JUNLP at SemEval-2020 Task 9: Sentiment Analysis of Hindi-English code mixed data using Grid Search Cross Validation

Avishek Garain[1], Sainik Kumar Mahata[2], Dipankar Das[3]
[123]Jadavpur University, Kolkata
[1]avishekgarain@gmail.com, [2]sainik.mahata@gmail.com,
[3]dipankar.dipnil2005@gmail.com

Abstract

Code-mixing is a phenomenon which arises mainly in multilingual societies. Multilingual people, who are well versed in their native languages and also English speakers, tend to code-mix using English-based phonetic typing and the insertion of anglicisms in their main language. This linguistic phenomenon poses a great challenge to conventional NLP domains such as Sentiment Analysis, Machine Translation, and Text Summarization, to name a few. In this work, we focus on working out a plausible solution to the domain of Code-Mixed Sentiment Analysis. This work was done as participation in the SemEval-2020 Sentimix Task, where we focused on the sentiment analysis of English-Hindi code-mixed sentences. our username for the submission was "sainik.mahata" and team name was "JUNLP". We used feature extraction algorithms in conjunction with traditional machine learning algorithms such as SVR and Grid Search in an attempt to solve the task. Our approach garnered an f1-score of 66.2% when tested using metrics prepared by the organizers of the task.

1 Introduction

India has a linguistically diverse population due to its long history of foreign acquaintances. English, one of those borrowed languages, became an integral part of the education system and hence gave rise to a population who are very comfortable using bilingualism in their day to day communication. Due to such language diversity and dialects, frequent code-mixing is encountered during conversations. Further, due to the emergence of social media, the practice has become even more widespread. The phenomenon is so common that it is often considered as a different (emerging) variety of the language, e.g., Benglish (Bengali-English) and Hinglish (Hindi-English).

This phenomenon poses a great challenge to the existing domains of Natural Language Processing (NLP) such as Sentiment Analysis as primarily the language technologies, such as parsing, Parts-of-Speech (POS) tagging, etc., are built for English. Furthermore, labeled/annotated data of such category are hard to come by and hence leads to misfiring when using straight-forward machine learning algorithms.

In this work, we participated in SemEval-2020 Sentimix Task[1] and attempted to solve the chore of sentiment analysis of English-Hindi code-mixed sentences.

Initially, our approach includes the use of feature extraction algorithms on the data, procured by the organizers. Thereafter, we used Support Vector Regression coupled with Grid Search algorithm to classify the code-mixed sentences to its respective sentiment class. This approach, when tested using the metrics prepared by the organizers, returned an f1-score of 66.2%.

The rest of the paper is organized as follows. Section 2 briefly the quantifies the English-Hindi code-mixed data procured by the organizers of the task. Section 3 provides a descriptive literature of our proposed approach. This will be followed by the results and concluding remarks in Section 4 and 5.

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: http://creativecommons.org/licenses/by/4.0/.

* All authors contributed equally to this work.

[1]https://competitions.codalab.org/competitions/20654

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1276–1280
Barcelona, Spain (Online), December 12, 2020.

2 Data

The English-Hindi code-mixed data that was used to train our model was collected from Twitter using the Twitter API, by searching for code-mixed Hindi keywords (Patwa et al., 2020). The sentiment labels are positive, negative, and neutral. Besides the sentiment labels, the language labels for every word of the code-mixed sentence were also provided. The word-level language tags were ENG (English), HIN (Hindi), and O (Other) for symbols, mentions, and hashtags.

The organizers provided a trial and a training data set and after adding both, we could gather 17,000 code-mixed instances. We further divided this data into two parts; (i.) 15,000 instances as training data and (ii.) 2,000 instances as validation data.

3 Methodology

Our approach included converting the given tweets into a sequence of words and then run the Grid Search Cross-Validation algorithm on the processed tweet. Initially, the tweets were pre-processed using methods as done by (Garain and Basu, 2019a) to remove the following:

1. Removing mentions
2. Removing punctuation
3. Removing URLs
4. Contracting white space
5. Extracting words from hashtags

The last step consisted of taking advantage of the Pascal Casing of hashtags (e.g. `#CoronaVirus`). A simple regex can extract all words. This extraction results in better performance mainly because words in hashtags, to some extent, may convey sentiments of hate. They play an important role during the model-training stage.

3.1 Feature Extraction

After obtaining clean tweets, various features were extracted by treating them as a sequence of words. Some of the features were manually extracted while some were extracted using pre-existing methodologies like the Bag-of-Words model, GloVe vectors. As our aim is Sentiment Analysis of the texts, so the presence of hate, offense, humor, etc., may have a great influence on the result. The extracted features are listed below.

1. TF-IDF Vector features: The TF-IDF feature vectors for the texts as a sequence of word vectors.
2. GloVe Vector features: GloVe vector embeddings for the texts as a sequence of word embeddings.
3. Humour label and score: Whether a text is humorous or not. If humorous what is its score in the range 0-1.(Garain, 2019)
4. Wordwise sentiment values: List of sentiment values of each word of the text.
5. Hate and offensiveness labels: Whether the text is offensive or not and if it constitutes hate speech.
6. Frequency of easy and difficult words: Included as a semantic feature for the texts. (Basu et al., 2019)

3.2 Learning Model

Grid search refers to the practice of tuning hyperparameters to determine the most optimal values for a given model. This has a massive significance as the performance of the entire model is highly dependent on the hyperparameter values specified.

The estimator parameter of the Grid Search Cross-Validation process requires the model that has been used for the hyperparameter tuning process. Here the model used is the linear and the RBF kernels of the estimator Support Vector Regression model (SVR).

This process requires certain parameters to be taken as manual input. The param_grid parameter itself in turn requires a list of parameters and the range of values for each parameter of the specified estimator. The flow diagram has been shown in Fig 1(Wang et al., 2019):

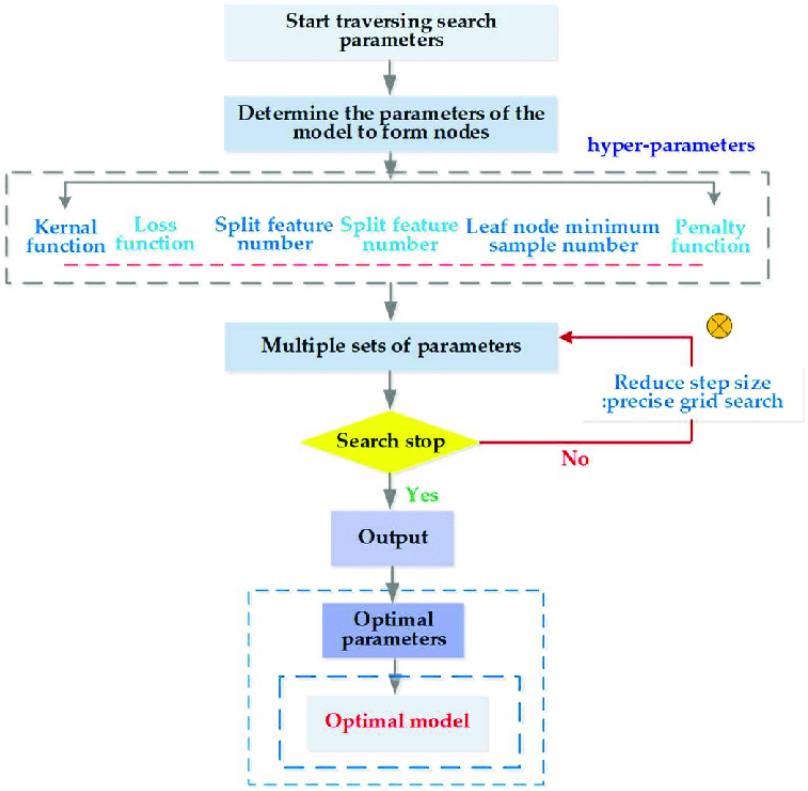

Figure 1: Grid Search parameter optimization overview

The SVR was fed with parameter values of

- C=best_params["C"]
- gamma=best_params["gamma"]
- coef0=0.1
- kernel=['linear','rbf']
- probability=False
- shrinking=True
- verbose=False
- epsilon=best_params["epsilon"]
- cache_size=200
- decision_function_shape='ovr'
- max_iter=-1
- random_state=None
- tol=.001

Class weight and degree were set to Ellipsis.

The most significant parameters required when working with the RBF kernel of the SVR model were "c", "gamma" and "epsilon". A list of values to choose from has been given to each hyperparameter of the model.

For the GridSearchCV algorithm, parameters like error_score, iid, param_grid, pre_dispatch, refit, return_train_score, scoring, and verbose were set to Ellipsis.

A cross validation process is performed in order to determine the hyper parameter value set which provides the best f1-score levels. The parameters for hyper-parameter selection are as follows:

- mean_fit_time
- mean_score_time
- mean_test_score
- mean_train_score
- param_C
- param_kernel
- params
- rank_test_score
- split0_test_score
- split0_train_score
- split1_test_score
- split1_train_score
- split2_test_score
- split_train_score
- std_fit_time
- std_score_time
- std_test_score
- std_train_score

Experimentation has been performed thoroughly and the parameters giving the best results have been accepted.

4 Results

The metric for evaluating the participating systems was as follows. The organizers used F1 averaged across the positives, negatives, and the neutral. The final ranking was based on the average F1 score. Our submitted system garnered an F1 score of 66.2%. The detailed results are shown in Table 1:

Class	Precision	Recall	F1-score	Support
negative	0.68	0.68	0.68	900
neutral	0.57	0.59	0.58	1100
positive	0.75	0.72	0.74	1000
Macro avg.	**0.66**	**0.66**	**0.662**	**3000**

Table 1: Class wise full result metrics

5 Conclusion

In the current work, we attempted to solve the problem of Sentiment Analysis of code-mixed English-Hindi data, while participating in the SemEval shared task. Our system was based on using traditional machine learning algorithms coupled with Beam Search Cross-Validation. Our system, when evaluated by the organizers garnered an F1 score of 0.662. There was an option of developing an unconstrained system, but we only used the provided data to develop the system. As future work, we would like to increase this data, use state-of-the-art Neural Network architectures on this data, taking into advantage the concept, matrix and embedded language, SentiWordNet, and other NLP features.

References

A. Basu, A. Garain, and S. K. Naskar. 2019. Word difficulty prediction using convolutional neural networks. In *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, pages 1109–1112.

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(Feb):281–305.

Avishek Garain and Arpan Basu. 2019a. The titans at semeval-2019 task 5: Detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 494–497.

Avishek Garain and Arpan Basu. 2019b. The titans at SemEval-2019 task 6: Offensive language identification, categorization and target identification. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 759–762, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

A. Garain, A. Basu, R. Dawn, and S. K. Naskar. 2019. Sentence simplification using syntactic parse trees. In *2019 4th International Conference on Information Systems and Computer Networks (ISCON)*, pages 672–676.

A. Garain, S. K. Mahata, and S. Dutta. 2020. Normalization of numeronyms using nlp techniques. In *2020 IEEE Calcutta Conference (CALCON)*, pages 7–9.

Avishek Garain. 2019. Humor analysis based on human annotation (haha)-2019: Humor analysis at tweet level using deep learning. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019). CEUR Workshop Proceedings, CEURWS, Bilbao, Spain (9 2019)*.

Avishek Garain. 2020. Garain at SemEval-2020 Task 12: Sequence based Deep Learning for Categorizing Offensive Language in Social Media. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*.

Sainik Kumar Mahata, Soumil Mandal, Dipankar Das, and Sivaji Bandyopadhyay. 2019. Code-mixed to monolingual translation framework. In *Proceedings of the 11th Forum for Information Retrieval Evaluation*, FIRE '19, page 30–35, New York, NY, USA. Association for Computing Machinery.

Sainik Kumar Mahata, Sushnat Makhija, Ayushi Agnihotri, and Dipankar Das. 2020. Analyzing code-switching rules for english–hindi code-mixed text. In Jyotsna Kumar Mandal and Debika Bhattacharya, editors, *Emerging Technology in Modelling and Graphics*, pages 137–145, Singapore. Springer Singapore.

Braja Gopal Patra, Dipankar Das, and Amitava Das. 2018. Sentiment analysis of code-mixed indian languages: An overview of sail_code-mixed shared task@ icon-2017. *arXiv preprint arXiv:1803.06745*.

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*.

Sainik Kumar Mahata Soumil Mandal and Dipankar Das. 2018. Preparing bengali-english code-mixed corpus for sentiment analysis of indian languages. In Kiyoaki Shirai, editor, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France, may. European Language Resources Association (ELRA).

Xiashuang Wang, Guanghong Gong, and Ni Li. 2019. Automated recognition of epileptic eeg states using a combination of symlet wavelet processing, gradient boosting machine, and grid search optimizer. *Sensors*, 19:219, 01.