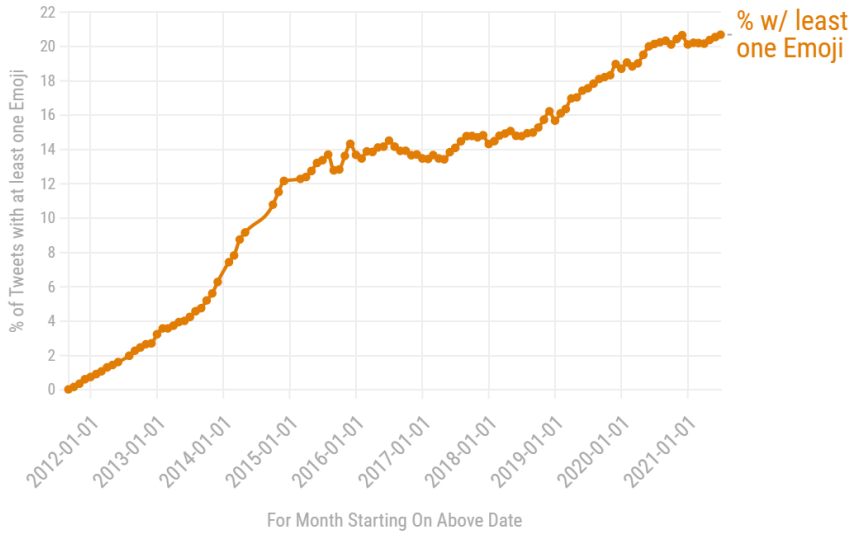
BERTm😎ji

LIGN 167

Jacky Lin, Andrew Zhou

Introduction

With the ever-growing age of technology, emojis have become more and more prevalent in social media. According to data from EmojiPedia, one in every five tweets in 2021 contain at least one emoji, up from one in six tweets in 2019.



Source: Emojipedia analysis of over 6.5 billion global tweets dated between 2011-09 and 2021-07 [Published July 2021]

Emojis, a representation of emotions, are often ignored in natural language processing models due to a mix of them being relatively new, a lack of support, and a lack of corpora with sufficient and meaningful emoji data. In this project, we aim to add support for and include emojis in the standard BERT model as we believe this will increase the performance of the model since emojis are often used to express emotions and should not be disregarded for emotion-related text classification — namely, sentiment analysis.

Model

In our model, we will analyze the performance with and without the inclusion of emojis during sentiment analysis using an existing popular model for sentiment analysis: BERT, which stands for Bidirectional Encoder Representations from Transformers. The BERT model used for the baseline comparison case is the pre-trained base-uncase model which consists of 12 layers and word embeddings of size 768. The tokenizer is pre-trained on a vocabulary of 30,522 words with special tokens such as [CLS] which is inserted at the beginning of each input and the last hidden state of this [CLS] token (12th layer) is the only token used for all classification tasks. There is also the [SEP] token which separates the sequences and the [UNK] token for unknown inputs outside of the known vocabulary of size 30,522.

To tackle our problem, we expand the known vocabulary by adding unique tokens for each emoji such that each unique emoji has its own unique token instead of the [UNK] token and its corresponding embedding being used for every emoji. We also takes this approach to directly tokenize the emojis uniquely instead of converting them to the widely used shortcode representation using colons (eg. 🙄 to :face_with_rolling_eyes:) to avoid the model and tokenizer to use the existing vocabulary tokenizations of the shortcode representation (eg. [face], [#with], [#rolling], [#eyes]). Ultimately, this prevents emojis from being classified with the unknown token as we hypothesize that there is valuable sentimental data in emojis and that they should be treated separately than truly unknown words. This also separates the different types of emojis into different tokens as happy emojis and heart emojis should be interpreted far more positively than angry emojis or sad emojis.

This results in the model taking in tweets as sequences of text and tokenizing each word and emojis with their respective tokens and appending the aforementioned special tokens appropriately. The token embedding, representing as a size 768 vector then goes through the 12 layers including the segment embedding, which differentiates between which sentence the word is from using the [SEP] token and the positional embeddings which specifies the position of the word/token in the sentence/sequence. Taking the final output layer of the [CLS] token, we can get the model weights as matrix of size 768 x 2 representing the size 768 vector for embeddings and the 2 classes (positive or negative sentiments). These weights are then fine-tuned for a specific classification class, in this case, sentiment analysis. Finally, taking softmax as the activation function for the model's final output logits results in a 2 x 1 matrix such that each value is the probability of its respective class in which we can simply take the class with the highest probability as our model's prediction.

Datasets

For our dataset, we use tweets that mention US airlines which can be found here: https://www.kaggle.com/crowdflower/twitter-airline-sentiment. The data is mostly labeled with a column denoting sentiment (positive or negative) as well as a confidence score in the label. For our task, we only take into account tweets with a sentiment confidence score of 1.0 meaning that the sentiment label for that tweet is 100% accurate. This still leaves most of the data to our disposal although the dataset to begin with is mostly imbalanced since people are more likely to tweet at an airline provider as a complaint. After all the cleaning, we end up with 8,897 data points with only 1,515 of them being positive and the other 7,382 being negative. The only necessary data here is the tweet itself which will be preprocessed and tokenized appropriately so

that the model can interpret it and the label. The preprocessing portion is done with code written by Orhan G. Yalçın as seen in https://towardsdatascience.com/sentiment-analysis-in-10-minutes-with-bert-and-hugging-face-294e8a04b671.

Results

The models were then fine-tuned for the task at hand using the Adam optimizer and various learning rates. The final results were that the default BERT model yielded an accuracy of 97% whereas the emoji model yielded an accuracy of 98%. This was done in batch sizes of 32 over 5 epochs with 50 steps per epoch. These values were chosen after tweaking the proportion of batch size, steps, and epochs according to the size of the data to avoid the model running out of data during the training process which occurred initially. The accuracy increased slightly each epoch and plateaued around the aforementioned values as well as the validation accuracies plateauing around the same amount.

Conclusion

Ultimately, when unique emojis are given their own unique tokens, the BERT model performs slightly better. This is as expected as emojis are usually indicative of emotion and disregarding them would be ignoring a potentially sentiment token. However, there is not a significant improvement from the base model and this may be due to the fact that the model without emoji support already performs very well and not every tweet contains an emoji for it to take into account and make a difference for. Although there exists neutral emojis such in which the model would learn to weigh less, there also exists emojis in which its sentiment can be either

positive or negative depending on the context as well as sarcastic emojis. This would require a more verbose dataset of emojis for the model to train on in order to catch. As such, future works may include training this model with a more balanced dataset as well as a greater number of emojis and varying emoji use cases as well as adding support for emoticons as well such as :), XD, >:(, etc.

References

Broni, K. (2021, July 13). *Emoji use at all-time high*. Emojipedia. Retrieved March 16, 2022, from https://blog.emojipedia.org/emoji-use-at-all-time-high/

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv, abs/1810.04805*.

Yalçın, O. G. (2021, February 2). *Sentiment analysis in 10 minutes with bert and hugging face*. Medium. Retrieved March 16, 2022, from https://towardsdatascience.com/sentiment-analysis-in-10-minutes-with-bert-and-hugging-face-294e8a04b671