Sentimiento Analysis: Classifying Affect in Mixed-Language Tweets

Connor Boyle
boyle128@uw.edu

Martin Horst
mmhorst@uw.edu

Nikitas Tampakis
tampakis@uw.edu

1 Introduction

Although a large percentage of the world's population is multilingual, most natural language processing (NLP) tasks focus on monolingual contexts. Our project aims to investigate how state-of-the-art machine learning approaches to sentiment analysis perform on code-mixed (i.e. multilingual) sentences. Speakers of Spanglish and Hinglish (Spanish-English code mixing and Hindi-English code-mixing respectively) naturally blend lexica and syntax across languages. Can language models from each language be combined to correctly interpret affect in tweets? Do features like emojis, punctuation, and orthographic variation also play a role in the emotion expressed in a tweet? We demonstrate that although accuracy of sentiment classification can be improved by leveraging state-of-the-art methods, there is still substantial improvement to be made in this NLP task.

2 Task Description

Our primary task consists of classifying code-mixed Spanish-English social media (Twitter) posts with sentiment labels, which was part of SemEval 2020's Task #9: SentiMix (Patwa et al., 2020). Our adaptation task will be the analogous Hindi-English sub-task of the same SemEval 2020 task. One interesting statistic from the SemEval 2020 overview paper is that there were over twice as many submissions for Hinglish as there were for Spanglish.

The training and testing data sets of each language sub-task are both comprised of tokenized mixed-language tweets, with token-level language labels and tweet-level sentiment labels. The sentiment labels are each one of three categories: "positive", "neutral", or "negative".

Submitted classifier models were evaluated based on their multi-class support-weighted F1 score on held-out evaluation ("test") data.

The fully-labeled Hinglish training and test data, as well as fully-labeled Spanglish training (but no test) data are publicly available (on the competition website) and we have downloaded them. However, the labels are missing for the Spanglish test records. We determined that the Spanglish test data is mostly a reshuffling of other publicly-available test data (Aguilar et al., 2020). We were able to match at least 2,200 out of 3,718 unlabelled SentiMix test data records to labels in the other publicly-available test data set. We also reached out to the original organizers, who informed us that we could submit to a new Spanglish competition on a different platform. The test set on this new platform has its labels withheld, however, many of the test instances overlap with our training data (therefore it does not have the expected guarantee against overfitting).

3 System Overview

In the first iteration of our affect recognition system (D2), we trained an SVM using a bag-of-ngrams ($n \in \{1, 2, 3\}$) representation of the input tweets. We did not change the tokenization from the provided tokenization. This system achieved a weighted F1 score of 50.41% on the Spanglish "dev" dataset.

For our improved system architecture (D3) we fine-tuned an instance of the Bidirectional Encoder Representations from Transformers (BERT) model presented in (Devlin et al., 2018). Our model is based on `bert-base-multilingual-cased` ("M-BERT") which was trained on a masked-language modeling (MLM) (Cloze) task on content in over 100 different languages. It can be adapted to sequence classification by replacing the MLM output model with a single classification layer. As the main BERT architecture is extensively covered in such papers as (Vaswani et al., 2017), (Kaiser

et al., 2017), and (Munikar et al., 2019), we will discuss only our changes to the model for this multi-class classification task as they relate to the idiosyncrasies of the shared task.

4 Approach

To fine-tune the BERT model, we had to prepare and format our training data in order to be compatible with the token vocabulary used in the base Multilingual BERT model. We first reconstituted each provided training instance (provided in tokenized, language-tagged form) into a single string sequence representing the entire (original) tweet, with the language tags discarded, and then passed into the the provided tokenizer for `bert-base-multilingual-cased` from the Transformers library (Wolf et al., 2020). The model token vocabulary identified tokens spanning entire words and also at the subword level (single and multiple characters).

We set our maximum sequence length to 140 tokens–significantly longer than the 40 tokens shown in the shared task baseline (Patwa et al., 2020), which significantly truncated their training dataset. We decided on this number by analyzing the tweets in our "training" and "dev" sets to discover the maximum length of tokenized tweets. We found that no tweet in the provided data set exceeded than 130 tokens, we chose a length of 140 and padded shorter tweets to that length.

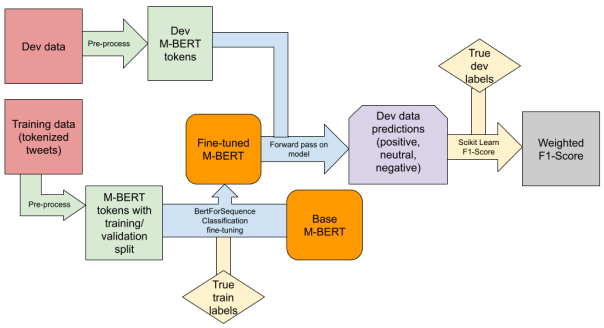

Figure 1: System architecture diagram for D3

We followed a tutorial by (McCormick and Ryan, 2019) in order to produce a fine-tuned model capable of classifying our code-mixed tweets. The principle differences for our system were our base model (we used the cased `bert-base-multilingual-cased` model (Devlin et al., 2018)) and the training data (code-mixed tweets instead of grammaticality judgements on full English sentences). We used Google Colab to train our model, whose free computational resources were a boon to our development process. For example, a model trained using a (free) GPU for 4 epochs finished training in under 20 minutes, compared to an estimated 6 hours on a local, high-end CPU.

5 Results

Results for our classifiers are given in table 1 below and are for Spanglish only. We report two of our own baseline systems that use no features to predict the sentiment of a given tweet. The first system assigns a random sentiment classification. The second system a positive (i.e. most common class) sentiment classification to every instance. The weighted F1 score from our first iteration's model (D2, bag-of-ngrams SVM) is reported in the fourth row and surpasses our two baselines. The score from our second iteration's model (D3, fine-tuned M-BERT) is reported in the fifth row, and is slightly lower than our first iteration. The below table also includes the shared task baseline, which was also created by fine-tuning multilingual BERT.

System	Precision	Recall	F1
Random	0.154	0.455	0.403
All Positive	0.167	0.333	0.333
Task Baseline	??	??	0.656
SVM	0.387	0.444	**0.508**
Tuned M-BERT	0.475	0.452	**0.504**

Table 1: Averaged results from five system trials

6 Discussion

Compared to our initial system design, the architecture we present in Section 3 was much more sophisticated, having been trained originally on the top 102 languages of Wikipedia, and uses a shared vocabulary to create bidirectional encoded representations for language input. However, our fine-tuned BERT model for Spanglish that we trained for this deliverable (weighted F1 = 0.50) did not remotely approach the performance of the shared task's baseline (weighted F1 = 0.66) or even match our previous SVM model (weighted F1 = 0.51). This is despite the fact that we specifically tried to copy the hyperparameters (specifically, learning rate and optimizer) described in the shared task paper. We also tried limiting our maximum sequence length to 40 (Bert WordPiece tokens) as the task organizers did–thus significantly truncating most

of our input. This did not lead to any performance gain.

In anticipation of our adaptation task of code-mixed Hindi-English ("Hinglish"), we also tried fine-tuning BERT on the Hinglish training data. Perplexingly, our BERT instance *does* achieve nearly the same level of performance (weighted F1 = 0.62) as the baseline described in the SemEval shared task paper (weighted F1 = 0.65), even without hyperparameter tuning.

We entertained a few reasons for why our fine-tuned BERT failed to meet expectations:

A: Irrelevant tokens
Twitter data is fraught with extra-linguistic tokens which may impact the fine-tuning approach that we took. Although our dataset was tagged to identify tokens that were not language data (in either English or Spanglish), our dataset was not cleaned of these non-language tokens (e.g. URLs, retweets, or "@" user mentions), and thus the performance of the model could have been below the expected mark, as in (Chiorrini et al., 2021).

B: Maximum length of BERT tokens
One hyperparameter necessary to train M-BERT is the maximum length of tokens per sequence. In the shared task's baseline system, the authors truncate all tweets to a max of 40 sub word tokens. With the intention of giving the model the most information possible, we did not truncate any tweet in our training set. However, due to the unpredictable nature of the tweet data, the model could have learned erroneous correlations, which may decrease the overall F1 score.

C: Language tags Another difference between our system and the shared task's baseline centers around using metadata provided with the tweets themselves. Our second model iteration (D3, fine-tuned M-BERT) did not incorporate this data into its learning or predictions. It is not clear if the original baseline incorporated these metadata; this is another feature that could have informed the prediction of sentiment classes. This might also explain why our preliminary attempt at a Hinglish model achieved performance near to the advertised baseline performance, but our Spanglish model(s) did not. the Hinglish dataset was in fact annotated using an automated tool (Patwa et al., 2020); these language tags are not of particularly high accuracy, and likely do not do much to inform BERT (compared to the human-created Spanglish language tags).

7 Conclusion

We are somewhat disappointed to not be able to replicate the scores of the baseline approach in the overview paper. However, we look forward to seeing how our system performs on the Hinglish dataset and we continue to further explore how language/data-aware and agnostic approaches can affect the model accuracy.

References

Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. 2020. Lince: A centralized benchmark for linguistic code-switching evaluation.

Andrea Chiorrini, Claudia Diamantini, Alex Mircoli, and Domenico Potena. 2021. Emotion and sentiment analysis of tweets using bert.

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*.

Lukasz Kaiser, Aidan N Gomez, Noam Shazeer, Ashish Vaswani, Niki Parmar, Llion Jones, and Jakob Uszkoreit. 2017. One model to learn them all. *arXiv preprint arXiv:1706.05137*.

Manish Munikar, Sushil Shakya, and Aakash Shrestha. 2019. Fine-grained sentiment classification using bert. In *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, volume 1, pages 1–5. IEEE.

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. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.