are shown in Table 8.

Code-Mix Train – Code-Mix Test: LSVC performed the best with an accuracy of 72.50% (shown in Table 5). Othe parameter values are shown in Table 9.

| Parameter | Formula |
|-----------|---------|
| Accuracy (A) | (TP + TN) / (P + N) |
| Precision (P) | TP / (TP + FP) |
| Recall (R) | TP / (TP + FN) |
| F1 Score | (2 * P * R) / (P + R) |
| G Measure | SQRT (P * R) |
| MCC | (TP * TN – FP * FN) / SQRT (TP + FP) * (TP + FN) * (TN + FP) * (TN + FN) |

Table 6. Parameters calculated along with formula.

6.1 English Train – English Test

LSVC (SVM based) performed the best on English Train – English Test with an accuracy of 84.45% which can be considered to be a satisfactory performance. NuSVC (SVM based) got an accuracy of 84.37% and got the second position.
### Table 7. LSVC trained on fe07 and tested on English testing dataset.

| Actual State | Predicted Negative | Predicted Positive |
|--------------|-------------------|--------------------|
| Negative (-ve) | TN: 1036           | FP: 164            |
| Positive (+ve) | FN: 209           | TP: 991            |

| Accuracy | Precision | Recall | F1 Score | G Measure | MCC |
|----------|-----------|--------|----------|-----------|-----|
| 84.45%   | 85.80%    | 82.58% | 84.16%   | 84.17%    | 5.79% |

Table 8. MNB trained on fe07 and tested on Code-Mix testing dataset.

| Actual State | Predicted Negative | Predicted Positive |
|--------------|-------------------|--------------------|
| Negative (-ve) | TN: 114           | FP: 86             |
| Positive (+ve) | FN: 78            | TP: 122            |

| Accuracy | Precision | Recall | F1 Score | G Measure | MCC |
|----------|-----------|--------|----------|-----------|-----|
| 59.00%   | 58.65%    | 61.00% | 59.80%   | 59.81%    | 2.57% |

6.2 English Train – Code-Mix Test

**MNB** (NB based) performed the best on English Train – Code-Mix Test. Confusion matrix along with values of some important parameters are shown in Table 2.

The challenges faced on Code-Mix data include old challenges known for English data as well as new challenges for Code-Mixing property. Since the entire training set was in English, classifiers were unable to identify Code-Mix words which are playing an important role in the sentence for imparting a sentiment (Sen 1). Majority of the misclassifications are due to this reason.

Sen 1. (Movie)**English** (*tar*)**Bengali** (*print quality*)**English** (*eto kharap je*)**Bengali** (*patience*)**English** (*chilo na purota dekhar*)**Bengali**. **clf - manual: neg, Exp2: pos**

Translation: The print quality was so bad that I didn’t have the patience to watch the whole movie.

Reason: Due to the fact that system couldn’t identify the word kharap. The word kharap, meaning bad is a pretty common word used in Bengali which was unidentified by the classifier. Also, the word patience is generally associated with positive sentiment.

Sen 2. (Finally)**English** (*bohudin por shobkichu bad diye tana ekta*)**Bengali** (*series*)**English** (*dekhte parlam*)**Bengali**. **clf - manual: pos, Exp2: neg**

Translation: Finally after a lot of days I could watch a series leaving everything aside.

Reason: Due to the fact that classifier thought the word bad to be an English word though in the given sentence, bad is used as a Bengali word meaning aside.
Sen 3. \textit{(Starting)}English (er diker)Bengali (portion)English (tah)Bengali (kemon jeno)Bengali (hollow)English (lagchilo)Bengali (but)English (shesher)Bengali (portion)English (ta asadharon)Bengali . (clf - manual: pos, Exp2: neg)

Translation: Starting portion kinda felt hollow but the ending was awesome.

Reason: Probably, due to the contribution of the English word hollow, the system couldn’t identify the word asadharon meaning awesome which has a stronger positive sentiment value compared to hollow which has a weaker negative sentiment value.

Sen 4. \textit{(Jodio)}Bengali (actors)English (der)Bengali (performance was satisfactory)English . (ami)Bengali (2/10)English (er besi debo na)Bengali . (clf - manual: neg, Exp2: pos)

Translation: Even though the actors performance was satisfactory, I wouldn’t rate it more than 2/10.

Reason: The error in classification of Sen 4 is not Code-Mix specific but we included it since a similar one was not found in the English dataset. Here the possible reason for classifying it as positive by the system is due to the word satisfactorily even though 2/10 is clearly a poor rating i.e negative (similar to Sen 3).

6.3 Code-Mix Train – Code-Mix Test

Like English Train – English Train, \textit{LSVC} (SVM based) performed the best on Code-Mix Train – Code-Mix Test. Examples of some snippets from Code-Mix testing data that were misclassified in \textit{Exp2} but correctly classified in \textit{Exp3}.

| Actual State | Predicted Negative | Predicted Positive |
|--------------|--------------------|--------------------|
| Negative (-ve) | TN: 141 | FP: 59 |
| Positive (+ve) | FN: 51 | TP: 149 |

Accuracy | Precision | Recall | F1 Score | G Measure | MCC
---------|-----------|--------|----------|-----------|--------
72.50%   | 71.63%    | 74.50% | 73.03%   | 73.05%    | 6.98%  |

Table 9. Results of \textit{LSVC} trained on \textit{fe13} and tested on Code-Mix testing dataset.

Sen 5. \textit{(Erokom ekta baje)}Bengali (cinema)English (korar por kono)Bengali (excuse)English (er)Bengali (public accept)English (korbe na)Bengali , (and)English (na korar e kotha)Bengali . (clf – manual: neg, Exp2: pos, Exp3: neg)

Translation: The public won’t accept any kind of excuse after making such a bad cinema, and they shouldn’t either.

Reason: The word baje means bad. Similar to bad in English, baje is a very common word used in Bengali to criticise something or to describe something as not good.

Sen 6. \textit{(Les Miserables)}Proper Noun (er plotter thekeo beshi bhalo laglo er)Bengali (drama)English (ar)Bengali (musicals)English (gulo)Bengali . (clf – manual: pos, Exp2: neg, Exp3: pos)
**Translation:** In Les Miserables I like the drama and musicals more than the plot.

**Reason:** The word bhalo means good. Similar to good in English, bhalo is a very common word used in Bengali to appreciate something. Examples of some snippets from Code-Mix testing data that were misclassified in Exp3 as well.

Sen 7. *(Jemon) Bengali (story) English (temn) Bengali (cast) English, (ghyam) Bengali (artwork) English, (Filmtar against bolar moton kicchu pelame na) Bengali.*

*Translation:* Like story like cast, awesome artwork. I have nothing to say against the film. *(clf–manual: pos, Exp3: neg)*

**Reason:** Small Code-Mix training data. The word ghyam meaning awesome or sometimes fantastic was not identified which clearly carries a positive sentiment. Small training set is a major issue for supervised classifiers.

Sen 8. *(Sherlock) Proper Noun (toh) Bengali (old times) Bengali (er motone ekhono ache) Bengali, (amar mote chilo na kharap) Bengali.* *(clf–manual: pos, Exp3: neg)*

*Translation:* Sherlock is same as the old times, according to me it wasn’t bad.

Sen 9. *(Starting) English (tah orokom shundor kore je ki hoye gelo) Bengali, (Ek-dome bhalo na) Bengali.* *(clf–manual: neg, Exp3: pos)*

*Translation:* With a starting as beautiful as that what happened later? ! Not at all good.

**Reason:** In both Sen 8 and Sen 9, the error is due to the property of negation which is quite commonly faced during sentiment analysis. Both the sentences use the negating word na which in the first sentence can be translated to wasn’t and as not in the second sentence. In both the translated sentences, the negating word, i.e wasn’t and not is following the sentiment carrying word, i.e bad and good respectively. This is the common pattern in English and most of the classifiers made in the past which take into account the property of negation is based on this idea. The interesting thing is that in Sen 8, the negating word na follows the sentiment carrying word kharap (English pattern) while in Sen 9, the negating word na is followed by the sentiment carrying word bhalo.

Sen 10. *(It seemed like director Box Office success) English (er jonho jeno teno prokare erokom ektu) Bengali (movie) English (te) Bengali (musicals) English (dhukiyechey) Bengali.* *(clf– manual: neg, Exp3: pos)*

*Translation:* It seemed like the director wanted to include musicals in such a movie by any means possible just to get a Box Office success.

**Reason:** System couldn’t identify the idiom jeno teno prokare which in the translated sentence is by any means, which clearly portrays a negative sentiment in the sentence. This Bengali idiom can be loosely matched with the English idiom by hook or by
crook. Some other errors that we noticed were related to context specific knowledge, domain specific knowledge, etc.

6.4 Others

Some interesting bigrams and trigrams were seen as well while experimentation. For example, in the bigram \((\text{money hol}o)\) in Bengali translation: thought, a POS tagger will tag the word money as noun referring to the currency but clearly it was meant to be something else. This can cause error specially if trained on a domain specific dataset where money leans towards a polarity. This can be seen in Sen 2 error, quite a few examples of such sort were found. For some words, though the root part was in English, it couldn’t be identified due to the the addition of suffixes like ta, e, je. Examples: time(ta), movie(ta), poor(e). A conventional stemmer doesn’t always work on such cases. If this process is done properly it can prove to be quite useful for sentiment extraction.

7 Conclusion & Future Work

In this paper, we have made an effort to perform binary sentiment analysis on English-Bengali Code-Mixed data using three types of supervised classifiers, Naïve Bayes (NB), Linear Models (LM) and Support Vector Machines (SVM). Classifiers trained on English features performed quite satisfactorily on English data (84.45%) but poorly on Code-Mixed data (59.00%). We used classifiers trained on Code-Mixed features as well and saw a big improvement in the accuracy (72.50%). Similar resources were used in both the cases, i.e n-grams and lexicons, thus showing that it is effective for Code-Mixed analysis as well. We can also conclude that a SVM tends to perform well when the train and test data are similar (Table 3, Table 5) while NB tends to perform better when the train and test data are different (Table 4). Also, including SentiWordNet in Exp3 gave a boost in the accuracy showing that users tend to use English words carrying sentiment quite often in spite of writing in Code-Mix. We have also analyzed misclassified data from both the experiments (Exp2, Exp3) and have found the probable reasons for it which can be fixed in the future for better results.

Our immediate goal is to collect more Code-Mix data with varied topics (e.g sports, politics, conversations, etc) and also from platforms other than Twitter and Facebook like blogs, websites, chat threads, etc. Collecting clean and noise free data is a challenge as well. By acquiring a large enough corpora with as much less noise as possible, it’ll be possible to build lexical tools for Code-Mix data (e.g Bengali SentiWordNet in Roman Script). Smaller NLP tools, for example negation dictionary, stop-word dictionary, intensifier dictionary, idiom dictionary, etc can also be made which can be quite useful for sentiment identification like it is in the case for English data. A method using language identification [14] followed by POS tagging [15] and then polarity detection can be tried as well. It’ll also be useful to explore in detail the different machine learning algorithms (specially neural nets and modified decision trees) on such data and try out different types of cascading [16] and ensemble techniques, both known ones and modified ones with different permutations and
combinations. Also, selection of features play a very important role in any supervised learning technique. Commonly used features like n-grams can be filtered with POS tags, stop words, etc. Features like Capitalization, Length of Sentence, Context Features, Quoted Portion, Emoticons, etc can be tried.

Reference

1. Hong, Lichan, Gregorio Convertino, and Ed H. Chi. "Language Matters In Twitter: A Large Scale Study." ICWSM. 2011.
2. Denecke, Kerstin. "Using sentiwordnet for multilingual sentiment analysis." Data Engineering Workshop, 2008. ICDEW 2008. IEEE 24th International Conference on. IEEE, 2008.
3. Narr, Sascha, Michael Hulfenhaus, and Sahin Albayrak. "Language-independent twitter sentiment analysis." Knowledge discovery and machine learning (KDML), LWA (2012): 12-14.
4. Tromp, Erik. Multilingual sentiment analysis on social media. Lap Lambert Academic Publ, 2012.
5. Brooke, Julian, Milan Tofiloski, and Maite Taboada. "Cross-Linguistic Sentiment Analysis: From English to Spanish." RANLP. 2009.
6. Davies, Alex, and Zoubin Ghahramani. "Language-independent Bayesian sentiment mining of Twitter." The 5th SNA-KDD Workshop ’11 (SNA-KDD’11). 2011.
7. Cui, Anqi, et al. "Emotion tokens: Bridging the gap among multilingual twitter sentiment analysis." Asia Information Retrieval Symposium. Springer Berlin Heidelberg, 2011.
8. Baccianella, Stefano, Andrea Esuli, and Fabrizio Sebastiani. "SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining." LREC. Vol. 10. 2010.
9. Taboada, Maite, et al. "Lexicon-based methods for sentiment analysis." Computational linguistics 37.2 (2011): 267-307.
10. Mohammad, Saif M., and Peter D. Turney. "Crowdsourcing a word–emotion association lexicon." Computational Intelligence 29.3 (2013): 436-465.
11. Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up?: sentiment classification using machine learning techniques." Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. Association for Computational Linguistics, 2002.
12. Tripathy, Abinash, Ankit Agrawal, and Santanu Kumar Rath. "Classification of sentiment reviews using n-gram machine learning approach." Expert Systems with Applications 57 (2016): 117-126.
13. Pak, Alexander, and Patrick Paroubek. "Twitter as a Corpus for Sentiment Analysis and Opinion Mining." LREC. Vol. 10. No. 2010. 2010.
14. Chanda, Arunavha, Dipankar Das, and Chandan Mazumdar. "Unraveling the English-Bengali Code-Mixing Phenomenon." EMNLP 2016 (2016): 80.
15. Ghosh, Souvik, Santanu Ghosh, and Dipankar Das. "Part-of-speech Tagging of Code-Mixed Social Media Text." EMNLP 2016 (2016): 90.
16. Boiy, Erik, and Marie-Francine Moens. "A machine learning approach to sentiment analysis in multilingual Web texts." Information retrieval 12.5 (2009): 526-558.
17. Das, Dipankar and Bandyopadhyay, Sivaji. “Word to Sentence Level Emotion Tagging for Bengali Blogs.” ACL-IJCNLP-2009, pp. 149-152.