Zyy1510 team at SemEval-2020 Task 9: Sentiment Analysis for Code-Mixed Social Media Text with Sub-word Level Representations

Yueying Zhu, Xiaobing Zhou*, Hongling Li, Kunjie Dong
School of Information Science and Engineering
Yunnan University, Yunnan, P.R. China
*Corresponding author, zhouxb@ynu.edu.cn

Abstract

This paper reports the zyy1510 team's work in the International Workshop on Semantic Evaluation (SemEval-2020)[1] shared task on Sentiment analysis for Code-Mixed (Hindi-English, English-Spanish) Social Media Text. The purpose of this task is to determine the polarity of the text, dividing it into one of the three labels positive, negative and neutral. To achieve this goal, we propose an ensemble model of word n-grams-based Multinomial Naive Bayes (MNB) and sub-word level representations in LSTM (Sub-word LSTM) to identify the sentiments of code-mixed data of Hindi-English and English-Spanish. This ensemble model combines the advantage of rich sequential patterns and the intermediate features after convolution from the LSTM model, and the polarity of keywords from the MNB model to obtain the final sentiment score. We have tested our system on Hindi-English and English-Spanish code-mixed social media data sets released for the task. Our model achieves the F1 score of 0.647 in the Hindi-English task and 0.682 in the English-Spanish task, respectively.

1 Introduction

Mixing language, also known as code-mixing, is a norm in multilingual societies. Many multilingual people tend to be code-mixed by using English-based speech types and the insertion of English into their main language (Patwa et al., 2020), which share their views on social media by combining local and English languages, creating lots of code-mixed text such as Hindi-English and English-Spanish (Ramanarayanan and Suendermann-Oeft, 2017). Today, many organizations rely heavily on sentiment analysis of social media texts for product performance and consider user feedback when upgrading to newer versions (Jhanwar and Das, 2018). The government can predict people's emotions and know people's opinions on the new policy and so on.

Code-mixing (Vyas et al., 2014) is a relatively new field compared to the general field of sentiment analysis (Zhao et al., 2010). Social media code-mixed texts generally have three forms: i) Mixed script: a combination of the native-Roman script; ii) Code-Mixed script: a script written in Roman script in native and English languages; iii) Native script: local languages written in native languages.

This type of text needs to be handled differently, which is very different from traditional English texts (Prabhu et al., 2016). Beyond some of the challenges of general sentiment analysis, code-mixed texts have some unseen difficulties in natural language processing (NLP) tasks. Traditional NLP systems heavily rely on monolingual resources to address code-mixed text, which limits their ability to handle problems such as English-based speech input, word-level code-mixing, et al (Patwa et al., 2020). Furthermore, there are several variations when switching from a phonetic language into a Roman script (Jhanwar and Das, 2018). To solve this problem, we preprocess the text and normalize irregular words. Before we preprocess the text, we also need to eliminate the noise in the text, and translate the abbreviations into the appropriate regular words, and perform a clustering algorithm to get the most suitable one of the last few variants when transliterating non-Roman script code-mixed data as Roman scripts in preprocessing step.

[1] http://alt.qcri.org/semeval2020/

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1354–1359
Barcelona, Spain (Online), December 12, 2020.

In our work, we introduce a Sentiment Analysis (SA) system, an ensemble model of word n-grams-based Multinomial Naive Bayes(MNB) and sub-word level representations in LSTM (Sub-word LSTM) (Prabhu et al., 2016). For Indian social media text which is developed for the SemEval-2020 shared task on sentiment analysis for Code-Mixed (Hindi-English, English-Spanish) social media text aims to detect the sentiment polarity (Wang et al., 2018) of the code-mixed text written in two different languages, Hindi and Spanish mixed with English. The traditional method MNB captures low-level word-groups of keywords to make up for grammatical inconsistencies, while the sub-word LSTM model encodes the rich sequential patterns in sparse and unstable text (Norouzi and Fleet, 2013). Our model achieves the F1 scores of 0.647 in the Hindi-English task and 0.682 in the English-Spanish task, respectively. The implementation of our system is made available via Github[2].

2 Related Work

Recently, research on emotion and mood analysis in texts became increasingly common, in part because of the availability of new sources of subjective information on the web. (Ortony et al., 1987) is one of the earliest in the area of sentiment classification. It is concerned with the actual classification and segregation of terms with emotional connotations.

(Solorio et al., 2014) described Language Identification in the First Shared Task of the Code-Switched Data held at EMNLP 2014. And (Bali et al., 2014) analyzed the data on Facebook posts generated by English-Hindi bilingual users. SAIL 2015 contest co-organized with MIKE 2015 considered sentiment analysis of tweets in Indian languages (Jhanwar and Das, 2018). (Prabhu et al., 2016) introduced learning sub-word level representations in LSTM architecture. (Ghosh et al., 2017) tried to use machine learning methods to automatically extract sentiment (positive or negative) from Facebook posts. (Shalini et al., 2018) addressed the performance of distributed representation methods for Bengali-English and Hindi-English languages in sentiment analysis tasks. (Kannan et al., 2016) used a machine learning algorithm called Multinomial Naive Bayes trained by using n-gram and SentiWordnet features, they also used a small SentiWordnet for English and Bengali without using any SentiWordnet for Hindi language, Hindi-English and Bengali-English code-mixed data. An ensemble model of character tri-grams based LSTM model and word n-grams based Multinomial Naive Bayes (MNB) model to classify the sentiments of Hindi-English code-mixed data was introduced by (Jhanwar and Das, 2018). (Ansari and Govilkar, 2018) designed the system which classifies Hindi as well as Marathi text transliterated (Romanized) documents automatically using supervised learning methods (KNN), Naive Bayes and Support Vector Machine (SVM) and ontology-based classification. (Lal et al., 2019) presented a hybrid architecture for the task of sentiment analysis of English-Hindi code-mixed data. (Mandal et al., 2018) prepared gold standard Bengali-English code-mixed data with language and polarity tag for sentiment analysis purposes, and a hybrid system combining rule-based and supervised models were developed for both languages.

3 Dataset

A recent shared task was conducted by International Workshop on Semantic Evaluation 2020 on NLP (SemEval-2020), for sentiment analysis of transliterated social media text. The organizer of SemEval-2020 provided the code-mixed data of Hindi-English and Spanish-English. The training and validation tweets were labeled one of the three labels - positive, negative and neutral. But the test was not labeled. The data split details are shown in Table 1.

Language	Training Data	validation Data	test Data
Hindi-English	14000	3000	3000
Spanish-English	12002	2998	3789

Table 1: Description of the data sets

[2]https://github.com/TroubleGilr/CodeMixed-Sentiment-Analysis

There are many inherent challenges of the code-mixed data as described previously. Examples like abbreviations of words ('please' to 'plz') and non-standard spellings (such as 'suppeerrrr' or 'timeeeeeee'). And there are several variations when switching from a phonetic language into a Roman script, as illustrated in Table 2.

Word	Meaning	Variation
मुबारक(Bahut)	more	Bahut bohot bohut
मुबारक(mubaarak)	wishes	Mobarak mubarak mubark
प्यार (pyaar)	love	Pyaar peyar pyara ... piyaar pyar

Table 2: Spelling variations of romanized words

Before training, we preprocessed the code-mixed raw data. We replaced the link in the data as the URL and removed the punctuation, stop-words and the useless emoji. We tried to make the data noiseless. We also found that some certain characters appear multiple times in a word. For instance, lol (meaning laughing out loud) can be written as loool, looool or looooool. We used a clustering algorithm to process it as lool in the pre-processing stage. And we divided the hash form into the appropriate form (HappyBirthdaySonakshiSinha as Happy Birthday Sonakshi Sinha). In addition, we transliterated non-Roman script (मुबारक, means wishes) code-mixed data as Roman scripts (mubarak). Here we chose a clustering algorithm(k-means clustering algorithm) to get the most suitable variant (Fard et al., 2018). All text is converted to lower-case and then fed to the classifier.

4 System Description

Our proposed system architecture is shown in Figure 1, which is an ensemble model of word n-grams-based Multinomial Naive Bayes (MNB) and sub-word level representations in LSTM (Sub-word LSTM) to identify the sentiments of Hindi-English and English-Spanish code-mixed data into one of the sentiment classes positive, negative or neutral. After pre-processing the sentence, we generate word-based uni-gram and bi-gram features of the sentence and then fed them to the MNB classifier. Finally, it outputs the probability of the sentence belonging to each class.

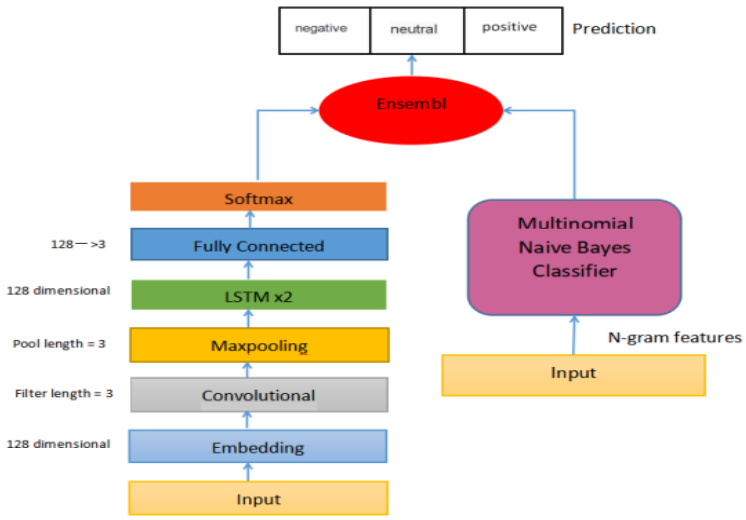

Figure 1: System Architecture for SA of code-mixed data

In our deep learning model, we feed an embedded matrix with length of 128 to the LSTM cell. We use middle level representations of the sub-word that the filter learned in the convolution operation. It propagates serviceable information with LSTM and obtains the final score of the text as illustrated in

Figure 2. The sub-word level representation is a better unit of language than characters, which can produce new lexical structures by combining characters of semantic weight.

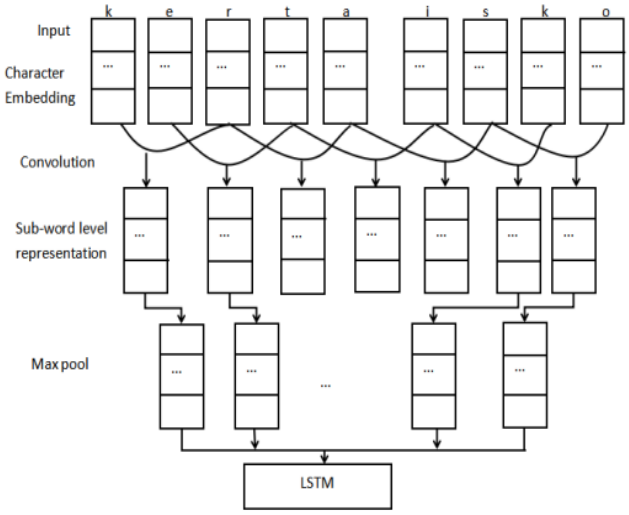

Figure 2: Illustration of the proposed methodology

Character Embedding	128	Filter length	3
Batch size	128	Pool length	3
Learning rate	0.01	dropout	0.2

Table 3: Hyper-parameters of Sub-word LSTM

Then the information is fed into a full connection (FC) layer, which achieves the interactions between these features and classes (Hochreiter and Schmidhuber, 1997). The soft maximum activation function was used to output the correct probability value. We use Adamax (Kingma and Ba, 2014) as the optimizer, a variant of Adam, to train this setup. The optimal hyperparameter configuration of the model is shown in Table 3.

5 Experiments detail

This section presents the results and compares them to several baselines. We submitted a run for each language: (1) one for Hindi-English and (2) one for Spanish-English. The final ranking for all participating systems would be based on the F1 score averaged across the positives, negatives, and the neutral.

We experimented an ensemble model of word-n-grams based Multinomial Naive Bayes (MNB) and sub-word level representations in LSTM (Sub-word LSTM) to identify the sentiments of code-mixed data of Hindi-English and English-Spanish. We chose the model with the highest combined probability by multiplying the output probability of the two models of each class. Table 4 shows the performances of our system for Hindi-English and English-Spanish language.

We observed that MNB (Unigram+Bigram) performed better for the sparse and inconsistent code-mixed data, especially rare keywords like 'fadu', meaning awesome in English (Jhanwar and Das, 2018) than SVM (Unigram+Bigram). Because the n-gram-based MNB model can successfully capture the unusual keywords. The Sub-word LSTM can extract better sequence information for long sentences. That's why we decided to use the ensemble model.

6 Conclusion and Future Work

Social media is becoming increasingly influential in people's lives. People in different positions and occupations express their views and attitudes on a certain topic. Some researchers are fascinated for the

Model	Hindi-English(F1-score)	English-Spanish (F1-score)
SVM(Unigram)	0.562	0.528
SVM(Unigram+Bigram)	0.574	0.589
MNB(Unigram)	0.569	0.631
MNB(Unigram+Bigram)	0.628	0.634
Char-LSTM(Prabhu et al.2016)	0.633	0.651
Subword-LSTM(Prabhu et al.2016)	0.635	0.653
Ensemble(our system)	0.647	0.682

Table 4: Quantitative comparison of various model proposed of Hindi-English and English-Spanish

sentiment analysis of social media text. In this paper, we proposed an ensemble model for sentiment analysis of code-mixed data for Hindi-English and English-Spanish. In the future, we're going to put emotional information into the system and also introduce new networks such as transformers, bert, attention mechanism etc.

Acknowledgements

This work was supported by the Natural Science Foundations of China under Grants 61463050, the NSF of Yunnan Province under Grant 2015FB113.

References

Mohammed Arshad Ansari and Sharvari Govilkar. 2018. Sentiment analysis of mixed code for the transliterated hindi and marathi texts. volume 7, 04.

Maziar Moradi Fard, Thibaut Thonet, and Eric Gaussier. 2018. Deep k-means: Jointly clustering with k-means and learning representations.

Souvick Ghosh, Satanu Ghosh, and Dipankar Das. 2017. Sentiment identification in code-mixed social media text.

S Hochreiter and J Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Madan Gopal Jhanwar and Arpita Das. 2018. An ensemble model for sentiment analysis of hindi-english code-mixed data. *CoRR*, abs/1806.04450.

Abishek Kannan, Gaurav Mohanty, and Radhika Mamidi. 2016. Towards building a SentiWordNet for Tamil. In *Proceedings of the 13th International Conference on Natural Language Processing*, pages 30–35, Varanasi, India, December. NLP Association of India.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *Computer Science*.

Yash Kumar Lal, Vaibhav Kumar, Mrinal Dhar, Manish Shrivastava, and Philipp Koehn. 2019. De-mixing sentiment from code-mixed text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*.

Soumil Mandal, Sainik Kumar Mahata, and Dipankar Das. 2018. Preparing bengali-english code-mixed corpus for sentiment analysis of indian languages. *CoRR*, abs/1803.04000.

Mohammad Norouzi and David J. Fleet. 2013. Cartesian k-means. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*.

Andrew Ortony, Gerald L. Clore, and Mark A. Foss. 1987. The referential structure of the affective lexicon. *Cognitive Science*, 11(3):341–364.

Parth Patwa, Gustavo Aguilar, Sudipta Kar, Suraj Pandey, Srinivas PYKL, Björn Gambäck, Tanmoy Chakraborty, Thamar Solorio, and Amitava Das. 2020. Semeval-2020 task 9: Overview of sentiment analysis of code-mixed tweets. In *Proceedings of the 14th International Workshop on Semantic Evaluation (SemEval-2020)*, Barcelona, Spain, December. Association for Computational Linguistics.

Ameya Prabhu, Aditya Joshi, Manish Shrivastava, and Vasudeva Varma. 2016. Towards sub-word level compositions for sentiment analysis of hindi-english code mixed text.

Vikram Ramanarayanan and David Suendermann-Oeft. 2017. Jee haan, i'd like both, por favor: Elicitation of a code-switched corpus of hindicenglish and spanishcenglish humancmachine dialog. In *Interspeech*.

K Shalini, HB Barathi Ganesh, M Anand Kumar, and KP Soman. 2018. Sentiment analysis for code-mixed indian social media text with distributed representation. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1126–1131. IEEE.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, and Pascale Fung. 2014. Overview for the first shared task on language identification in code-switched data. In *First Workshop on Computational Approaches to Code Switching*.

Yogarshi Vyas, Spandana Gella, Jatin Sharma, Kalika Bali, and Monojit Choudhury. 2014. Pos tagging of english-hindi code-mixed social media content. In *Conference on Empirical Methods in Natural Language Processing*.

Jin Wang, Bo Peng, and Xuejie Zhang. 2018. Using a stacked residual lstm model for sentiment intensity prediction. *Neurocomputing*, 322(DEC.17):93–101.

Yan Yan Zhao, Bing Qin, and Ting Liu. 2010. Sentiment analysis. *Journal of Software*, 21(8):1834–1848.