IUST at SemEval-2020 Task 9: Sentiment Analysis for Code-Mixed Social Media Text using Deep Neural Networks and Linear Baselines

Soroush Javdan, Taha Shangipour ataei and Behrouz Minaei-Bidgoli
Computer Engineering Department
Iran University of Science and Technology
Tehran, Iran
soroush_javdan,taha_shangipour@comp.iust.ac.ir
b_minaei@iust.ac.ir

Abstract

Sentiment Analysis is a well-studied field of Natural Language Processing. However, the rapid growth of social media and noisy content within them poses significant challenges in addressing this problem with well-established methods and tools. One of these challenges is code-mixing, which means using different languages to convey thoughts in social media texts. Our group, with the name of IUST(username: TAHA), participated at the SemEval-2020 shared task 9 on Sentiment Analysis for Code-Mixed Social Media Text, and we have attempted to develop a system to predict the sentiment of a given code-mixed tweet. We used different preprocessing techniques and proposed to use different methods that vary from NBSVM to more complicated deep neural network models. Our best performing method obtains an F1 score of 0.751 for the Spanish-English sub-task and 0.706 over the Hindi-English sub-task.

1 Introduction

Sentiment Analysis identification is a sub-field of natural language processing that explores the automatic inference for fine-grained opinion polarity of textual data. The recent growth of social media, review forums and text messaging platforms create a surge in the amount of user-generated textual data, and so increased the urgent need for automatic opinion extraction. However this evolution created many opportunities for language technology and researchers, it provided verity of new challenges, namely, spelling errors, creative invented spelling (”iz” for ”is”), abbreviation (”LOL” for ”laugh out loud”), Meta tags(Hashtags) and code-mixing (Das and Gambäck, 2014). Non-English speakers frequently use multiple languages to express their feelings, which they know as code-mixing. Speakers or writers tend to shift from one language to another either to express their feelings adequately, to show like-mindedness with a group, to distinguish oneself, to discuss a specific topic, or to look impressive to their audience. The SemEval-2020 shared task 9 (Patwa et al., 2020), which is part of the SemEval-2020 workshop, focused on coping with this challenge. This task focus is automatic sentiment analysis in a code-mixed social media text. The task consists of two subtasks, Spanish-English(Spanglish) and Hindi-English(Hinglish) code-mixing, with over 30 groups who participate in each sub-task.

This article presents a system that we have implemented for predicting the sentiment of a given code-mixed tweet. The system was developed for both subtasks. Table 1 shows the results that we achieved with our system in the SemEval-2020 competitions. To create a highly accurate classifier, we tested different methods that varied from linear (NBSVM with n-grams) and different deep learning architectures (LSTM, BiLSTM, and combinations). Also, we tested a new architecture based on Wang's (2019) work on offensive language detection, who used four Convolutional Neural Networks(CNN) with different window sizes(1,2,3,4) and k max-pooling ahead of them. However, the combination of Term Frequency-Inverse Document Frequency(TF-IDF) embedding and NBSVM yield the best result.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1270–1275
Barcelona, Spain (Online), December 12, 2020.

Competition	Placement
English-Hinish	6th
English-Spanish	7th

Table 1: SemEval-2020 results.

2 Related Work

Sentiment analysis is a well-established and well-studied topic in social media analysis and natural language processing(NLP); however, most of the effort has been applied to addressing this problem in the monolingual context. Since English is an international language and is the second most spoken language globally, it is most likely to be code-mixed with other languages. Hence, most of the works in the code-mixed criteria are focused on the combination of English with other languages.

Bhargava (2016) had divided the Code-mixed Sentiment Analysis of English and Indian languages problem, into the Language Identification and Sentiment Analysis sub-tasks. First, they used word-level language labeling and then applied SentiWordNet to analyze the semantic of each word. Overal sentiments of the sentence had obtained from all the words' sentiments. Ghosh (2017) adopted a multi-layer perceptron with varieties of extracted features such as Part-Of-Speech Tags, frequency of code switches, the density of curse words, and the number of slang words. Parvalika (2017) proposed a method based on a lexicon-based and machine learning approaches to address Hindi-English Sentiment Analysis. In the lexicon-based method, they used dictionaries of words annotated with their polarity to classify text. In the machine learning-based part, first, they did data cleaning and replace slang words with their associated meaning. Then, Naive Bayes(NB), Support Vector Machine(SVM), Decision Tree(DT), Random Tree(RT), Multilayer Perceptron were applied for text classification purposes. Prabhu (2016) addressed the same problem with sub-word level representations in LSTM architecture. They applied character-level embedding followed by convolution with a filter length of 3 in order to obtain sub-word level representations. LSTM was utilized to model the relationships between obtained features. The LSTM output feeds to a fully connected layer to calculate the final label. Lal (2019) proposed a model that consists of three different components. The first component prepared the sub-word level representations. Then the output was feed to the second component, which comprises two different BiLSTM. One BiLSTM aims to learn a representation for the overall sentiment of the sentence, and the other one, which is followed by an affixed attention mechanism, enables the selection of subwords that contribute the most towards the input text's sentiment. Finally, the concatenation of two BiLSTM aligns with linguistic features feed to a fully connected layer to predict sentence polarity. In order to address the Bambara-French code-mixed sentiment analysis problem in Facebook comments, Konate (2018) employed a variety of deep learning methods, including LSTM, BiLSTM, CNN, and their combinations. Character and word embedding produced from fixed indexes were used due to the lake of pre-trained word vectors for the Bambara language. Vilares (2015) investigated sentiment analysis for English and Spanish language separately and their combination on twitter. They extracted a set of linguistic features. They also make use of dependency parsers to achieve syntactic features. They developed an utterly monolingual model, a pipeline that adopted language identification techniques to discover the language of texts, and a multilingual model trained on Spanish-English texts. For the word embedding phase in resolving the Spanish-English code-mixed sentiment analysis problem, Yadav (2020) used various methods. In the first approach, they separately learn word embedding for each language and then used supervised and unsupervised approaches to learn a transformation matrix to map representation from one language to the representation of the other language. In the second method, they trained word, and subword representation on corpus contains text from both languages and their combination. Last, they adopted a Machine Translation approach to learning the cross-lingual sentence-level representation. They achieved the best result with the combination of subword representation and LSTM.

Labels	Hindi-English (%)	Spanish-English (%)
English words only	41.48	32.41
Hindi/Spanish words only	58.52	67.59

Table 2: A brief description of language labels distribution for the Hindi-English and Spanish-English datasets.

3 Data

Two corpora were used in Code-Mixed Sentiment Analysis shared task. The Spanish-English dataset consists of 15000 data samples for the train and 3789 for the test. On the other hand, the Hindi-English dataset contains total 15131 data samples for the train and 3000 for the test set. The sentiment labels are positive, negative, or neutral. Besides the sentiment labels, the language labels at the word level have been provided. The distribution of both language labels for each dataset is shown in Table 2. The distribution of Hindi-English indicates that using English words is a more common practice in India.

Figure 1 shows the overlap between the unique tokens between the first and second languages. There are 614 words in the Spanish-English dataset, which have the same spelling in both languages. This number is much higher for the Hindi-English dataset. It contains 5884 words with similar spelling for both Hindi and English languages, and approximately, there is an 11% overlap between English and Hindi vocabulary.

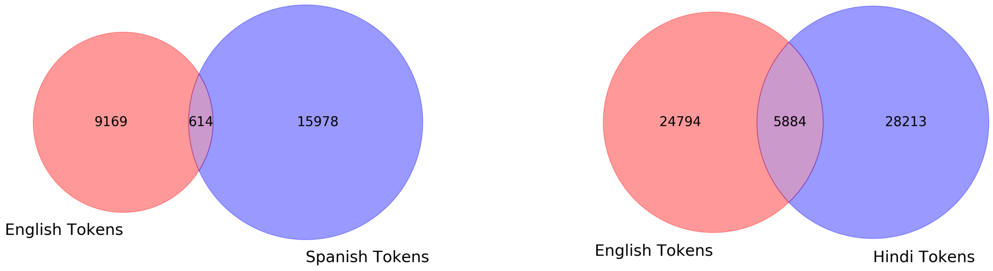

Figure 1: Vocabulary overlap for language pairs for Spanish-English and Hindi-English datasets.

4 Methodology

In this section, we describe the techniques and models that we applied to address the code-mixed Sentiment Analysis problem.

4.1 Preprocessing

Tokenization: Since social media content is noisy and contains user mentions and hashtags, the tokenization process can be turned to a challenging task. We utilized Keras[1] and nltk's[2] tweet tokenizers.

Hashtag segmentation: Hashtag is metadata, helping people find messages with the same topic within social media. Hashtag start with a number sign, #, and usually follows by a combination of words. We removed # sign and segmented the words in hashtags.

Removing URLs: we removed all of the URLs in data.

Misc.: We lowercased all the letters of the training and test data.

[1] https://keras.io/preprocessing/text/ - tokenizer

[2] http://nltk.org/

4.2 Models

NBSVM: We employed the NBSVM (Wang and Manning, 2012) model, a combination of Naïve Bayes and linear models such as Support Vector Machine(or logistic regression). The calculation ratio of log Naïve Bayes used as a feature for Support Vector Machine. Notwithstanding its simplicity, NBSVM methods have been shown to be both fast and robust across a wide variety of different text classification tasks. We fed the TF-IDF matrix with character n-gram features to model as input. We applied this method over both of the Hindi-English and Spanish-English datasets.

SVM: We used Support Vector Machine with the TF-IDF matrix with character n-gram made from data as input features.

XGboost: We utilized XGBoost (Chen and Guestrin, 2016), an optimized and distributed version of the gradient boosting model, and the TF-IDF matrix with character n-gram made from data.

Bi-GRU-CNN+BiLSTM-CNNL: Convolutional neural network is employed in text classification tasks as it is suitable for detecting patterns, and also it can detect different patterns by changing kernel sizes in different positions. The recurrent neural network is a sequence of linked network blocks that each block passes a message to its successor, this feature enables the network to demonstrate dynamic temporal behavior for a time sequence and capture sequential data. ata. For this model, we also used fastText (Mikolov et al., 2018) word embedding to extract word vector representation of data. The excellent characteristic of fastText is that it is derived from character n-gram that makes it convenient for social media's noisy content. We employ a neural network architecture built on top of a fastText English embedding with 300 dimensions. Then, the network splits into two parallel sections; the first section combines a bidirectional gated recurrent unit (GRU) with 128 hidden units and a convolutional layer with a kernel size of 2 and 64 hidden units. The second section combines a bidirectional long short term memory (LSTM) with 128 hidden units and a convolutional layer with a kernel size of 2 and 64 hidden units. Finally, we concatenate global max-pooling and global average pooling layers over parallel parts and send them to a dense layer followed with the softmax layer for classification purposes.

DistilBERT-LR: Bidirectional Encoder Representation from Transformer (BERT) (Devlin et al., 2018) released by Google research team, achieved state of the art in many NLP tasks. Moreover, since it has more than 100 million parameters, the fine-tuning procedure requires GPUs and may take a long time. Thus, we use a distilled version of BERT introduced in Sanh (2019). They reduced its parameters by 40% and retained its language understanding by 97%. We used the last layer output of DistilBERT and fed it to the logistic regression classifier. We tested both base-cased and multilingual pre-trained BERT. However, the result of multilingual BERT was slightly better.

CNN-GloVe+fastText: Same as Wang (2019), we used the average of fastText and GloVe (Pennington et al., 2014) embeddings with 300 dimensions as an input of a neural network which consists of four parallel convolutional layers with 64 hidden units and kernel sizes of 1,2,3 and 4 that their output was fed to a specific k max-pooling layers and the final output was used for prediction.

CNN-fastText: Like the previous model, we employed fastText embedding with 300 dimensions as an input of a deep neural network, which consists of four parallel convolutional layers with 128 hidden units and kernel sizes of 3, 4, 5 and 6. The batch normalization was applied after each convolutional layer, and the dropout and global max-pooling layers were used after that. Finally, the concatenation of four outputs is used for prediction.

5 Results

For all models, we used inputs with and without preprocessing. Also, we experiment with different parameter and reports the set of parameters with the best performance. Moreover, we tested all the models, as mentioned earlier, for both Hindi-English and Spanish-English datasets. The performance of NBSVM was quite impressive. We used the character n-gram with a range of 2 to 6 and the TFIDT matrix. The language independence characteristic of this model makes it a strong candidate for the code-mixed classification tasks. Our experiment with the DistilBERT model was not satisfying. However, multilingual BERT performed slightly better. The combination of CNN, fastText, and GloVe was not as expected. The poor performance was as a result of using only English embedding and ignoring the other language. The

	f1 score			
Method	**Negative**	**Neutral**	**Positive**	**Macro Average**
NBSVM ngram range 2 and 6	**0.72**	**0.65**	**0.76**	**0.71**
XGB ngram range 2 and 5	0.59	0.59	0.67	0.62
CNN-Fasttext	0.69	0.59	0.55	0.66
DistilBERT + LR	0.64	0.50	0.70	0.61
CNN-GloVe+Fasttext	0.70	0.56	0.72	0.66

Table 3: Models performance over Hindi-English dataset

complete results of our tested models for Hindi-English and Spanish-English are shown in Table 3 and 4, respectively.

	f1 score			
Method	**Negative**	**Neutral**	**Positive**	**Macro Average**
NBSVM ngram range 2 and 6	0.336	**0.164**	**0.841**	**0.750**
Bi-GRU-CNN+BiLSTM-CNN	**0.372**	0.158	0.837	0.736
DistilBERT-LR (BERT-base-cased)				0.730
DistilBERT-LR (BERT-multilingual)				0.734
CNN-Fasttext				0.682
SVM				0.671

Table 4: Models performance over Spanish-English dataset

6 Conclusion

Our proposed methods ranked 6 out of 62 groups for the Hindi-English dataset and ranked 7 out of 29 for the Spanish-English dataset. This result shows the strength of the combination of NBSVM and TF-IDF as a language independence Model. Also, our experiment shows the substantial adverse effect of ignoring one of the language's representation. For future work, we are going to utilize both languages pre-trained embedding. Averaging is one way to combine both embeddings. However, the different output space of both embedding could be challenging. Also, the attention mechanism could be used to give the model the ability to decide the importance of each language representation within each sentence.

References

Rupal Bhargava, Yashvardhan Sharma, and Shubham Sharma. 2016. Sentiment analysis for mixed script indic sentences. In *2016 International conference on advances in computing, communications and informatics (ICACCI)*, pages 524–529. IEEE.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Amitava Das and Björn Gambäck. 2014. Identifying languages at the word level in code-mixed indian social media text.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Souvick Ghosh, Satanu Ghosh, and Dipankar Das. 2017. Sentiment identification in code-mixed social media text. *arXiv preprint arXiv:1707.01184*.

Arouna Konate and Ruiying Du. 2018. Sentiment analysis of code-mixed bambara-french social media text using deep learning techniques. *Wuhan University Journal of Natural Sciences*, 23(3):237–243.

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*, pages 371–377.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

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.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Ameya Prabhu, Aditya Joshi, Manish Shrivastava, and Vasudeva Varma. 2016. Towards sub-word level compositions for sentiment analysis of hindi-english code mixed text. *arXiv preprint arXiv:1611.00472*.

A Pravalika, Vishvesh Oza, NP Meghana, and S Sowmya Kamath. 2017. Domain-specific sentiment analysis approaches for code-mixed social network data. In *2017 8th international conference on computing, communication and networking technologies (ICCCNT)*, pages 1–6. IEEE.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

David Vilares, Miguel A Alonso, and Carlos Gómez-Rodríguez. 2015. Sentiment analysis on monolingual, multilingual and code-switching twitter corpora. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 2–8.

Sida Wang and Christopher Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 90–94, Jeju Island, Korea, July. Association for Computational Linguistics.

Bin Wang, Xiaobing Zhou, and Xuejie Zhang. 2019. YNUWB at SemEval-2019 task 6: K-max pooling CNN with average meta-embedding for identifying offensive language. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

Siddharth Yadav and Tanmoy Chakraborty. 2020. Unsupervised sentiment analysis for code-mixed data. *arXiv preprint arXiv:2001.11384*.