NITS-Hinglish-SentiMix at SemEval-2020 Task 9: Sentiment Analysis For Code-Mixed Social Media Text Using an Ensemble Model

Subhra Jyoti Baroi Nivedita Singh Ringki Das Thoudam Doren Singh
Department of Computer Science and Engineering
National Institute of Technology Silchar
Silchar, Assam -788010
{tushar.barroi99,s.nivedita279,ringkidas,thoudam.doren}@gmail.com

Abstract

Sentiment Analysis is the process of deciphering what a sentence emotes and classifying them as either positive, negative, or neutral. In recent times, India has seen a huge influx in the number of active social media users and this has led to a plethora of unstructured text data. Since the Indian population is generally fluent in both Hindi and English, they end up generating code-mixed Hinglish social media text i.e. the expressions of Hindi language, written in the Roman script alongside other English words. The ability to adequately comprehend the notions in these texts is truly necessary. Our team, **rns2020** participated in Task 9 at SemEval2020 intending to design a system to carry out the sentiment analysis of code-mixed social media text. This work proposes a system named NITS-Hinglish-SentiMix to viably complete the sentiment analysis of such code-mixed Hinglish text. The proposed framework has recorded an F-Score of 0.617 on the test data.

1 Introduction

With the advent of social media, India alone stood at 326.1 million users in 2018. A large portion of the population is fluent in both English and Hindi henceforth it is regular to see the use of code-mixed Hinglish language on social media platforms. In the world of natural language processing(NLP), sentiment analysis of the code-mixed Hinglish text is a great challenge. The task at hand is the SemEval-2020 Task 9 (Patwa et al., 2020): Sentiment Analysis for Code-Mixed Social Media Text for Hinglish. Code-mixed Hinglish tweets are provided as training, validation, and test sets for model training, validation, and testing respectively. The proposed framework NITS-Hinglish-SentiMix, an ensemble model wherein different models have been consolidated to improve the general F-Score of the framework. The referenced group model is a mix of a basic LSTM (Long Short Term Memory), an LSTM+Convolution, a BiLSTM (Bidirectional LSTM), and a CNN (Convolution Neural Network) model. This paper is composed as follows, Section 2 is a concise writing overview of related works. Section 3 talks about the strategy which covers bits of insights into the dataset given, alongside the pre-processing, the details on the model architecture. Section 4 talks about the training and results acquired and the paper concludes with Section 5.

2 Literature Survey

In recent years, a lot has changed in the field of sentiment analysis. This section discusses a few of the works which helped us build insight into the problem and come up with a new solution. The initial approach towards sentiment analysis includes systems powered by traditional machine learning techniques like the work by (Agarwal and Bhattacharyya, 2005) explores the performance of SVM for the task of multinomial classification of documents. The approach used the WordNet synonym graph with good and bad as anchor points to find out the mutual relationship of words between the documents and the

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1298–1303
Barcelona, Spain (Online), December 12, 2020.

sentences.These relationships were then used for sentiment prediction with the help of SVM. Graph cut techniques were also applied to improve classification accuracy. The work by (Wang et al., 2012) proposes a real-time system for sentiment analysis of tweets made of many steps but only 2 steps were of prime importance i.e pre-processing and sentiment prediction. Pre-processing includes tokenization with a tokenizer that handled the URLs, common emoticons, phone numbers, HTML tags, twitter mentions and hashtags, numbers with fractions and decimals, repetition of symbols and Unicode characters dexterously and ensured no noise in the processed data. The statistical classifier used for sentiment analysis or prediction was a Naïve Bayes model on unigram features. The architecture and the methods used were generic and hence the domain of this model could be extended. The paper proposed by (Mathur et al., 2018b) focuses on the classification of offensive tweets written in the Hinglish language. They approach the problem of sentiment analysis or classification of the tweets in the HEOT (Hindi-English Offensive Tweet) dataset using the concepts of transfer learning where the proposed model consists of convolutional neural networks (CNN) pre-trained English tweets followed by retraining on Hinglish tweets. Their paper also examined various CNN based models for the task but leaves out other deep learning models based on LSTM and BiLSTM which are expected to show a high affinity towards semantic-based tasks like sentiment analysis. The works proposed by (Kenyon-Dean et al., 2018) helped in manually annotating the data as it provides an insight into how humans annotate data. The study was conducted on a dataset specifically prepared for this purpose called the McGill Twitter Sentiment Analysis (MTSA) dataset containing 7,026 tweets. Translating code-mixed social media comments is attractive and challenging research. Work by (Singh and Solorio, 2018) suggested a translation model on code-mixed Facebook comments. The proposed system was based on two approaches, one is language identified and without using language identifier. The Hindi-English code-mixed model has achieved an improved result over the baseline model. Code-switching is an alternation of spoken language i.e. utterance or conversation in the multilingual community. In various natural language processing tasks like named entity tasks, parts of speech tagging, sentiment analysis, machine translation, and conversational system code-switching played a major role. (Jose et al., 2020) surveyed a current code-switching dataset and categorized them.

3 Methodology

In this section, a detailed description of the dataset used is given along with the detailed steps employed for the pre-processing of the text data and the details of the proposed model architectures.

3.1 Dataset

The given dataset comprises of tweets(entries) in code-mixed Hinglish. Each entry contains Twitter handle(s) and links to the corresponding tweet at the beginning and the end of respectively. The entire dataset (train set + validation set) has a total of 17000 sentences with an average sentence length of 134.9 characters, has a vocabulary size of 60141. The training set alone has a count of 381970 words making 14594 sentences with 136.2 characters as average sentence length and a vocabulary size of 60115. With a total of 3000 sentences, the validation set had a vocabulary size of 19499 and average sentence length 127.7 characters. The test set also had a total of 3000 sentences vocabulary size of 19331 with an average sentence length of 129.9 characters.

3.2 Pre-Processing

Removing Noise: The twitter handles and URL links are the main source of noise in the given dataset. Simple regular expressions were created to delete anything which was followed by @ to remove the twitter handles and anything starting with HTTP to remove the URL links.

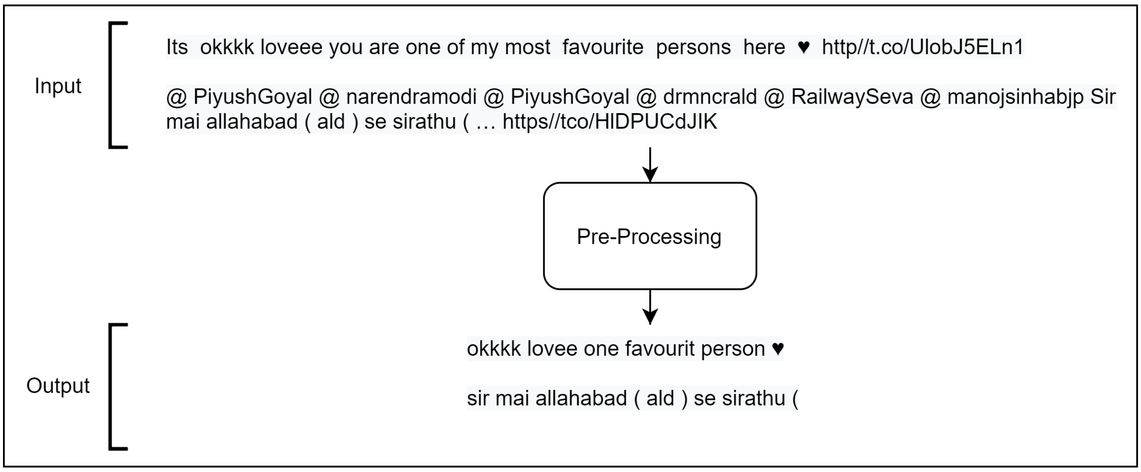
Removing Punctuation marks and Special symbols: All kinds of punctuation marks and special symbols appearing in the dataset were cleared since these characters add no value to text-understanding and in turn induce noise into systems.

Removing Stop words: In English, stop words like a, an, the, is, etc. are generally added to make sentences grammatically correct and since they carry minimal value it is apt to remove them so that the focus stays on sentiment determining words. A list of such words is present in the NLTK library.

Stemming: Using a snowball stemmer (chosen experimentally) the process of stemming was carried out on the dataset. This was done to bring the words to their respective base forms. This helped a lot in correcting words with unwanted suffixes and skewed spellings which are a common sight in the social media text.

Label Encoding: Categorical sentiment values were label encoded as 0,1,2 to negative, neutral, and positive sentiments respectively. This was done to give a numeric representation to the categorical data.

Removing high-frequency Hindi words: Unlike English, there does not exist any list of Hindi stop words so such a list was prepared based on Term Frequency (TF) over the entire dataset. The most frequently occurring 1000 words were removed.



Its okkkk loveee you are one of my most favourite persons here ♥ http//t.co/UlobJ5ELn1

@ PiyushGoyal @ narendramodi @ PiyushGoyal @ drmncrald @ RailwaySeva @ manojsinhabjp Sir mai allahabad (ald) se sirathu (… https//tco/HIDPUCdJIK

okkkk lovee one favourit person ♥

sir mai allahabad (ald) se sirathu (

Figure 1: Pre Processing Example

3.3 Model Architecture

The proposed system NITS-Hinglish-SentiMix[1] is an ensemble model made using on 4 different individual models each of which has been discussed in details in the following section.

3.3.1 LSTMs

A simple model which consists of a single layer of LSTM(Rao et al., 2018) followed by 2 dense layers, the details are shown in Figure 2(a).

3.3.2 LSTM + Convolution Layer

This model contains a convolution layer with kernel size 3 followed by a global max pool layer, LSTM layer and a dense layer(Mathur et al., 2018a) for which the details are shown in Figure 2(b).

3.3.3 BiLSTMs

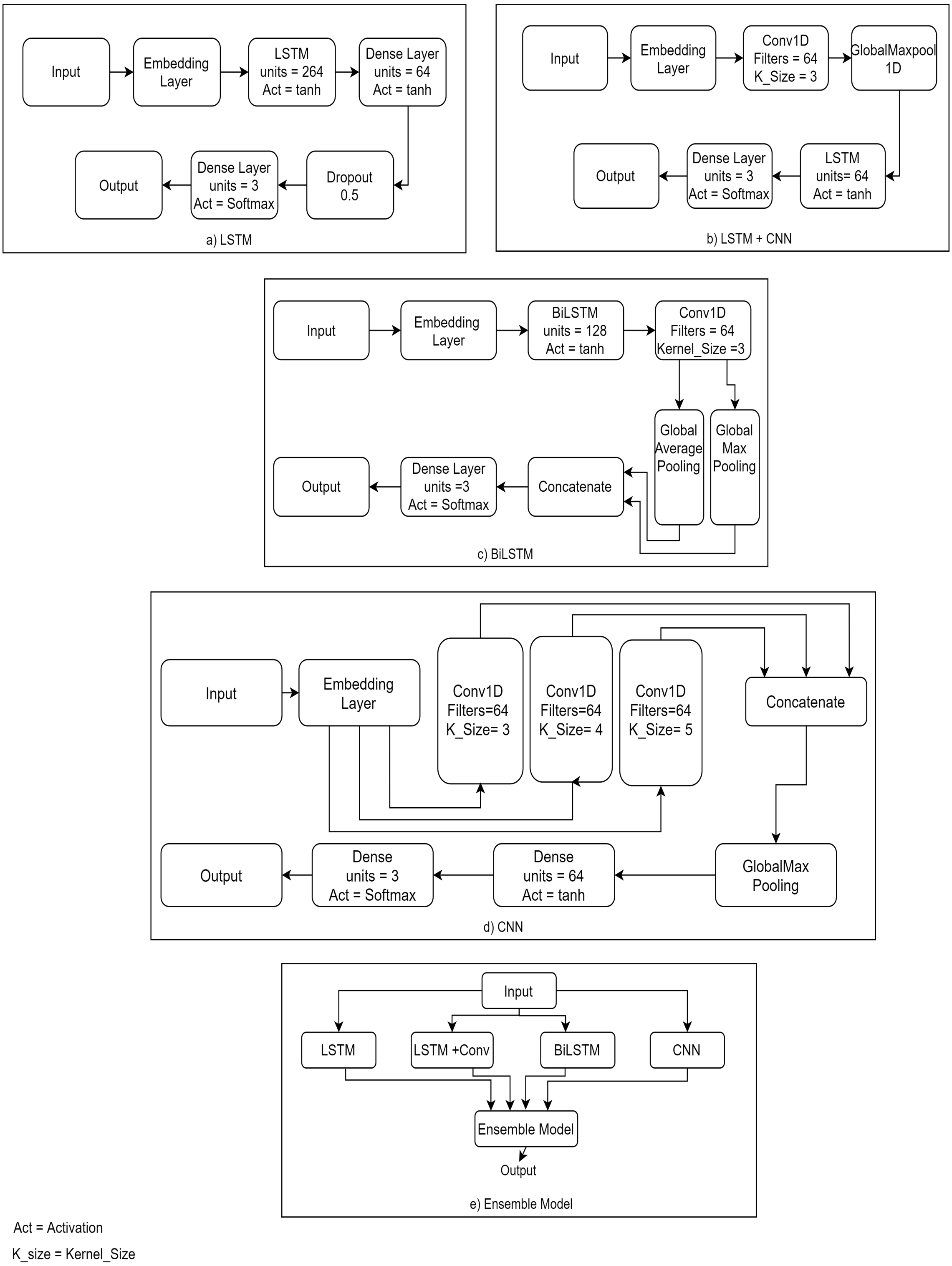
In this model, a BiLSTM layer followed by a convolution layer with kernel size 3 is employed. The output of this layer is passed through two different layers namely the global average pool and global max pool. The output is concatenated and then passed to a dense layer[2]. Figure 2(c) shows details of Model 3.

3.3.4 CNN

This particular model uses 3 different convolution layers with kernel size 3, 4 and 5 connected to the embedding layer. The outputs of each layer is concatenated and then passed to a global max pool layer followed by 2 dense layers (Abu Farha and Magdy, 2019) as shown in Figure 2(d).

[1] https://github.com/singhnivedita/SemEval2020-Task9

[2] https://github.com/tensorflow/docs/blob/master/site/en/tutorials/text/text_classification_rnn.ipynb


Input
Embedding Layer
LSTM units = 264 Act = tanh
Dense Layer units = 64 Act = tanh
Dropout 0.5
Dense Layer units = 3 Act = Softmax
Output
Input
Embedding Layer
Conv1D Filters = 64 K_Size = 3
GlobalMaxpool 1D
LSTM units= 64 Act = tanh
Dense Layer units = 3 Act = Softmax
Output
Input
Embedding Layer
BiLSTM units = 128 Act = tanh
Conv1D Filters = 64 Kernel_Size =3
Global Average Pooling
Global Max Pooling
Concatenate
Dense Layer units =3 Act = Softmax
Output
Input
Embedding Layer
Conv1D Filters=64 K_Size= 3
Conv1D Filters=64 K_Size= 4
Conv1D Filters=64 K_Size= 5
Concatenate
GlobalMax Pooling
Dense units = 64 Act = tanh
Dense units = 3 Act = Softmax
Output
Input
LSTM
LSTM +Conv
BiLSTM
CNN
Ensemble Model
Output
Act = Activation
K_size = Kernel_Size

a) LSTM

b) LSTM + CNN

c) BiLSTM

d) CNN

e) Ensemble Model

Figure 2: Models

3.3.5 NITS-Hinglish-SentiMix Framework

For better results, an ensemble model was constructed to harness the strength of each individual model. On passing the inputs to all models, the outputs were denoted by

$$O_n \text{ where n was the no of the model stated in the previous section.}$$

$$O_n = \sum_i O_n^i \text{where i = no of sentences.}$$

$$O_n^j \text{ denotes the probability of class j for the } n^{th} \text{ model}$$

The final output matrix was calculated using the formula shown below. After calculating, each sentence was assigned a class with the maximum probability.

$$O_final = max(O_{10}, O_{20}, O_{30}, O_{40}), max(O_{11}, O_{21}, O_{31}, O_{41}), max(O_{12}, O_{22}, O_{32}, O_{42})$$

The Figure 2(e) shows the diagrammatic representation of the proposed ensemble model.

4 Results

Each of the independent models were trained for 200 epochs with a batch size of 128, vocabulary size of 20000, text sequence length of 50 with sparse categorical loss and learning rate of 0.01. The results have been summarised in the Table no 1.

Parameter	**LSTM**	**LSTM+Conv**	**BiLSTM**	**CNN+Dense**
Validation F-Score	0.8413	0.8660	0.9023	0.7757
Test F-Score	0.5640	0.5747	0.576	0.5737

Table 1: Individual Model Accuracies

NITS-Hinglish-SentiMix ensemble model performed optimally with 5 epochs with all the other training parameters remaining same as mentioned above for the respective models. The overall F-Score achieved by NITS-Hinglish-SentiMix on the final submission was 0.617. Table no 2 contains the individual F-score of all the different models on validation and test set.

Parameter	**LSTM**	**LSTM+Conv**	**BiLSTM**	**CNN+Dense**	**Our Model**
Validation F-Score	0.7720	0.8043	0.8510	0.8360	0.7623
Test F-Score	0.5953	0.6077	0.5810	0.6070	0.617

Table 2: NITS-Hinglish-SentiMix(Our Model) validation accuracies for each model

Conclusion

In this paper, a detailed approach for the sentiment analysis of the code- mixed Hinglish data is described. NITS-Hinglish-SentiMix is an ensemble model over four distinct models that on their own did not fare well with the task which is very much visible in their respective test accuracies on the test data as seen in Table 1. However, it is observed that each individual model was able to catch a particular sentiment exceptionally well hence the decision to adopt an ensemble model. NITS-Hinglish-SentiMix, an ensemble model built on detailed pre-processing and extensive model training, achieves an F-score of 0.617 on the test data. A voted ensemble may be attempted to improve the score further as a future direction of the work.

References

Ibrahim Abu Farha and Walid Magdy. 2019. Mazajak: An online Arabic sentiment analyser. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 192–198, Florence, Italy, August. Association for Computational Linguistics.

Alekh Agarwal and Pushpak Bhattacharyya. 2005. Sentiment analysis: A new approach for effective use of linguistic knowledge and exploiting similarities in a set of documents to be classified. In *Proceedings of the International Conference on Natural Language Processing (ICON).*

N. Jose, B. R. Chakravarthi, S. Suryawanshi, E. Sherly, and J. P. McCrae. 2020. A survey of current datasets for code-switching research. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 136–141.

Kian Kenyon-Dean, Eisha Ahmed, Scott Fujimoto, Jeremy Georges-Filteau, Christopher Glasz, Barleen Kaur, Auguste Lalande, Shruti Bhanderi, Robert Belfer, Nirmal Kanagasabai, Roman Sarrazingendron, Rohit Verma, and Derek Ruths. 2018. Sentiment analysis: It's complicated! In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1886–1895, New Orleans, Louisiana, June. Association for Computational Linguistics.

Puneet Mathur, Ramit Sawhney, Meghna Ayyar, and Rajiv Shah. 2018a. Did you offend me? classification of offensive tweets in Hinglish language. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 138–148, Brussels, Belgium, October. Association for Computational Linguistics.

Puneet Mathur, Rajiv Shah, Ramit Sawhney, and Debanjan Mahata. 2018b. Detecting offensive tweets in Hindi-English code-switched language. In *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, pages 18–26, Melbourne, Australia, July. Association for Computational Linguistics.

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.

Guozheng Rao, Weihang Huang, Zhiyong Feng, and Qiong Cong. 2018. Lstm with sentence representations for document-level sentiment classification. *Neurocomputing*, 05.

Thoudam Doren Singh and Thamar Solorio. 2018. Towards translating mixed-code comments from social media. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, pages 457–468, Cham. Springer International Publishing.

Hao Wang, Dogan Can, Abe Kazemzadeh, François Bar, and Shrikanth Narayanan. 2012. A system for real-time twitter sentiment analysis of 2012 U.S. presidential election cycle. In *Proceedings of the ACL 2012 System Demonstrations*, pages 115–120, Jeju Island, Korea, July. Association for Computational Linguistics.