Preparing Bengali-English Code-Mixed Corpus for Sentiment Analysis of Indian Languages

Soumil Mandal[1], Sainik Kumar Mahata[2], Dipankar Das[3]

[1]Department of Computer Science & Engineering, SRM University, Chennai
[2,3]Department of Computer Science & Engineering, Jadavpur University, Kolkata
{soumil.mandal, sainik.mahata, dipankar.dipnil2005}@gmail.com

Abstract

Analysis of informative contents and sentiments of social users has been attempted quite intensively in the recent past. Most of the systems are usable only for monolingual data and fails or gives poor results when used on data with code-mixing property. To gather attention and encourage researchers to work on this crisis, we prepared gold standard Bengali-English code-mixed data with language and polarity tag for sentiment analysis purposes. In this paper, we discuss the systems we prepared to collect and filter raw Twitter data. In order to reduce manual work while annotation, hybrid systems combining rule based and supervised models were developed for both language and sentiment tagging. The final corpus was annotated by a group of annotators following a few guidelines. The gold standard corpus thus obtained has impressive inter-annotator agreement obtained in terms of Kappa values. Various metrics like Code-Mixed Index (CMI), Code-Mixed Factor (CF) along with various aspects (language and emotion) also qualitatively polled the code-mixed and sentiment properties of the corpus.

Keywords: code-mixed, sentiment classification, language tagging, Twitter data, social media analysis

1. Introduction

India has a linguistically diverse and vast diaspora due to its long history of contact with foreigners. English, one of those borrowed languages, became an integral part of the Indian education system and has been recognized as one of the official languages as well, thus giving rise to a population where bilingualism is very common. This kind of language diversity coupled with various dialects instigates frequent code-mixing in India. This phenomenon has become even more transparent with the rise of social networking sites like Twitter and Facebook and also instant messaging services like WhatsApp etc. The writing style in such media indicates phonetic typing transliterated in Roman, generally mixed with English words through code-mixing and also Anglicism. Three facts are involved in this sort of code-mixing cases, 1. lack of knowledge in using appropriate native words, 2. typing convenience and 3. popularity of Roman script to cater to a large set of audience.

Social networking services has been gaining popularity very rapidly since their first appearance and has led to an exponential growth of minable data which is rich and informative. In developing countries where majority of the population are bilinguals, in social media data, we frequent observe a unique trend in typing where two or more languages are mixed for expression known as code-mixing. It is also observed that such code-mixed data are growing rapidly in WWW because multilingual users in social networks frequently share their sentiments and thus it becomes an important task to mine and analyze such data for gathering crucial informatics related to sentiment too. However, the complexity involved in mixing of multiple rules of grammars, scripts, use of transliteration in such code-mixed data possesses a big challenge for NLP tasks. Thus, it becomes an ever so important task to solve this problem since a huge chunk of the data on social media possesses this property and will be of great use if mined.

It has to be mentioned that the conventional methods devised for a single language inevitably fail or give poor results in such cases. Thus to bring more attention of researchers towards this important and challenging aspect, we developed code-mixed corpora for sentiment analysis in Indian languages. India is country with 255 million [1] multilingual speakers and one of our goals in this was to challenge the participants and researchers into building advanced and robust systems for sentiment analysis of such code-mixed data. In the present article, we describe the systems and strategies used for making the Bengali-English code-mixed resources. Bengali is an Indo-Aryan language of India where 8.10% of the total population are the first language speakers and is also the official language of Bangladesh. The original script in which Bengali is written by locals is the Eastern Nagari Script [2]. Majority of our collected data is from Twitter. The reasons why Twitter is an ideal source for collection of such data has been explained by (Pak and Paroubek, 2010). The contributions of our paper are as follows:

1. A method for collecting code-mixed data using filtering techniques to assure quality and reduce manual effort.
2. A fast and reliable language identification algorithm (accuracy = 81%) for code-mixed data with known target languages.
3. A sentiment classification system for code-mixed data using a hybrid system (accuracy = 80.97%) combining rule based and supervised models.
4. Gold standard Bengali-English code-mixed data with language and polarity tags.

[1] http://rajbhasha.nic.in/UI/pagecontent.aspx?pc=MzU=
[2] https://www.omniglot.com/writing/bengali.htm

5. Several useful polarity tagged lexicons like phrasal lexicon of length 1200, uni-gram lexicon of length 3000 consisting of phonetically transliterated Bengali words, English acronyms commonly used on social media and a list commonly used emoticons.
6. Also, a seed list of length 1500 for querying Twitter API for retrieving Bengali-English code-mixed data.

2. Related Work

Several automated systems for Twitter data collection have been made before for corpus collection targeting different aspects but none with the aim to collect code-mixed data as far as our knowledge. On the other hand, various language tagging models have been made recently for code-mixed data and quite a few where a common script has been used for both the languages and one of them is phonetically translated. Among these one of the most relevant works is by (Das and Gambäck, 2014). Here they demonstrated a system which uses modified character n-gram with weights combined with a lexicon based approach, minimum edit distance as well as context info. (Barman et al., 2014) used a hybrid system by combining a lexicon based module with supervised classifiers like SVM, CRF and decision trees. Some of them have also been made as a sub-part for a part-of-speech tagging system like the one by (Vyas et al., 2014). For sentiment analysis on code-mixed, binary polarity classification has been tried using different classes of supervised models by (Ghosh et al., 2017b) and for ternary polarity by (Ghosh et al., 2017a) and (Sharma et al., 2015). A comparative study of classifiers trained on different code-mixed features was done by (Mandal and Das, 2018). Sophisticated methods using sub-word LSTM for learning sentiments in noisy code-mixed data has been tested as well by (Joshi et al., 2016).

3. Code-mixed Corpus Development

Corpus collection was done in two steps by collecting raw data from Twitter followed by filtering and cleaning code-mixed data from raw data.

3.1. Raw Twitter Data Collection

Our primary aim was to collect quality Bengali-English code-mixed data. However, we observed several instances of phonetically transliterated Bengali utterances (written in Roman script) that do not convey the code-mixed property (Muysken, 2000). We were also eager to collect intra-sentential i.e code-switched data instead of inter-sentential since the former is much more common on social media and is relatively more challenging for polarity classification as compared to the latter. For collecting Twitter data, we used the public streaming Twitter API via the Twitter4j [3] using keywords for querying. The initial keyword list was prepared by considering commonly used positive and negative Bengali words (e.g., bhalo, kharap, baje) and their polarities were validated using Bengali SentiWordNet (Das and Bandyopadhyay, 2010). We collected a total of 600 code-mixed sentences manually from the initial search output. In order to overcome the saturation problem of the retrieved data with respect to a few query words, we made a validated Bengali keyword list of 1500 unique query words from 600 sentences in decreasing order of frequency.

3.2. Data Filtering & Cleaning

The collected raw Twitter data contained noise, mostly contributed by words from other languages than the required pair, partially or fully (e.g. *bahar* which is a commonly used Hindi word meaning "outside"), words or full texts not in Roman script, etc. Thus, it was very important to build and apply a filtering module for retaining relatively better quality data in order to reduce manual efforts. Moreover, in order to avoid the problem of duplicacy due to short interval of querying, we have considered two parameters for devising our filtration strategy. The first parameter is α which denotes the minimum number of Bengali tokens with respect to our seed list whereas β refers to the minimum length of a tweet. It was observed that, the coverage of the top frequent keywords from the seed list helped us to filter majority of our code-mixed instances from the raw data if we vary the values of α only in the range of 1 to 3 and β in between 4 to 6. However, in order to filter more code-mixed instances for fulfilling our requirement, we had to increase the value of α up to 5 and the β up to 8 to maintain the code-mixed property in our filtered tweets. The total amount of raw tweets collected was around 89k and the our filtering system filtered out about 10k tweets from it. The statistics are shown in the Table 1. Here N denotes the information of n^{th} settings using which the Twitter API was queried, filtered data denotes the number of data remaining after removal.

N	α	β	Keywords Spent	Filtered Data
1	2	4	150	3800
2	2	5	250	2500
3	3	6	300	1800
4	4	7	350	1500
5	5	8	450	900
sum			1500	$\approx$ 10500

Table 1: Filtering statistics with respect to α and β .

During the cleaning process, spams, incomplete tweets, ones with conflicting sentiments were removed manually. Sarcastic tweets were not removed as it has become a very common tool for expression in the 21st century, especially on social media and thus it is important to classify them properly using more advanced techniques. URLs and Hashtags were kept as well as they too are important for sentiment analysis [4] (e.g. visiting the URL for analysis). We wanted to keep the data as untouched as possible to urge the future researchers to build highly robust systems which can be directly used on social media contents without much modification. Table 2 show the retrieved, filtered and used code-mixed data counts. It can be seen

[3] http://twitter4j.org/en/

[4] https://open.blockspring.com/bs/sentiment-analysis-from-url-with-alchemyapi

that our filtering system filtered out quite a lot of data and retained only about 11.79%.

Type	R	Count
Retrieved Tweets (RT)	≈	89000
Filtered Tweets (FT)	≈	10500
Code-Mixed Tweets (CT)	≈	5000

Table 2: Tweets retrieved statistics.

Some examples from our collected data after filtering are given below (underlined - EN, normal - BN) -

1. Thik fairy tale er ending tar moton amra shobaio happily ever after thakte lagilam. (Trans: *Just like a fairy tale ending we also lived happily ever after.*)
2. Script ta khub tiring chilo amar mote, aro onek better hote parto. (Trans: *The script was very tiring according to me, could have been much better.*)

4. Annotation

In order to annotate the language and sentiment tags to the filtered and cleaned tweets, we developed a system that help in basic annotation. One of the motivations of our annotation task was to reduce the manual tagging effort as we had to deal with huge amount of tweets ~10K. Therefore, in order to cope up with the problems of manual annotations, we planned to build two basic annotation systems, one is for language tagging and another is for sentiment tagging. Both of the annotation systems are described in subsection 4.1.. Finally, the outputs of these systems were evaluated by two sets of annotators, one set (A) consisted of a single annotator from Computer Science background with Bengali as mother tongue, where as the second set (B) consisted of five experts and the final evaluation was done by them. In order to handle the confusion cases, an annotation guideline as discussed in subsection 4.2. was provided to the annotators prior tagging.

4.1. System based Annotation

Out of 10k filtered tweets given by the system, we manually selected a collection of 5k tweets (as all filtered tweets were not code-mixed) and then we fed it the language tagging and sentiment tagging systems.

4.1.1. Language Tagging System

For language tagging, we used a two-step modular approach by combining lexicon based module (LBM) along with a supervised learning module (SLM).

LBM: As our target was simple, that is only to tag Bengali (BN) or English (EN) at word level, we tried to develop a relatively simple system. All the other unknown words are tagged as UN. The resources used to build the language tagging system are -

1. A list of Bengali words of size 3000 was prepared from the code-mixed data used in (Mandal and Das, 2018). Same words with different phonetic transliterations (e.g. *bhalo* and *balo*) both meaning good were also kept in the list.
2. *English Words (EW)* - A list containing 466k English words [5] was collected from online open sources.
3. *Suffix List (SL) & Acronym List (AL)* - An English suffix list [6] (e.g. *ing*, *ism*, *ious*) and an English acronym list [7] (e.g. *bbl*-be back later, *omg*- oh my god) was collected.
4. *N-Grams* - Bi-grams and tri-grams dictionary at character level was prepared from the above mentioned Bengali (BW) and English (EW) word lists, where keys were the n-grams and the respective values were frequency.

SLM: A supervised language tagger was developed by training the Linear Support Vector Machine (LSVC) implemented using scikit learn on two features which were character n-grams (n:2,3) as described in LBM features. For training, Bengali word list and list of most common English words [8] were used. The langauge tagging algorithm first searches the target token into our lexicons and if found, the appropriate tag is given. If not found, the supervised tagger is used to output the tag of that target token. The system was tested on ICON 2016 [9] POS tagging contest data and achieved a score of 86.24%.

4.1.2. Sentiment Tagging System

We used a hybrid system for sentiment classification. Similar to language tagging system, the sentiment tagging system also checks whether a tweet sentence is positive / negative / neutral using rule based method and if it fails, the supervised classifier is employed to produce the output sentiment tag. The resources which were prepared and used in the rule based were also used in supervised method as features.

Rule Based Method - For our rule based checking, three rules that were used to identify the sentiment of a tweet are as follows -

1. *Feeling (FLNG)* - A regular expression was used to extract the word that follows '– feeling' which is commonly used to express how the author feels. As such instances were self-tagged by the authors, there is no chance of ambiguity with respect to sentiment tagging. These tags are used since the stand alone texts may send different emotional signals or the author might simply be trying to convey his emotions directly.
2. *Hashtag (HT)* - Hashtags which used camel-casing or underscore separation were split and matched with lexicons and n-grams.
3. *Emoticon (EMO)* - Emoticons have a very strong impact on sentence level sentiment. We have used both

[5] https://github.com/dwyl/english-words

[6] https://www.learnthat.org/pages/view/suffix.html

[7] http://www.muller-godschalk.com/acronyms.html

[8] http://www.ef.com/english-resources/english-vocabulary/top-3000- words/

[9] http://ltrc.iiit.ac.in/icon2016/

Unicode and Icon representations of positive and negative emoticons for our experiments. Emoticon scoring has been experimented in three ways, e.g. higher frequency, greater index and average index. The second method which is based on the theory that the emoticon with the greatest index has the greatest influence on the tweet sentiment showed the best results.

Supervised Method - We have experimented with several supervised classifiers. In the Naïve Bayes (NB) family, we have used Gaussian Naïve Bayes (GNB), Bernoulli Naïve Bayes (BNB) and Multinomial Naïve Bayes (MNB). The Linear Models (LM) we have tested with are Linear Regression (LRC) and Stochastic Gradient Descent (SGDC). The scikit-learn [10] implementations of the models were used. The features used for supervised methods are as follows -

1. *Word N-Grams (WN)* - Word level uni-grams, bi-grams and trigrams were adopted as features. Each of the n-grams was sorted according to frequency in non-increasing order and the top 2000 n-grams were selected for training.
2. *Negation (NEGA)* - Negation in a message always reverses its sentiment orientation. If the number of negating words is odd, the polarity is reversed otherwise the calculated polarity is retained. Therefore, we collected a total of 25 English and 130 Bengali unique negation words.
3. *Tagged Words (TGW)* - We have also collected 1198 positive and 1802 negative Bengali uni-grams from an external code-mixed data available in (Mandal and Das, 2018). We combined them with English positive and negative words collected from NRC Emotion Lexicon and SOCAL lexicon to build a lexicon containing positive uni-grams (POSU) and negative uni-grams (NEGU).
4. *Tagged Phrases (TGP)* - In addition to words, we made a phrasal lexicon of length 1200 by extracting phrases ($\geq$ 1 from each sentence) from the code-mixed data described in (Mandal and Das, 2018). Such phrases are responsible to convey sentiment at the sentence level. For example, *boshe dekha jaye na* (trans: can't sit and watch), *onekei couldn't sleep* (trans: many couldn't sleep), etc. In case of tagged phrases, four scenarios were tested, *perfect match* - the phrase present in the sentence is identical to the tagged phrase, *sparse match* - all the unigrams of the tagged phrase are present in the sentence but not in the same order, *partial match* - a bigram from the tagged phrase (if $|phrase| \geq 2$) is present in the sentence in exact order (a bigram unit of stopwords is not considered) and finally, *no match* - none of the uni-grams is matched or the matched uni-gram is a stop-word.
5. *Tagged Acronyms (TA)* - Commonly used abbreviations on social networking sites were collected and polarity tagged as either positive or negative.
6. *SentiWordNet 3.0 (SWN)* - A word appeared in SentiWordNet (Baccianella et al., 2010) containing scores positive, negative and objective.
7. *SOCAL* - This lexicon is used for calculating semantic orientation (Taboada et al., 2011). For utilizing intensifiers of the lexicon, we used the logic that if both the intensifier and word is positive add their score, if both are negative add their scores and negate, if intensifier is positive and word is negative then subtract intensifier score from word score and finally if intensifier is negative and word is positive then add their score.
8. *NRC Emotion Lexicon* - a list of English words and their association with eight basic emotions and sentiment tags (Mohammad and Turney, 2013). In case of our classifier, we only utilized two polarity tags

For training our supervised classifiers, we used a manually tagged gold-standard dataset containing a total of 1500 training instances, i.e 500 of each polarity, created by merging data from (Mandal and Das, 2018) and (Ghosh et al., 2017b). In case of testing, we used a total of 600 tweets, i.e 200 of each polarity. The data (training and testing) had no data in common in the released versions. However, the features as mentioned for supervised learning were also used to train these classifiers. Different evaluation parameters scored by each of the classifiers are described in Table 3. Other than the *accuracy*, the mean value was considered over the three polarities for each of the other parameters. In Table 3, we can clearly find that SGDC achieved the best F1-Score with a value of 78.70. Thus, for building our polarity tagger, we finally used the trained model of SGDC. Paramaters (*Param*) were Accuracy (*Acc.*), Precision (*Prec.*), F1- Score (*F1*) and G-Score (*G*).

	Naïve Bayes (NB)			**Linear Model (LM)**	
Param	*GNB*	*BNB*	*MNB*	*SGDC*	*LRC*
Acc.	74.83	76.16	78.16	78.66	77.00
Prec.	75.05	76.25	78.56	79.20	77.40
Recall	74.83	76.16	78.16	78.66	77.00
F1	74.87	76.17	78.18	**78.70**	77.02
G	74.90	76.19	78.27	78.81	77.11

Table 3: Performance of different classifiers.

The confusion matrix of the best performing classifier, that is SGDC, is shown in Table 5. We can see that the classifier is quite stable and not very biased towards a single polarity. The best individual polarity accuracy is for neutral tweets (83%), which again supports the point regarding it's stability.

	pos	*neg*	*neu*
pos	161	12	27
neg	17	145	38
neu	13	21	166

Table 4: Confusion matrix of SGDC classifier (italics - predicted values, roman - true values).

The final algorithm we used for sentiment tagging by combining rule based and supervised into a hybrid routine

[10] http://scikit-learn.org/stable/

is described below -

```
Input ← sentence
Output → polarity

Step 1: if FLNG (sentence) $\neq$ neutral then
        return FLNG (sentence) else goto Step 2
Step 2: if EMO (sentence) $\neq$ neutral then
        return EMO (sentence) else goto Step 3
Step 3: if HT (sentence) $\neq$ neutral then
        return HT (sentence) else goto Step 4
Step 4: return SGDC (sentence)
```

Here FLNG, EMO and HT are the functions described under rule based methods in feature section and SGDC is our trained supervised classifier.

4.2. Annotators' Guidelines

As the data is already language and sentiment tagged by the systems, the manual annotation efforts were reduced drastically. However, in order to prepare a gold standard corpus with good quality, we finally handed it over to our annotators along with a number guidelines. We provided a very less number of guidelines as most of the urgent issues were already considered by using our systems.

Language Tagging - In case of language tagging, the scope of the current target word and the words preceding and succeeding the target word were considered.

Bengali (BN) & English (EN) Tag

LG1 The word is present in the respective language dictionary or is a slang or acronym of that language.
e.g. "*hall*" tagged as EN and "*ghor*" tagged as BN.

LG2 whether the word in context belongs to that respective language or not.
e.g. "*bar*" in "*onek bar bolechi*" is tagged as BN.

LG3 The word has any English/Bengali prefix or any English/Bengali suffix.
e.g. "*hall is*" tagged to EN and "*ghor ta*" tagged to BN.

Unknown (UN) Tag

LG4 The word does not belong to Bengali or English.
e.g. "*amr*" is tagged to UN.

LG5 The token is not recognized (like misspelled words).
e.g. "*ankushloveuall*" is tagged to UN.

LG6 The token is a special character, emoticon, URL, etc.
e.g. "@" is tagged as UN.

Sentiment Tagging - For polarity tagging, the authors' perspectives were taken into account and the emotions conveyed from the overall tweet were considered as well.

Positive Tag & Negative Tag

SG1 The tweet clearly expresses the sentiment towards the aspect term, for example a person, group or an object.
e.g. "*Sir, Boss 2 hit movie hobe. Eid ar sera movie.*" is tagged as positive.

SG2 The tweet clearly expresses the polarity state in mind of the author.
e.g. "*Dhurr ar posachhe na all these things.*" is tagged as negative.

SG3 The tweet clearly reports a polar sentiment or mood which may or may not be attributed directly by the author.
e.g. "*@username1 yes ami @username2 dadar pagol fan onek diner.*" is tagged as positive.

Neutral Tag

SG4 The tweet contains a mere observation or mention of an objective fact.
e.g. "*Dure oi yellow building ta holo shopping mall.*" is tagged as neutral.

SG5 It does not particularly convey any state of mind or opinion. A neutral sentiment is expressed towards the aspect term(s).
e.g. "*Cinema ta release koreche.*" is tagged as neutral.

Conflicts - The confusions occurred during annotation were tabulated as follows

English (EN) Tag

1. In the context of a word that contains numerical values were considered by the annotators. For example '11 AM' was tagged as EN by Annotator A while Annotator B tagged "11" as UN and "AM" as EN, separately.
2. Country names were tagged as EN and UN by Annotator A and B, respectively.
3. Universal words such as, "*table*" were tagged as EN by Annotator A and BN by Annotator B.
4. Words such as "*to*" were tagged as both EN and BN depending on the context and their phonetic representations.

Bengali (BN) Tag

1. The role of the suffix in a word was also dealt ambiguously. For example "*film*" is tagged as EN whereas "*film (ta)*" was tagged as BN.

Unknown (UN) Tag

1. Numerical values such as "1", "2" were tagged as UN.

We considered two sets of human annotators A and B along with system as the third set. The inter annotator agreement values or Cohen's Kappa (K) are shown in Table 5 with respect to each pairs of annotators. In case of sentiment tagging, the annotators agreed on majority of the tweets. However, in both language as well as sentiment tagging, the agreement scores between the sets of manual annotators were comparatively better than the agreements that were calculated with respect to systems. One of the reasons that degrades the system results is relatively small set of training instances in case of both language and sentiment tagging. The annotation details of the system and human annotators are shown in Table 6.

Language Tagging - Kappa	
Annotator A-System	0.69
Annotator B-System	0.65
Annotator A-Annotator B	0.83
Sentiment Tagging - Kappa	
Annotator A-System	0.83
Annotator B-System	0.82
Annotator A-Annotator B	0.94

Table 5: Inter annotator agreement.

Training Data			
	Language Tag		
	BN Tag	**EN Tag**	**UN Tag**
System	22801	15130	331
Annotator A	22460	15478	324
Annotator B	22471	15471	320
	Sentiment Tag		
	Pos Tag	**Neg Tag**	**Neu Tag**
System	988	926	586
Annotator A	1010	987	503
Annotator B	1000	1000	500
Testing Data			
	Language Tag		
	BN Tag	**EN Tag**	**UN Tag**
System	22896	12129	421
Annotator A	22418	12620	408
Annotator B	22416	12616	414
	Sentiment Tag		
	Pos Tag	**Neg Tag**	**Neu Tag**
System	1077	642	741
Annotator A	1094	698	668
Annotator B	1090	705	665

Table 6: Annotation details of system and human annotators.

5. Corpus Aspect Analysis

The released data distribution is shown in Table 7. In both training and testing, the quantity of neutral data is comparatively less as we found that most of the tweets we mined had a polarity. Here, we have analyzed different aspects of our developed gold standard data like code-mixing complexity and generic language aspects. Statistics on some of sentiment affecting aspects like polarity word count, emoticons count, etc were also carried out.

Distribution			
Purpose	**Positive**	**Negative**	**Neutral**
Training Data	1000	1000	500
Testing Data	1090	705	665

Table 7: Data distribution.

Language Aspects - Here we analyzed both complexity aspect contributed by code-mixing property (shown in Table 8) as well as other aspects like polarity token counts and mean length (shown in Table 9). Code-Mixing Index (CMI) introduced by (Das and Gambäck, 2014) indicates us the amount of code-mixing found in discourse. Another metric we have calculated which shows the complexity of multilingual corpus is the Complexity Factor (CF) proposed by (Ghosh et al., 2017b). CF takes into account three factors- language (LF), switching (SF) and mix (MF) factors. CF was calculated using all the three methods mentioned in that paper. From Table 8, we have observed that the collected code-mixed data has a higher code-mixing index as compared to FIRE 2015 [11] Shared Task Corpus (CMI = 11.65) and ICON 2015 [12] Shared Task Corpus (CMI = 5.73). Thus, we can conclude that our data is more complex from code-mixing point of view as compared to FIRE and ICON corpus. We can also see that on an average, positive data has higher code-mixing as compared to other polarities while neutral has comparatively lower code-mixing. From the training and testing values we can also see that the variance is quite nominal, thus adding to the quality of prepared corpus.

		Training			**Testing**		
index	**f**	**pos**	**neg**	**neu**	**pos**	**neg**	**neu**
CMI	min	4.02	4.24	4.20	4.16	4.20	4.18
	max	50.0	48.6	46.2	48.6	48.6	47.5
	mean	31.0	27.9	22.6	23.4	21.6	20.0
CF1	min	0.38	0.52	0.46	0.44	0.46	0.45
	max	20.8	18.0	23.0	37.5	23.0	37.5
	mean	4.20	3.93	3.71	4.14	3.67	4.17
CF2	min	4.58	4.81	4.76	4.63	4.76	4.72
	max	57.5	62.4	54.8	69.2	64.6	69.2
	mean	26.1	24.4	20.5	23.3	21.4	20.7
CF3	min	4.25	4.41	4.36	4.27	4.36	4.31
	max	53.8	58.4	51.8	68.0	61.5	68.0
	mean	24.2	22.6	19.1	21.6	19.9	19.3

Table 8: Complexity statistics (f - function).

Other important language related aspects are are shown in Table 9. The relation for negation count is $\geq$ as lexical checking was done so whereas there might be more number of negations. The aspect values were calculated based on post annotator tagging of language and sentiment. The probable reason for higher negation in negative data is mainly because of the habit of users to express negative sentiment by negating positive words, e.g. *bhalo na* which means "not good". This can be confirmed as well by skimming through the data. The table also tells us that users tend to write relatively more to the point and short tweets while expressing negative sentiments. This is checked from the mean length and UN word count values. Also, BN/EN ratio tells us that users tend to use more Bengali words for expressing objective sentiments.

[11] http://fire.irsi.res.in/fire/2015/home

[12] http://ltrc.iiit.ac.in/icon2015/

Language Aspects					
	Training Data				
N	Attribute	R	Pos	Neg	Neu
1	Negation Count	≥	148	449	170
2	Mean Length	=	18.50	18.06	17.91
3	BN word count	=	8541	8866	5064
4	EN word count	=	6997	6535	1939
5	UN word count	=	110	93	117
6	BN/EN Ratio	=	1.220	1.356	2.611
	Testing Data				
1	Negation Count	≥	182	375	200
2	Mean Length	=	18.94	16.23	17.46
3	BN word count	=	8664	7388	6364
4	EN word count	=	5985	4329	2302
5	UN word count	=	168	118	128
6	BN/EN Ratio	=	1.447	1.706	2.764

Table 9: Language statistics. (R - relation)

Emotion Aspects - Statistics of sentiment affecting aspects are shown in Table 10. Users tend to explicitly convey their feelings by using the feeling tag more so while expressing negative sentiment as compared to positive. For emoji count the relation is ≥ as lexical checking was done, so in reality there might be more number of emoticons. Same is the case for polarity word count, but here ≈ is used instead as contextually the word may not be positive or negative. From positive and negative word count in Table 10 we can see that users tend to use English polarity words more often as compared to Bengali while expressing.

Sentiment Aspects					
	Training Data				
N	Attribute	R	Pos	Neg	Neu
1	POS emoji count	≥	18	2	2
2	NEG emoji count	≥	3	17	1
3	POS word count	≈	1187/587	118/51	35/26
4	NEG word count	≈	103/65	757/416	32/19
5	Feeling tag count	=	5	10	1
	Testing Data				
1	POS emoji count	≥	22	5	3
2	NEG emoji count	≥	6	20	1
3	POS word count	≈	918/435	106/42	27/19
4	NEG word count	≈	119/72	673/341	28/13
5	Feeling tag count	=	4	8	2

Table 10: Sentiment affecting aspects. For POS, NEG word count representation format is EN/BN. (R - relation)

Other Aspects - The most common polarity carrying words from the code-mixed data are shown in Table 11. From the table we can see that the most common polar words are highly polar. These words are commonly used while speaking as well. It can also be seen that a lot of counterparts are present in the table, like bhalo - good, osadharon - special, kharap - bad, betha - pain, etc.

Most Common Words (freq>150)		
	Bengali	English
Positive	bhalo, besh, shundor, darun, moja, pochondo, osadharon	love, best, good, comedy, better, special, famous, happy
Negative	kharap, baje, kosto, boka, bekar, chinta, jhogra, betha	poor, bad, problem, old, sad, busy, bogus, pain

Table 11: Some common Bengali and English words, training and testing data combined.

6. System Performance on Final Data

After the final annotation was done we tested our systems again on the new gold-standard data. Both the language tagging system and sentiment tagging system (SGDC) was trained on the training data and evaluated on the testing data. The language tagger performed surprisingly well and got an accuracy of 81%. With the sentiment tagging system we expected a significant improvement due the increased size of the training data. It indeed performed better and got an accuracy of 80.97% and F1-Score of 81.2%. In future we would like to test different feature combinations and add contextual features as well to improve our system.

7. Release Format

The final gold-standard dataset is available in JSON format. We have chosen JSON since it is more compact, lightweight, flexible and easier to use compared to XML. CSV was ignored as well since we needed to represent a hierarchical structure which is much easier with JSON as well. Another problem with CSV is that a standard reader application (e.g. Excel) is quite slow at opening large files as well as unstructured encoded values and spilling. The objects/values provided in the released JSON file are id (data number), lang_tagged_text (language tagged text), sentiment (-1 ← negative, 0 ← neutral, 1 ← positive) and text (without language tag). A single sample from the JSON file is given below -

id: 83
lang_tagged_text: Onekdin\bn por\bn spotlight\en e\bn fire\bn eshe\bn nijeke\bn besh\bn bikheto\bn bikheto\bn lagche\bn ,\un I\en am\en toh\bn very\en hpy\en .\un
sentiment: 1
text: Onekdin por spotlight e fire eshe nijeke besh bikheto bikheto lagche, I am toh very happy.

8. Conclusion & Future Work

In this paper we have described the steps involved in building the system which we have used for collecting and preparing gold-standard Bengali-English code-mixed data for sentiment analysis. To the best of our knowledge, it is the first publicly released data of its kind. The data we present also has a reliable inter-annotator agreement, K - 0.83 for language tag and K - 0.94 for sentiment tag. We also discuss the challenges faced in each step which should be overcome in future for an improved system. In future, we wish to improve the quality of our system by increasing the population size of our resources and training our classifiers on bigger data. We also wish to find a correlation between α (BN token count) and the keyword used for querying to the API so that the value of α can be varied automatically using computationally calculated rules to fetch more relevant data which in this case was Bengali-English code-mixed.

References

Baccianella, S., Esuli, A., and Sebastiani, F. (2010). Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204.

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

Das, A. and Bandyopadhyay, S. (2010). Sentiwordnet for bangla. *Knowledge Sharing Event-4: Task*, 2.

Das, A. and Gambäck, B. (2014). Identifying languages at the word level in code-mixed indian social media text.

Ghosh, S., Ghosh, S., and Das, D. (2017a). Complexity metric for code-mixed social media text. *arXiv preprint arXiv:1707.01183*.

Ghosh, S., Ghosh, S., and Das, D. (2017b). Sentiment identification in code-mixed social media text. *arXiv preprint arXiv:1707.01184*.

Joshi, A., Prabhu, A., Shrivastava, M., and Varma, V. (2016). Towards sub-word level compositions for sentiment analysis of hindi-english code mixed text. In *COLING*, pages 2482–2491.

Mandal, S. and Das, D. (2018). Analyzing roles of classifiers and code-mixed factors for sentiment identification. *arXiv preprint arXiv:1801.02581*.

Mohammad, S. M. and Turney, P. D. (2013). Nrc emotion lexicon. Technical report, NRC Technical Report.

Muysken, P. (2000). *Bilingual speech: A typology of code-mixing*, volume 11. Cambridge University Press.

Pak, A. and Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10.

Sharma, S., Srinivas, P., and Balabantaray, R. C. (2015). Text normalization of code mix and sentiment analysis. In *Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on*, pages 1468–1473. IEEE.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., and Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307.

Vyas, Y., Gella, S., Sharma, J., Bali, K., and Choudhury, M. (2014). Pos tagging of english-hindi code-mixed social media content. In *EMNLP*, volume 14, pages 974–979.