MUCS@Dravidian-CodeMix-FIRE2020: SACO-Sentiments Analysis for CodeMix Text

Fazlourrahman Balouchzahi, H L Shashirekha

Department of Computer Science, Mangalore University, Mangalore - 574199, India

Abstract

The increasing use of social media and online shopping are generating a lot of text data that consists of sentiments or opinions about anything and everything available over these platforms. Users usually use Roman script to pen their sentiments in their language in addition to using English words due to technological limitations of using their native scripts. Sentiment Analysis (SA), an automatic way of analyzing these sentiments is gaining popularity as analyzing them manually is challenging due to the huge size of the texts and the language used in these texts. In this paper, we, team MUCS, have proposed a SA model and submitted it to 'Sentiment analysis of Dravidian languages in CodeMixed Text' shared task at FIRE 2020 to analyze Tamil-English and Malayalam-English code-mixing texts. The proposed approach uses a Hybrid Voting Classifier (HVC) by combining Machine Learning (ML) models using word embeddings and n-grams features extracted from sentences with Deep Learning (DL) models based on BiLSTM using sub-words embedding features. Our team obtained 4th rank in Tamil-English and 6th rank in Malayalam-English code-mixed SA.

Keywords

Sentiments Analysis, Code-Mixing, Machine Learning, Deep Learning

1. Introduction

Sentiments in the online era are comprised of feelings or opinions of users in social media and customers' reviews or opinions of products available in online shops. The increasing use of social media such as YouTube, Facebook, WhatsApp, Instagram, Twitter, etc., and online shopping are generating lot of text data that consists of sentiments/opinions about anything and everything available on the internet. Sentiments or reviews may be positive, negative, neutral or mixed ones. Sentiment Analysis (SA) which deals with the automatic analysis of sentiments or opinions is becoming important as these texts can be a reason to raise the popularity of a product or a person or to discard a product such as a video on social media or a laptop on the online shop. It is gaining popularity as a Recommender system as many people tend to read reviews about the products available on the internet such as movie reviews, reviews about a digital camera, before deciding to watch a movie or buy a digital camera respectively. As there is no restriction on the language, content, or rules used to express the opinions, comments or reviews in social media, users are at the ease to express their feelings in any language without any hesitation [1]. However, due to technological limitations, users usually use Roman script

to pen their sentiments/opinions in their language in addition to using English words rather than using their native or local language script. One reason for this is the availability of Roman letters which can be keyed in directly as opposed to a combination of keys for a character for most of the Indian languages. This combination of more than one language using the same script in any text is called code-mixing and code-mixing texts are increasing with the popularity of social media and online shopping [2]. As words of different languages are used to write sentiments or reviews, the complexity of these texts increases and hence it becomes difficult to analyze such texts. SA is challenging due to huge size of these texts and also the code-mixing at various levels. Code-mixing includes mixing of languages in various linguistics attributes such as words, phrases, and sentences[3, 4, 5, 6]. Code-mixed text affiliate features such as vocabulary and grammar of different languages and builds new structures by combining attributes of these languages. This is a challenging process in SA models as conventional semantic analysis models do not capture the meaning of the sentences [7]. In this work, we propose a Hybrid Voting Classifier (HVC) using ML and DL approaches. While ML approaches include Multi-Layer Perceptron (MLP) and Multinomial Naïve Bayes (MNB) classifiers using n-grams and word vectors as features respectively, the DL approach uses a Bidirectional Long Short Term (BiLSTM) classifier model with sub-words embeddings as features. The implementation of this paper is available in our Github repository[1]. The rest of the paper is organized as follows: an overview of literature in the related area is discussed in Section 2. Feature extraction for the proposed model is discussed in Section 3 followed by the proposed methodology is described in Section 4. Section 5 presents the experiments and results and Section 6 concludes the paper with future work.

2. Related works

Several studies have been carried out by various researchers in collecting code-mixed data of different pairs and analyzing them for various applications including SA, language identification, POS tagging, NER, etc. A few of the important ones are given below: Chakravarthi et .al.[1] have built two code-mixed benchmarked corpus namely TamilMixSentiment and Malayalam-MixSentiment [8] for SA by collecting YouTube comments and annotating them with the help of voluntary annotators. Among the several baseline models applied on created datasets, Random Forest that randomly generates trees without defining rules gives a weighted average f-score of 0.65 for TamilMixSentiment corpus. Further, a BERT model performed better compared to other baselines with a weighted average f-score of 0.75 for MalayalamMixSentiment corpus using encoder-decoder architecture along with a mechanism to read a sequence in both (left to right and vice versa) directions. An overview of shared task on SA of Bengali- English (BN-EN) and Hindi-English (HI-EN) code-mixed data at ICON-2017 is presented by Patra et. al. [9]. Datasets are collected using Twitter4j[2] API from Twitter and have been provided to shared task participants. IIIT-NBP team (baselines) used several features such as GloVe word2vec, TF-IDF of word n-grams (n = 1, 2, 3), and character n-grams (n = 2 to 6) and achieved the highest score for both datasets with macro average f-score of 0.569 for HI-EN dataset and 0.526 for

---

[1]https://github.com/fazlfrs/SACO-SentimentsAnalysis-for-CodeMix-Text
[2]http://twitter4j.org/en/

BN-EN. A code-mixed SA using ML and Neural Network approaches has been proposed by Mishra et. al. [10] for BN-EN and HI-EN code-mixed datasets. They built the classifiers using the dataset provided by Sentiment Analysis for Indian Languages (SAIL)-Code Mixed task at ICON-2017[3]. The first classifier is a Voting classifier consisting of three classifiers namely, SVM, Logistic Regression, and Random Forests using TF/IDF with 2 to 6 char n-grams. They further experimented with the word level n-grams as features for both SVM and MultiLayer Perceptron (MLP) classifiers. The mean of GloVe vectors in a sentence (GloVe averaged) as features for SVM and MLP and Bi-LSTM with GloVe were also explored. The best results of an f-score of 0.58 and 0.69 obtained for Hindi-English and Bengali-English datasets respectively have used SVM with 2 to 6 char n-grams. Ansari et. al. [2] collected 1200 Hindi and 300 Marathi documents from social media comments and designed a model using three classification algorithms namely, Naïve Bayes and Support Vector Machine using RBF and Linear SVM. The results show the accuracy of up to 90% with consistency for Marathi language, but for Hindi language, it is in the range of 70% to 80%. A model based on contrastive learning proposed by Choudhary et. al.[7] use twin Bidirectional LSTM networks and a clustering based method to capture alteration of code-mixed transliterated words. Based on different configuration of language pairs their models obtained accuracy in the range of 71.30% to 79.80%. A hybrid model proposed by Vaibhav et. al.[3] for SA task in HI-EN code-mixed text includes sub-words embeddings. The main component of this model includes generating sub-words using CNN and a dual encoder BiLSTM network that captures the entire information about sentiments and selects more informative sentiments-bearing parts of a sentence. The system proposed by Y Joshi et. al. [11] achieved an accuracy of 83.54% and an f-score of 0.827 on the dataset released which contains 3,879 code-mixed English-Hindi sentences collected from Facebook.

3. Feature Extraction

Extracting features from text is the basis for all text processing applications [12]. The proposed approach uses three different feature extraction techniques namely, n-grams, word vectors, and sub-words vectors to extract the relevant features from code-mixing data.

- `N-grams`: An n-gram is a sequence of n items in a given sequence where an item can be a letter, phoneme, word, etc.[13]. For example, a word 2-gram (or bigram) is a two-word sequence of words like "please turn", "turn off" or "off TV" and a word 3-gram (or trigram) is a three-word sequence of words like "please turn off" or "turn off TV". Word N-gram models assign probabilities to all the words in the sentence and estimate the probability of the last word of an n-gram given the previous words. N-gram model is integrated with most of the text classification tasks and is expected to boost the accuracy of classification tasks.
- `Word Vectors`: Word2Vec transforms a text into a row of numbers such that words with similar meanings have similar vector representation called as embeddings [14]. Skip Gram and Common Bag of Words (CBOW) are the two common methods that are used to obtain Word2Vec. CBOW method predicts the target word based on a given input

---

[3]https://brajagopalcse.github.io/$SAIL_CodeMixed - ICON - 2017$
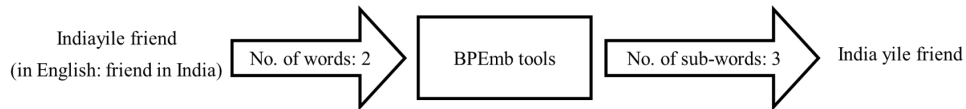
Figure 1: Converting sentence to sub-words using BPEmb tools.

context and Skipgram predicts the most probable context based on a given word. As word embeddings are not available for code-mixed data, they need to be trained from the raw code-mixed data.

- `Sub-words Vectors:` Generally NLP systems discard words that are rarely seen in training corpus as learning efficient representation for these words is a difficult challenge [15]. The problem gets further complicated due to code mixing as the code-mixed words such as "Indiayile" is not present in any language as it is a combination of two languages. However, a part of this word such as "India" which indicates a location is a valid word. This problem can be handled conveniently by considering the substrings of a word called sub-words. For example, the word "Indiayile" (meaning 'in India') commonly used in Malayalam can be considered as two sub-words "India" and "yile" where India is a name in any language and the suffix "yile" is a word in Malayalam which means 'in'. Sub-words allow finding parts of the words that otherwise seems to be unknown or out-of-vocabulary. Word embeddings are generally trained for one language and are not ideal for code-mixed data as code-mixed data, in general, may contain more than one language. Further, as word embeddings cannot handle the spelling variations in social media data [3], using sub-words embeddings will be the right choice to represent code-mixed data in addition to word embeddings. However, as pre-trained sub-words embeddings are not readily available for code-mixed data they have to be trained from the raw code-mixed data. Byte-Pair Encoding (BPEmb) [15, 16, 17, 18, 19] is a collection of pre-trained sub-words embeddings available for 275 languages trained on respective language Wikipedia[4]. It can be used to train sub-words embeddings by extracting all sub-words from a given sentence. In the proposed method, BPEmb tools from BPEmb[5] library are used to extract all the sub-words from a given sentence. Figure 1 gives a snapshot of using BPEmb tools to convert an input sentence to sub-words. These extracted sub-words are used to train sub-words embeddings using Word2Vec and is defined as Sub-Word2Vec. Sub-words embeddings can be more effective than usual word embeddings as it can capture more context than the word embeddings.

4. Methodology

As sentiments or reviews will be written in code-mixed language usually using Roman script in most of the cases, code-mixed text do not adhere to the grammar of any language thus increasing the complexity of designing the SA system. Therefore, the architecture that helps

---

[4]https://nlp.h-its.org/bpemb/
[5]https://github.com/bheinzerling/bpemb
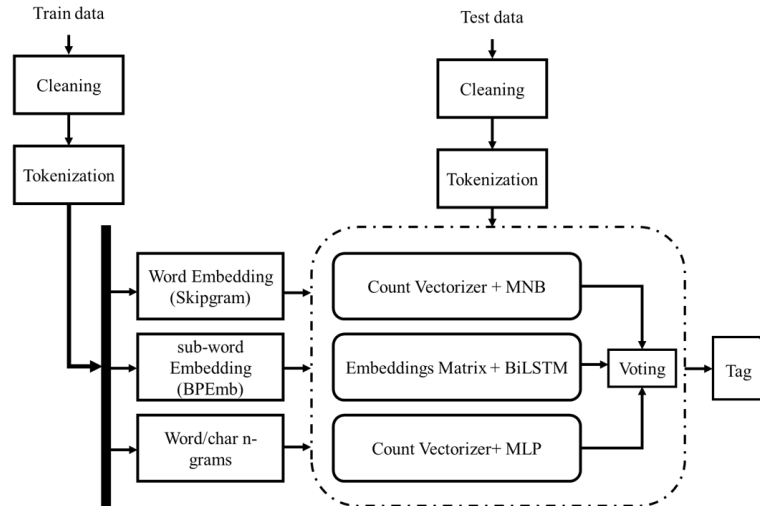
Figure 2: The architecture of Hybrid Voting Classifier.

to classify code-mixed text and overcome the challenges of analyzing such text has to be used. Hence, word and sub-word vectors and also word/char n-grams are chosen to train different learning models. For generating word and sub-word vectors, Skipgram methods are used as it calculates the probability of contexts in which a word can appear and hence will predict the most probable context of a given word. N-grams have already proved its efficiency in many natural language processing applications. Both char and word n-grams allow the model to utilize all features of a text. A voting classifier is an ensemble classification model that works based on majority voting and the tag with a higher number of votes will be the final predicted tag. It is usually built using more than two base classifiers. The proposed HVC model includes training three base models namely, BiLSTM, MNB, and MLP. The architecture of proposed approach is shown in Figure 2.

The proposed model includes a phase of feature engineering after cleaning and tokenization followed by model construction. Details of the model construction are as follows:

- `BiLSTM`: Building a DL model using BiLSTM includes two main steps: i) training sub-words embeddings Sub-Word2Vec of 100 dimensions using Word2Vec Skipgram model and ii) utilizing the sub-words embeddings to train BiLSTM networks for classifying the sentiments. The sub-words embeddings model has been trained on various batch sizes (128, 64, 32) each for 10 epochs.
- `MNB`: Raw code-mixed text is used to train Skipgram word2vec model and generated vectors are transformed using CountVectorizer from the Sklearn library and are used as features for building MNB classifiers.
- `MLP`: Building MLP includes extracting features such as char (n = 1, 2, 3, 4, 5) and word n-grams (n = 1, 2, 3) and transforming the obtained features using CountVectorizer which is fed to build an MLP Classifier.

After training, all the base models were evaluated on the test set provided by the organizers,

Table 1
Details of datasets

Class	Dataset	
	Tamil-English	Malayalam-English
Positive	10,559	2,811
Negative	2,037	738
Mixed Feelings	1,801	403
Unknown State	850	1,903
Other language	497	884
Total	15,744	6,739

Table 2
Results of top ranked teams of Dravidian-CodeMix task

Team Name	Tamil-English				Malayalam-English			
	Precision	Recall	f score	Rank	Precision	Recall	f score	Rank
MUCS	0.60	0.66	0.62	4	0.68	0.68	0.68	6

and the predicted labels were submitted to the Dravidian-CodeMix task in FIRE 2020 as MUCS team. Overview of the Dravidian-CodeMix task is discussed in reference papers [20, 21].

5. Experimental Results

Two code-mixed SA datasets namely, Malayalam-English and Tamil-English were provided by the Dravidian-CodeMix organizing team. Details of the datasets are given in Table 1. As it is mentioned in the shared task website, the performances of systems are measured by weighted averaged precision, weighted averaged recall, and weighted averaged f-score across all the classes. As per the results announced by task organizers, MUCS team obtained 4th rank in Tamil-English and 6th rank in Malayalam-English code-mixed SA. Results of the proposed approaches are shown in Table 2. MUCS team obtained an average weighted f-score of 0.62 for Tamil-English code-mixed SA which is only 0.03 less than the first rank model. Also, an average weighted f-score of 0.68 for Malayalam-English code-mixed SA is only 0.06 less than first rank.

6. Conclusion and Future work

'Sentiment analysis of Dravidian languages in CodeMixed Text' is a shared task at FIRE 2020. We, MUCS team, submitted an HVC consisting of ML approaches namely, MLP and MNB classifiers using n-grams and word-vectors as features respectively, and a DL model namely, BiLSTM classifier with sub-words embeddings as features for Sentiment analysis of Dravidian languages in CodeMixed text. Our team obtained 4th and 6th rank in Tamil-English and Malayalam-English shared task respectively. The future work is to explore code-mixed language models for low resource Indian languages and Persian language.

References

[1] B. R. Chakravarthi, V. Muralidaran, R. Priyadharshini, J. P. McCrae, Corpus creation for sentiment analysis in code-mixed Tamil-English text, in: Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL), European Language Resources association, Marseille, France, 2020, pp. 202–210. URL: https://www.aclweb.org/anthology/2020.sltu-1.28.

[2] M. A. Ansari, S. Govilkar, Sentiment analysis of mixed code for the transliterated hindi and marathi texts, International Journal on Natural Language Computing (IJNLC) Vol 7 (2018) , pp. 202–210.

[3] Y. K. Lal, V. Kumar, M. Dhar, M. Shrivastava, P. Koehn, De-mixing sentiment from code-mixed text, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, (2019), pp. 371–377.

[4] N. Jose, B. R. Chakravarthi, S. Suryawanshi, E. Sherly, J. P. McCrae, A survey of current datasets for code-switching research, in: 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020.

[5] R. Priyadharshini, B. R. Chakravarthi, M. Vegupatti, J. P. McCrae, Named entity recognition for code-mixed Indian corpus using meta embedding, in: 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020.

[6] B. R. Chakravarthi, Leveraging orthographic information to improve machine translation of under-resourced languages, Ph.D. thesis, NUI Galway, 2020.

[7] N. Choudhary, R. Singh, I. Bindlish, M. Shrivastava, Sentiment analysis of code-mixed languages leveraging resource rich languages, arXiv preprint arXiv:1804.00806 (2018).

[8] B. R. Chakravarthi, N. Jose, S. Suryawanshi, E. Sherly, J. P. McCrae, A sentiment analysis dataset for code-mixed Malayalam-English, in: Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL), European Language Resources association, Marseille, France, 2020, pp. 177–184. URL: https://www.aclweb.org/anthology/2020.sltu-1.25.

[9] B. G. Patra, D. Das, A. Das, Sentiment analysis of code-mixed indian languages: An overview of sail_code-mixed shared task@ icon-2017, arXiv preprint arXiv:1803.06745 (2018).

[10] P. Mishra, P. Danda, P. Dhakras, Code-mixed sentiment analysis using machine learning and neural network approaches, arXiv preprint arXiv:1808.03299 (2018).

[11] A. Joshi, A. Prabhu, M. Shrivastava, V. Varma, Towards sub-word level compositions for sentiment analysis of hindi-english code mixed text, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, (2016), pp. 2482–2491.

[12] H. Liang, X. Sun, Y. Sun, Y. Gao, Text feature extraction based on deep learning: a review, EURASIP journal on wireless communications and networking 2017 (2017) , pp. 1–12.

[13] P. Stefanovič, O. Kurasova, R. Štrimaitis, The n-grams based text similarity detection approach using self-organizing maps and similarity measures, Applied Sciences 9 (2019) 1870.

[14] E. L. Goodman, C. Zimmerman, C. Hudson, Packet2vec: Utilizing word2vec for feature extraction in packet data, arXiv preprint arXiv:2004.14477 (2020).

[15] B. Heinzerling, M. Strube, Bpemb: Tokenization-free pre-trained subword embeddings in 275 languages, arXiv preprint arXiv:1710.02187 (2017) , pp. 2989–2993.

[16] B. R. Chakravarthi, M. Arcan, J. P. McCrae, Improving Wordnets for Under-Resourced Languages Using Machine Translation, in: Proceedings of the 9th Global WordNet Conference, The Global WordNet Conference 2018 Committee, 2018. URL: http://compling.hss.ntu.edu.sg/events/2018-gwc/pdfs/GWC2018_paper_16.

[17] B. R. Chakravarthi, M. Arcan, J. P. McCrae, Comparison of Different Orthographies for Machine Translation of Under-Resourced Dravidian Languages, in: 2nd Conference on Language, Data and Knowledge (LDK 2019), volume 70 of *OpenAccess Series in Informatics (OASIcs)*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2019, pp. 6:1–6:14. URL: http://drops.dagstuhl.de/opus/volltexte/2019/10370. doi:10.4230/OASIcs.LDK.2019.6.

[18] B. R. Chakravarthi, M. Arcan, J. P. McCrae, WordNet gloss translation for under-resourced languages using multilingual neural machine translation, in: Proceedings of the Second Workshop on Multilingualism at the Intersection of Knowledge Bases and Machine Translation, European Association for Machine Translation, Dublin, Ireland, 2019, pp. 1–7. URL: https://www.aclweb.org/anthology/W19-7101.

[19] B. R. Chakravarthi, R. Priyadharshini, B. Stearns, A. Jayapal, S. S, M. Arcan, M. Zarrouk, J. P. McCrae, Multilingual multimodal machine translation for Dravidian languages utilizing phonetic transcription, in: Proceedings of the 2nd Workshop on Technologies for MT of Low Resource Languages, European Association for Machine Translation, Dublin, Ireland, 2019, pp. 56–63. URL: https://www.aclweb.org/anthology/W19-6809.

[20] B. R. Chakravarthi, R. Priyadharshini, V. Muralidaran, S. Suryawanshi, N. Jose, E. Sherly, J. P. McCrae, Overview of the track on Sentiment Analysis for Dravidian Languages in Code-Mixed Text, in: Proceedings of the 12th Forum for Information Retrieval Evaluation, FIRE '20, (2020).

[21] B. R. Chakravarthi, R. Priyadharshini, V. Muralidaran, S. Suryawanshi, N. Jose, E. Sherly, J. P. McCrae, Overview of the track on Sentiment Analysis for Dravidian Languages in Code-Mixed Text, in: Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2020). CEUR Workshop Proceedings. In: CEUR-WS. org, Hyderabad, India, (2020).