IRLab_DAIICT at SemEval-2020 Task 9: Machine Learning and Deep Learning Methods for Sentiment Analysis of Code-Mixed Tweets

Apurva Parikh
DA-IICT
Gandhinagar, India
apurvakparikh
@gmail.com

Abhimanyu Singh Bisht
DA-IICT
Gandhinagar, India
bisht2492
@gmail.com

Prasenjit Majumder
DA-IICT
Gandhinagar, India
p_majumder
@daiict.ac.in

Abstract

The paper describes systems that our team IRLab_DAIICT employed for the shared task Sentiment Analysis for Code-Mixed Social Media Text in SemEval 2020. We conducted our experiments on a Hindi-English CodeMixed Tweet dataset which was annotated with sentiment labels. F1-score was the official evaluation metric and our best approach, an ensemble of Logistic Regression, Random Forest and BERT, achieved an F1-score of 0.693.

1 Introduction

The past few years has seen a massive surge in the number of people using social media sites such Twitter, Facebook, etc., in the Indian subcontinent. A large faction of these Netizens do not always use Unicode to voice their opinions on social media platforms. Instead, they resort to roman script/transliteration and frequent insertion of English words or phrases through code-mixing, with many using a mix of multiple languages. This points towards a growing need for technologies that can handle such code-mixed text, especially in India which is home to a plethora of languages. Sentiment analysis being a core NLP feature, especially for social media analysis, makes the development of systems capable of performing accurate sentiment analysis on code-mixed data an immediate necessity.

The SentMix (Patwa et al., 2020) task was organised for Hindi-English and Spanish-English code-mixed data. The data comprised of tweets which had been annotated with token-level language labels and sentence-level sentiment labels. Our team conducted experiments on the Hindi-English dataset.

Our experiments followed two approaches, the first involved feature extraction using TF-IDF and using Logistic Regression or Random Forests to classify tweets as positive, negative or neutral. The second involved transfer learning using BERT. We tried the this approach because BERT utilizes a sub-word vocabulary and hence, can handle OOV words allowing us to use a BERT (Devlin et al., 2019) pre-trained on English data for Hindi-English code-mixed data.

The rest of paper is organized as follows. In Section 2 we discuss related work. In Section 3 we briefly discuss the dataset which was provided. Proposed methods and obtained results are presented in Section 4. Finally we conclude our work and future works in Section 5.

2 Related Work

Given the rise to prominence of social media platforms like Facebook and Twitter in the past few years, sentiment analysis of social media posts has become a popular avenue for research. Sentiment analysis of code-mixed data is considered a difficult task as the data is quite noisy and there is a lack of large volumes of annotated data. A popular shared task which focuses on sentiment analysis of tweets is the Sentiment Analysis for Indian languages (SAIL). SAIL 2015 focused on performing sentiment analysis on tweets in three Indian languages: Bengali, Hindi, and Tamil (Patra et al., 2015). SAIL-2017 had two code-mixed datasets Hindi-English and Bengali-English for developing sentiment analysis systems (Patra et al., 2018).

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1265–1269
Barcelona, Spain (Online), December 12, 2020.

K. Ravi and V. Ravi (2016) combined TF-IDF with gain ratio based feature selection and RBF Neural Network for sentiment analysis on Hindi-English code-mixed data. Joshi et al. (2016) generated sub-word representations of Hindi-English code-mixed data collected from Facebook using character embeddings and 1-D convolution. The sub-word representations were then passed to LSTM layers for sentiment classification. K. Rajput et. al. (2020) investigated the application of transfer learning for hate speech detection on code-mixed data. They trained CNN-based neural models on tweets in a chosen primary language, followed by fine-tuning on a smaller dataset which had been transliterated to the primary language.

3 Dataset

The organizers provided a dataset containing Hindi-English code-mixed tweets. Table 1 shows the statistics of the provided dataset. Each tweet had a unique UID and a sentiment label which could positive, negative or neutral. Each token in a tweet had a lang_id denoting the token's language. If a token is a Hindi word then the lang_id is 'Hin', if it is an English word then the lang_id is 'Eng' and the lang_id is 'O' if it is neither English nor Hindi. An example of Hindi-English code mixed tweet and its token wise tagged text is as shown below.

tweet: "@Nationalist_Om @sonu86844114 Bilkul Theek Bole Bhai"

tokens	@	Nationalist	_	Om	@	sonu86844114	Bilkul	Theek	Bole	Bhai
lang_id	O	Eng	O	Eng	O	Eng	Eng	Hin	Hin	Hin

The dataset had a few inconsistencies with respect to annotation. In certain cases the lang_id given for a token was incorrect. Emoticons were tagged as 'O' and in some cases and as 'EMT'. This prompted us to not use the lang_id annotation in our experiments. Some of the tweets were not English-Hindi code-mixed. A few examples have been listed below,

Non-Hinglish Tweet (meta:4330): "nen á vist bolest vztek smutek zmatek osam ě lost beznad ě j a nakonec jen klid Asi takhle vypad á m ů j life ..."

tokens	nen	á	vist	bolest	vztek	smutek	zmatek	osam	ě	lost	beznad	ě	j	a	nakonec	jen	klid	Asi	takhle	vypad	á	m	ů	j	life	...
lang_id	Eng	O	Eng	Eng	Eng	Eng	Hin	Hin	O	Eng	Eng	O	Hin	Eng	Eng	Hin	Hin	Hin	Hin	Hin	O	Hin	O	Eng	Eng	O

Wrong Language ID Annotation (meta:15893): "EVRYONE SHIT THE FUCK UP https//t.co/RIN0mqkPsZ"

tokens	EVRYONE	SHIT	THE	FUCK	UP	https	//	t	.	co	/	RINOmqkPsZ
lang_id	Hin	Hin	Hin	Hin	Eng	Eng	O	Eng	O	Eng	O	Hin

Different tags used for tagging Emojis (meta:11381): "RT @Ambar_Aum @shoaib100mph Na naa your boys played really well!😜 This meme explains it all 😉 #PAKvWI #WIvPAK https//t.co/RcoKLF01Ny"

tokens	RT	@	Ambar	_	Aum	@	shoaib100mph	Na	naa	your	boys	played	really	well	!	😜	This	meme	explains	it	all	😉	...
lang_id	Eng	O	Hin	O	Eng	O	Eng	Eng	Hin	Hin	Eng	Eng	Eng	Eng	O	O	Eng	Eng	Eng	Eng	Eng	EMT	...

Table 1: Dataset Statistics

Details	**Train**	**Validation**	**Test**
Data points	14000	3000	3000
Neutral	5264	1128	1100
Positive	4634	982	1000
Negative	4102	890	900

4 Experiments and Results

4.1 Text Preprocessing

As social media data contains a lot of noise text preprocessing needs to be done so as to aid feature extraction. The text pre-processing performed on the data were as follows,

- The words in a hashtag were retained because the tags are unique and informative, therefore we felt that they may help with categorization.
- User mentions i.e text followed by @ was removed as it don't contribute towards sentiment of sentence.
- All punctuation, numbers, URLs and emojis were removed.
- Data also had few single characters that were also removed.
- Some tweets were retweets so it had rt as starting token, it was also removed.
- Stopwords were removed for some methods with the help of Python package NLTK[1].
- Text was converted to lowercase.

4.2 Sentiment Analysis

The task was a ternary classification problem to predict if the sentiment of a given tweet is positive, neutral or negative. Below are methods used for this task.

1. **TF-IDF with Logistic Regression:** For this method we generated the TF-IDF representation for each tweet using the Sklearn[2] Python library. We used the obtained TF-IDF representations as features for performing Logistic Regression using Sklearn library. Training was done using default hyper-parameters.
2. **TF-IDF with Random Forest:** For this method we generated the TF-IDF representation for each tweet using Sklearn. For the obtained features we applied an ensemble-based classifier Random Forest using Sklearn library. Training was done using default hyper-parameters.
3. **BERT:** As BERT utilizes sub-word embeddings it is capable of handling Out-of-Vocabulary words. We fine-tuned a bert-base-uncased for text classification from the HuggingFace[3] transformer library on the given dataset for 4 epochs. **Maximum number of tokens was set to 64** and **the batch size was 32**. For this model stopwords were not removed as BERT generates contextual representations through self-attention and maintaining the original structure of the input text aids performance. We used AdamW optimizer (HuggingFace) with default parameters and a linear learning rate schedule with warm-up for training.
4. **Ensemble:** We created an ensemble model by doing a majority vote decoding for all three models defined above. This was done so as to compensate for the shortcomings of the individual models and boost their classification capabilities.

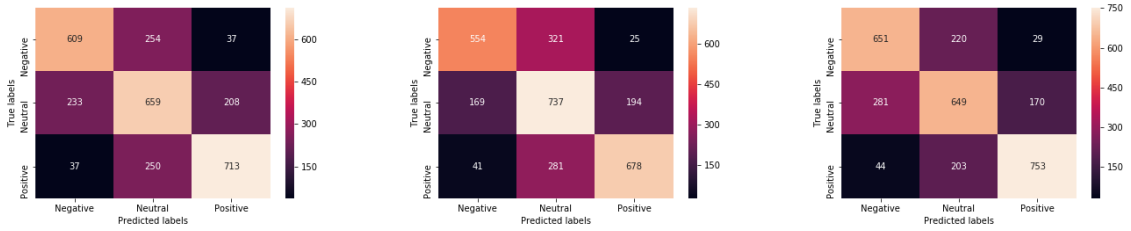
For ranking organizers were going to use only first three submission submitted by a team, so our first three approaches were considered while creating leader-board. Primary metric for evaluation was averaged F1-scores across all the three classes. Table 2 shows results obtained by our methods on the test set. Our best approach i.e BERT achieved 13^{th} rank (CodaLab ID: apurva19) out of 62 teams who submitted predictions.

Figure 1 shows the heat-map of first three predictions by our models that organizers considered for ranking. An observation for it is that statistical based feature extracting using TD-IDF was able to better predict "Neutral" class than BERT, for "Positive" and "Negative" class BERT gave better predictions.

[1] https://www.nltk.org/
[2] https://scikit-learn.org/
[3] https://huggingface.co/

Table 2: Obtained F1 Score on test set

Method	Neutral	Positive	Negative	Overall
TF-IDF and Logistic Regression	0.58	0.73	0.68	0.66
TF-IDF and Random Forest	0.60	0.71	0.67	0.66
BERT	0.60	**0.77**	0.69	0.68
Ensemble	**0.63**	0.76	**0.70**	**0.69**



(a) TF-IDF and Logistic Regression (b) TF-IDF and Random Forest (c) BERT

Figure 1: Heat map of our first three predictions used for ranking

5 Conclusion

In this paper we presented methods used to solve the SentMix task organised by SemEval 2020 and our results for the task. We utilized feature extraction using TF-IDF and machine learning techniques like Logistic Regression/ Random Forest but these methods were outperformed by our transfer learning approach which used an English language BERT. A possible cause can be the vocabulary of the corpus. It is quite common for Hindi-English code-mixed data to have different spellings (tweets in general contain a lot of spelling errors and internet jargon) for the same Hindi word which leads to a higher frequency of Out-of-Vocabulary (OOV) words. Since transformer-based language models, like BERT, employ subword embeddings these OOV words can be handled without much overhead whereas the same is not possible for TF-IDF. In future we would like to improve our results by including emoticons and employing code-mixed data specific preprocessing techniques.

Acknowledgements

We would like to thank the SentiMix 2020 shared task organizers for organizing this interesting shared task and for promptly replying to all our inquiries. We further thank the anonymous reviewers for their feedback.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Aditya Joshi, Ameya Prabhu, Manish Shrivastava, and Vasudeva Varma. 2016. 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*, pages 2482–2491.

Braja Gopal Patra, Dipankar Das, Amitava Das, and Rajendra Prasath. 2015. Shared task on sentiment analysis in indian languages (sail) tweets - an overview. In Rajendra Prasath, Anil Kumar Vuppala, and T. Kathirvalavakumar, editors, *Mining Intelligence and Knowledge Exploration*, pages 650–655, Cham. Springer International Publishing.

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.

Kshitij Rajput, Raghav Kapoor, Puneet Mathur, Hitkul, Ponnurangam Kumaraguru, and Rajiv Ratn Shah, 2020. *Transfer Learning for Detecting Hateful Sentiments in Code Switched Language*, pages 159–192. Springer Singapore, Singapore.

Kumar Ravi and Vadlamani Ravi. 2016. Sentiment classification of hinglish text. In *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, pages 641–645. IEEE.