International Journal of Advanced Science and Technology
Vol. 30, No. 1, (2021), pp. 01-11

Sentiment Analysis of Code-Mixed language

Sukhpreet Kaur
Research Scholar, Department of Computer Science, Punjabi University, Patiala, Punjab, India,
sidhu.sukhpreet12@gmail.com

Gurpreet Singh Josan
Associate Professor, Department of Computer Science, Punjabi University, Patiala, Punjab, India,
josangurpreet@pbi.ac.in

Abstract

Code-mixed language is very commonly used in today's multilingual society. It is the phenomenon of mixing the syntax and vocabulary of many languages in single sentence. Sentiment analysis of code-mixed language aims at identifying the polarity value of the sentence. This paper mainly focuses on sentiment analysis of Tweets consisting of words from Hindi and English language along with other symbols. The dataset contains 20,000 tweets. We generate word level, character level and subword level representation for the Tweets which are used as input to the different models such as CNN, LSTM and BiLSTM. The performance of BiLSTM model is better as compared with other models. The accuracy of WORD Level BiLSTM, BPE Level BiLSTM and WORD Level CHAR Level BiLSTM is 60.27%, 58.59% and 54.24% respectively.

Keywords: Sentiment analysis, Tweets, Polarity, Code-mixed language.

1. Introduction

Sentiment analysis is a branch of natural language processing. It has a variety of applications such as analyzing movie reviews, user modeling, curating online trends and tone of a text, speech and opinion mining. In literature, we find many names performing slightly different tasks, e.g. sentiment analysis, opinion extraction, opinion mining, sentiment mining, subjectivity analysis, affect analysis, emotion analysis, review mining, etc. However, they are all under the umbrella of sentiment analysis or opinion mining. Sentiment analysis from a formal text is well researched topic (Liu, 2012).

India is a multilingual country and the multilingual people; those are non-local English speakers use more than one languages to communicate within each other. The switching in between the languages is called code-switching or code-mixing, that depends on the type of mixing. Code mixing behavior at the word level is fairly common than the sentence level. In northern region of India, combination of English with Hindi (Hinglish) is more prevalent.
For example:

"Waglenikhil U saw caste and religion in them… nation saw talent and trust in them!!. Problem is tum paida hi ulte hue the!! "

In the above example some words are the English language and some words are the Hindi language however they all are written in Roman script.

ISSN: 2005-4238 IJAST

The social media texts like blogs, micro-blogs (e.g. Twitter) and chats (e.g. WhatsApp and Facebook messages) has made numerous of new opportunities for information access and language technology, but it has additionally presented many new challenges making it one of the current prime research areas. Although current language technologies are essentially worked for English, when the non-native English speakers use the social media they combine English with other languages.

Code-mixing represent several unseen difficulties to NLP tasks like word-level language identification, part-of-speech tagging, dependency parsing, machine translation and semantic processing. Sentiment analysis becomes more difficult in the situation when data is noisy and collected from the social media. Code-mixed text adopts the syntax and vocabulary of multiple languages. This becomes a challenge for sentiment analysis as traditional semantic analysis approaches do not capture the meaning of the sentences. Another challenge is the short abbreviated data present in the sentences. Same words can also be written in many forms in the sentence which is an another limitation. Pre-processing operations need to be performed to solve these challenges. This paper mainly focuses on pre-processing of tweet and to classify tweets into their corresponding sentiments-positive, negative or neutral.

2. Related Work

A lot of research work has been carried out in the area of sentiment analysis. **(Deshmukh, 2015)** presented the different levels of opinion analysis i.e. document level, sentence level, feature level, word level and phrase level. Data source for review collection and Approaches for sentiment classification. Most work has been done on product reviews downloaded from Amazon. **(G Remmiya Devi, 2016)** presented a task on Code Mix Entity Extraction for Indian Languages(CMEE-IL). Different methods are used for entity extraction from a code-mixed data. For feature extraction, trigrams are used. Evaluation of this model is carried out using SVM-light. **(Aditya Joshi, 2016)** introduced learning sub-word level representations in LSTM (Subword-LSTM) architecture rather than character-level or word-level representations. (**Nurfadhlina Mohd Sharef, 2016**) reviewed the state of the art of SA approaches, including sentiment polarity detection, SA features (explicit and implicit), sentiment classification through machine learning and applications of SA. A number of ML techniques have been adopted to perform the classification task in SA. **(Abdul Fatir Ansari, 2017)** attempted to conduct sentiment analysis on “tweets” using various different machine learning algorithms. **(Souvick Ghosh, 2017)** presented an approach that tackle code-mixed text from three different languages Bengali, Hindi, and Tamil - apart from English. Their system uses Conditional Random Field, a sequence learning method, which is useful to capture patterns of sequences containing code switching to tag each word with accurate part-of-speech information. (**Braja Gopal Patra, 2018**) presented the sentiment identification task from Hindi-English (HI-EN) and Bengali-English (BN-EN) code-mixed datasets. **(P. V. Veena, 2018)** presented that mixing of text is common in social media platform. For language processing, text classification and language identification is common. They used two word-based embedding features and character-based context features. **(Nurendra Choudhary, 2018)** proposed novel methodology called Sentiment Analysis of Code-Mixed Text (SACMT) to order sentences into their corresponding sentiment - positive, negative or

neutral, using contrastive learning. They use the mutual parameters of siamese networks to outline the sentences of code-mixed and standard languages to a typical sentiment space. **(Pruthwik Mishra, 2018)** proposed a sentiment Analysis for Indian languages (SAIL). Code Mixed tools aimed at identifying the sentence level sentiment polarity of the code- mixed dataset of Indian languages pairs (Hi-En, Ben-Hi-En). **(Yash Kumar Lal, 2019)** introduced code-mixing in which mixing the syntax and vocabulary of multiple languages in the same sentence. A hybrid architecture for the task of sentiment analysis of English- Hindi code-mixed data is presented. **(Ivan Provilkov, 2019)** introduced BPE-dropout method. BPE-dropout method is simple and effective sub-word regularization method based on and compatible with conventional BPE. This method outperforms both BPE and previous sub-word regularization on a wide range of translation tasks.

3. Dataset Details

The Dataset contains 20,000 code-mixed tweets along with their sentiments which is provided in a task on *SentiMix Hindi-English Competition* organized by Codalab (open-source web based platform). The training dataset consists of 17000 tweets and test dataset is of 3000 tweets. Each tweet from the training dataset starts with word *meta* and contain a unique *id* and *polarity/sentiment* value that describe whether the sentiment value is *positive*, *negative* or *neutral*. Every word in tweet is tagged with the corresponding language such as *Hin* for Hindi, *Eng* for English and *O* for other symbols.

The format of tweet in dataset is given as follows:

meta 1134680000000000000 neutral
RT Eng
@ O
UAAPconfessions Eng
Love Eng
looks Eng
good Eng
on Eng
Maddie Eng
!!! O
Ako Eng
lang Eng
ba Eng
yung Eng
sobrang Eng
masaya Hin
kasi Hin
may Hin
zolo Eng
sya Eng
? O
Before Eng

with Eng
the Eng
past Eng
Z Hin
medyo Eng
lowkey Eng
s Eng
‰Û_ O

We used 80% data from the given dataset for training and 20% for validation. Table 1 shows the number of tweets in training, validation and test data.

Table 1: Number of tweets in training, validation and test datasets

Language	Training	Validation	Test
Hin-Eng	13600	3400	3000

Table 2: Statistics of Training and Test dataset.

	Training Data	Test Data
Total Tweets	17000	3000
Total Words	443689	78380
Positive Polarity	5620	1023
Negative Polarity	4990	974
Neutral Polarity	6390	1003
No. of English Words	147028	26454
No. of Hindi Words	206899	36123
No. of Other Language Words	89762	15803
Average Length of a Tweet	26	26

4. Design and Implementation

In sentiment analysis, data pre-processing is a crucial step. Data pre-processing is performed to eliminate inconsistent data and prepare the data to perform downstream task. Before passing the dataset to different models for training, we need to perform some pre-processing operations to prepare the dataset for models.

4.1 Pre-processing of Dataset

- **Converting the raw dataset into required format**
  The dataset is first converted into a format as shown in Table 3. The tweet id, tweet, tweet having each word tagged with corresponding language and its sentiment value were extracted.

Table 3: Format of Dataset after loading

Tweet_id	Tweet	Lang_tag	Sentiment
1134680000000000000	RT @UAAPconfessions Love looks good on Maddie!!! Ako lang ba yung sobrang masaya kasi may zolo sya? Before with the past Z medyo lowkey s‰Û_	RT\Eng @\O UAAPconfessions\Eng Love\Eng looks\Eng good\Eng on\Eng Maddie\Eng !!!\O Ako\Eng lang\Eng ba\Eng yung\Eng sobrang\Eng masaya\Hin kasi\Hin may\Hin zolo\Eng sya\Eng ?\O Before\Eng with\Eng the\Eng past\Eng Z\Hin medyo\Eng lowkey\Eng s\Eng ‰Û_\O	Neutral
1131370000000000000	@imVkohli Best of luck @imVkohli sir World Cup ke liye bhot bhot subhkamnaye	@\O imVkohli\Eng Best\Eng of\Eng luck\Eng @\O imVkohli\Hin sir\Hin World\Eng Cup\Hin ke\Hin liye\Hin bhot\Hin bhot\Hin subhkamnaye\Hin	Positive

- **Converting tweet into lowercase**

  The reason to convert the whole tweet into lowercase is that the words *"HELLO, hEello, hello, hELLO"* have different word count for vocabulary size.

- **Replace repeating characters with maximum length of two characters**
  There are also some contiguous repeating characters in tweets that change the vocabulary size such as *"cittttty, ciiiity, ccccciiiitttyyyy, citttyyyy, ciitty"*. All of these words are different ways of representing the word "*city"*. Furthermore, the question arises why we choose to replace upto only two characters but not with one character, the reason is that if we replace with only one character, some words lost their meaning. For example: "GOOOOOOd, goood, good" if we replace with only one character then the word *good* is replace with the word *god* and the meaning of the sentence is totally lost. Pre-processing of the original tweet is shown in Table 4.

Table 4: Pre-processing of Tweet

Original Tweet	**Convert tweet into lowercase**	**Replace repeating characters with maximum length of two characters**
RT @DeepStateExpose @realDonaldTrump GRAB A COPY OF MY CRITICALLY ACCLAIMED #1 BEST SELLLER 'HISTORY OF THE DEEEP STATEEEEE'!!! Eboooook format iâ€¦!	*rt @deepstateexpose @realdonaldtrump grab a copy of my critically acclaimed #1 best selller 'history of the deeep stateeeee'!!! eboooook format iâ€¦!*	*rt @deepstateexpose @realdonaldtrump grab a copy of my critically acclaimed #1 best seller 'history of the deep statee'!!! ebook format iâ€¦!*

4.2 Feature Extraction

Following features were extracted for each tweet:

a) **Number of capital words**: Count the number of words that have only capital letters.
b) **Number of extended words**: Count the number of words with multiple contiguous repeating characters.
c) **Aggregate positive and negative sentiments:** SentiWordNet (Version 1.0) is a lexical resource in which each synset of WORDNET is tagged according to three numerical scores Obj(s), Pos(s) and Neg(s) and each of score ranges from 0.0 to 1.0 and their sum is 1.0 for each synset. In aggregate positive and negative sentiments, the polarity value of each word is evaluated that ranges from 0.0 to 1.0 using SentiWordNet and add that word in positive or negative word dictionary and atlast evaluate the net positive and net negative score of the tweet, sum of the score value of the tweet is 1.0.
d) **Number of repeated punctuation**: Count the number of sets of two or more contiguous punctuations.
e) **Exclamation at the end of sentence**: Check whether there is exclamation at the end of tweet or not and gives result in a boolean value.

For example:

"RT @ zarramalik super hilarious buttt reality of these NOORA HARAMKHOR FAMILY is verY differentttt!!!!".

In above tweet "*NOORA*", "*HARAMKHOR*" and "*FAMILY*" are the words that contain only capital letters. The number of capital words is 3 so the value of this feature is 1 in the list. In case if there is no capital word in tweet, the value of this feature will be 0. The words "*butt"* and *"differentttt"* contains contiguous repeating characters so the value of this feature is 1. In above tweet the sentiment polarity value of the tweet is net positive and value store in list as a 0.7. The repeated punctuation mark is exclamation mark and it is repeated more than two times so the value of this feature in the list is 1. There is exclamation mark at the end of the tweet so the value of this feature in the list is 1.

4.3 Tokenization

Tokenization is the process of splitting a text into a list of tokens. In a paragraph, a sentence is a token and in a sentence, a word is a token. For example, the text "This is a glass" can be tokenized into 'This', 'is', 'a', 'glass'. There are different methods and libraries available to perform tokenization. In this task, the Byte-Pair Encoding (BPE) tokenization is used which is a bottom up sub word tokenization technique.

5. Models

The different neural network models have been implemented and compared in the proposed research work. We have implemented Convolutional Neural Network (CNN), Long Short Term Memory (LSTM) and Bidirectional Long Short Term Memory (BiLSTM). Convolutional neural network models apply convolutions over the inputs to compute the outputs. In CNNs, each layer applies different filters over the data, and during this process, the model learns the values of its filters based on the task that we want to perform. In the end of this process, the results of each layer are combined. The architecture of CNN model is shown in Figure1(a). The unidirectional LSTM model preserves the information of the past because the only inputs it has seen are from the past. The bidirectional LSTM model connects hidden layers of opposite directions to the same output, so that the output layer can access information from past and future states. The architecture of BiLSTM model is shown in Figure1(b).

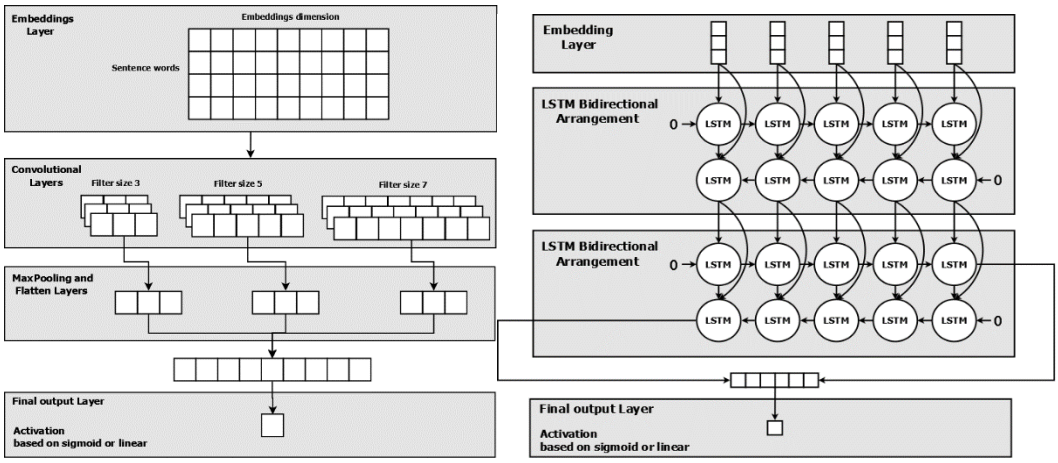

Figure 1: (a) Convolutional Neural Network Model Architecture, (b) Bidirectional LSTM Model Architecture.

Table 5 shows the implementation details for neural network models.

Table 5: Model-wise hyper-parameters

Model	Hyper-parameters	Value
CNN	Filters	128
	Kernel_size	3
	Optimizer	adamax
LSTM	#Units	256

	Recurrent_dropout	0.2
	Dropout	0.2
	Optimizer	adamax
BiLSTM	#Units	128
	Recurrent_dropout	0.2
	Dropout	0.2
	Optimizer	adamax

6. Results and Discussions

The evaluation methodology has components that are specific for sentiment analysis task dataset. In the experiments, the selected quantity that was intended to be analyzed was the validation loss. The validation loss is measured in each training epoch through a validation set that has been previously defined. In our case, the validation set that was defined as the last 20% of the training data, and the training procedure will stop when we observe at least two consecutive epochs with no improvements and total epochs is set to 25 and batch-size is finally set to 512 but also make experiments with batch-size of 128 and 256 and verbose is set to 1.

Table 6: Results of Models using training dataset

MODEL	ACCURACY (Without Attention mechanism)	ACCURACY (With Attention mechanism)
WORD Level LSTM	58.2%	59.90%
WORD Level CNN	56%	59.44%
CHAR Level LSTM	52%	43.80%
CHAR Level CNN	48%	52.64%
SUBWORD Level LSTM	55%	50.85%
SUBWORD Level CNN	53%	52.37%

After the results from the training dataset, we make experiment with attention mechanism and without attention mechanism. Attention mechanism is an improvement over the encoder/decoder mechanism.

In Table 6, we can see a slight increase in performance for WORD Level LSTM model, WORD Level CNN MODEL and CHAR Level CNN model, but with CHAR Level LSTM model, SUBWORD Level LSTM and SUBWORD Level CNN there is great loss in

performance by using the attention mechanism. With attention mechanism, WORD Level LSTM model is now the one with better performance.

With the attention mechanism, the performance of the CHAR Level LSTM model is decreased, for that we will apply the Bidirectional LSTM model and concatenate the WORD-CHAR levels. Instead of sub-word level representation, we use the byte pair encoding technique with the Bidirectional LSTM models.

At the end of the experiments using the test dataset, we use three best models i.e. WORD Level Bi-LSTM model, WORD-CHAR Level Bi-LSTM model AND Byte-Pair Encoding Level Bi-LSTM model.

Table 7: Results of Models using test dataset

MODEL	PRECISION	RECALL	F1-SCORE	ACCURACY
WORD Level BiLSTM	O.67	0.62	0.64	60.27%
BPE Level BiLSTM	0.64	0.63	0.63	58.59%
WORD-CHAR Level BiLSTM	0.53	0.62	0.57	54.24%

Table 7 shows the results of different models using the test dataset with attention mechanism. The accuracy of WORD-CHAR Level Bi-LSTM, Byte-Pair Encoding Level Bi-LSTM and WORD Level Bi-LSTM model is 54.24%, 58.59% and 60.27% respectively and the F1-score of WORD-CHAR Level Bi-LSTM, Byte-Pair Encoding Level Bi-LSTM and WORD Level Bi-LSTM model is 0.57, 0.63 and 0.64 respectively.

7. Conclusion and Future scope

In this paper, different neural network models with attention mechanism are trained to predict the sentiments of code-mixed Tweets. Extensive experiments are conducted on real world code-mixed social media dataset of Tweets. The WORD-BiLSTM model performs very well with limited dataset and achieves the accuracy of 60.27% and F1 score of 0.64.

In Future, many other experiments can also be done based on the given experimental evaluation. One of the possibility is that the models can be trained with large dataset to improve their accuracy. This task can be extended to some other type comments on social media sites and applications like Facebook, Instagram or Whatsapp etc.

References

[1]. Akshi Kumar, T. M. (2012). Sentiment Analysis: A Perspective on its Past, Present and Future. *Intelligent Systems and Applications, 2012, 10, 1-14*.

[2]. Aditya Joshi, P. A. (2016). Towards sub-word level compositions for sentiment analysis of hindi-english code mixed text. ArXiv , arXiv:1611.00472.

[3]. Abdul Fatir Ansari, A. S. (2017). Twitter Sentiment Analysis.

[4]. Abney, B. K. (2014). Labeling the Languages of Words in Mixed-Language Documents using Weakly Supervised Methods. Proceedings of the 2013 Conference of the North {A}merican Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics.

[5]. Birdsong, T. (2018, jan 13). *2018 Texting Slang Update: How to Decode What Your Teen is Saying Online*. Retrieved from mcafee.com: https://www.mcafee.com/blogs/consumer/family-safety/2018-texting-slang-update-decode-teen-saying-online/

[6]. Braja Gopal Patra, D. D. (n.d.). Sentiment Analysis of Code-Mixed Indian Languages: An Overview of SAIL_Code-Mixed Shared Task @ICON-2017.

[7]. Braja Gopal Patra, D. D. (2018). Sentiment Analysis of Code-Mixed Indian Languages: An Overview of SAIL_Code-Mixed Shared Task @ICON-2017.

[8]. Bandyopadhyay, A. D. (2010). SentiWordNet for Indian Languages. Das2010SentiWordNetFI.

[9]. Fabian Pedregosa, G. V. (2012). Scikit-learn: Machine Learning in Python. CoRR, abs/1201.0490.

[10]. Hongliang Yu, Z.-H. D. (2013, 08). Identifying Sentiment Words Using an Optimization-based Model without Seed Words. *ACL 2013, 2*, 855-859.

[11]. Ivan Provilkov, D. E. (2019). BPE-Dropout: Simple and Effective Subword Regularization. ArXiv.

[12]. Klenner, M., Tron, S., Amsler, M., & Hollenstein, N. (2014). The Detection and Analy- sis of Bi polar Phrases and Polarity Con icts. *Proceedings of 11th International Workshop on Natural Language Processing and Cognitive Science, Venice, Italy, 2014 - 2014*. ZORA: Zurich Open Repository and Archive, University of Zurich ZORA.

[13]. Kim, E. (2006). Reasons and Motivations for Code-Mixing and Code-Switching. 4 (EFL).

[14]. Liu, B. (2012). *Sentiment Analysis and Opinion Mining.* calofornia: Morgan & Claypool Publishers, May 2012.

[15]. *Monkeylearn.com*. (n.d.). Retrieved from Monkeylearn.com: https://monkeylearn.com/sentiment-analysis-examples/

[16]. Nurendra Choudhary, R. S. (2018, april 3). Sentiment Analysis of Code-Mixed Languages leveraging Resource Rich Languages.

[17]. Nurfadhlina Mohd Sharef, H. M. (2016). Overview and Future Opportunities of Sentiment Analysis Approaches for Big Data. Journal of Computer Science, 12, 153-168.

[18]. P. V. Veena, M. A. (2018). Character Embedding for Language Identification in Hindi-English Code-mixed Social Media Text.

[19]. Pruthwik Mishra, P. D. (2018). Code-Mixed Sentiment Analysis Using Machine Learning and Neural Network Approaches. ArXiv , abs/1808.03299}.

[20]. Piotr Bojanowski, E. G. (2017). Enriching Word Vectors with Subword Information. ArXiv.

[21]. Preslav Nakov, A. R. (2016). Proceedings of the 10th International Workshop on Semantic Evaluation. SemEval-2016 Task 4: Sentiment Analysis in Twitter. San Diego, California}: Association for Computational Linguistics.

[22]. RudolfEremyan. (n.d.). *Four Pitfalls of Sentiment Analysis Accuracy*. Retrieved from https://www.toptal.com/deep-learning/4-sentiment-analysis-accuracy-traps

[23]. Rico Sennrich, B. H. (2016). Neural Machine Translation of Rare Words with Subword Units. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. 1, pp. 1715-1725. Berlin, Germany: Association for Computational Linguistics.

[24]. Supriya B. Moralwar1, S. N. (2015). Different Approaches of Sentiment Analysis. *International Journal of Computer Sciences and Engineering*, 60-165.

[25]. Svetlana Kiritchenko, X. Z. (2014). Sentiment Analysis of Short Informal Texts. *Journal of Artifical Intelligence Research 50(2014) 723-762*, 723-762.

[26]. Svetlana Kiritchenko, X. Z. (2014). NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. ArXiv , 723-762.

[27]. Souvick Ghosh, S. G. (2017). Complexity Metric for Code-Mixed Social Media Tex. Computación y Sistemas , 21.

[28]. Sartiano, G. A. (2016). UniPI at SemEval-2016 Task 4: Convolutional Neural Networks for Sentiment Classification. Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). San Diego, California: Association for Computational Linguistics.

[29]. Utsab Barman, A. D. (2014). Code Mixing: A Challenge for Language Identification in the Language of Social Media. Proceedings of the First Workshop on Computational Approaches to Code Switching (pp. 13-23). Association for Computational Linguistics.

[30]. Yash Kumar Lal, V. K. (2019). De-Mixing Sentiment from Code-Mixed Text. Proceedings of the ACL 2019, Student Research Workshop (pp. 371-377). Florence, Italy: Association for Computational Linguistics.

[31]. Yogarshi Vyas, S. G. (2014). POS Tagging of English-Hindi Code-Mixed Social Media Content. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 974-979). Doha,Qatar: Association for Computational Linguistics.

ISSN: 2005-4238 IJAST