ECE 1786

Creative Applications of Natural Language Processing

EMOJIMOTION Final Report

Project Name: Emojimotion

Group Members:
Ge Gao 1003803280,
Yuchen Yuan 1004549740

Word Count: 1995

Name	Permission to post video	Permission to post report	Permission to post source code
Yuchen Yuan	Yes	Yes	Yes
Ge Gao	Yes	Yes	Yes

Emojimotion

Carol Gao 1003803280 Yuchen Yuan 1004549740

***Abstract*—This paper would demonstrate how to predict emojis based on the input sentence and selected emotion using natural language processing. Training a dataset of tweets and emoji labels with baseline model and Bert model to predict emojis based on context meaning. Final emoji prediction is resulted by combining context scores with emotion scores.**

Keywords—NLP, Bert, Emoji

I. Introduction

Emoji originated from Japan in 1997 and it became widely used in texting, social media posts and webpages. We use it everyday when messaging to our friends and posting on social media. For the ios keyboard, it only predicts an emoji based on keywords instead of the meaning of the whole message. Also, different emojis with the same sentence will convey different meanings or ideas. For example, "first day at school" means the school day starts. By adding a , it means that I'm happy to go back to school and start the new term. By adding a , it means that the unwillingness to go back to school. Our project Emojimotion is an application to predict the emoji based on both the sentence's meaning and user's personal choice on eight different emotion types, including anger, anticipation, disgust, fear, joy, sadness, surprise and trust.

II. Illustration

As shown in Figure 1, Emojimotion contains two parts. The first part takes sentences as input, passes a neural network model (baseline vs. BERT), and outputs a list of top 15 emojis with the highest probability by sorting the model output logits. The second part of Emojimotion combines the model outputs and the emotion score of the user's choice of emotion and generates the final predictions. To combine the two numbers, the model output logits are passed through a softmax function and converted to context score. We extract the emotion score of each emoji in the top-15 list from the Emotag dataset and find the new emotion score by formula 1.

$$new\ emotion\ score_i = \frac{emotion\ score_i}{\sum_{j=1}^{15} \quad emotion\ score_j} \tag{1}$$

After that, equal weights are assigned to the two scores as in formula 2, and the application produces 3 emojis with highest possibility.

$$Final\ score_i = 0.5*context\ score + 0.5*new\ emotion\ score \tag{2}$$

As the second part is pure mathematical calculations, the models and results discussed in the following sections are related to part 1 only.

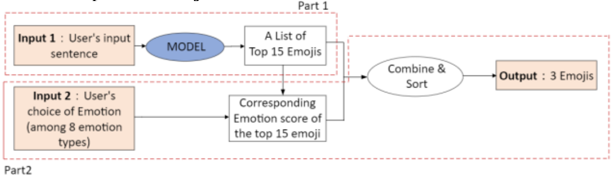

Figure 1: Flow chart of Emojimotion

III. Background & Related Work

As the rapid growth of emojis on social media, there are a lot of studies about emoji embeddings, sentiment analysis, and emoji predictions in recent years. We went through several papers and summarized 2 of which inspired us the most to create our EMOJIMOTION application.

In 2016, Eisner et al. [1] developed an emoji embedding model called Emoji2vec. It is a set of pre-trained embeddings of all Unicode emojis. Different from other pre-trained emoji vectors trained from a large number of tweets and Instagram posts, Emoji2vec is trained from the description of emojis. Besides training the word embeddings, this research also contains a simple sentiment analysis on tweets, which proves that the accuracy of the classification task can be improved by including the emoji in the sentences. The paper gave us our initial idea of exploring the emoji field.

In 2017, Francesco et al. [2] introduced Bi-LSTMs to predict the emojis. They compared the Bi-LSTM model with two baseline models: Bag of Words classifier and Skip-Gram Vector Average. The paper shows that the Bi-LSTM has the best performance among all the models in the experiment. The experiment result indicates that the frequent word is more likely to be mis-predicted as the model prefers to output the more frequent emoji which have similar meaning with some less frequent emoji. They also did a social experiment to let humans predict the emojis. Surprisingly, the Bi-LSTM achieved a higher accuracy.Abbreviations and Acronyms

IV. Data and Data processing

Firstly, we collected realtime Twemoji using ranking from emojitracker API [3]. 54 emojis are randomly selected to be labels in which 20 emojis are from top 50 used emojis and the rest are from top 50-150. Raw tweet dataset containing ID, time, text and emoji was collected by implementing Twitter API [4]. The search requirement is collecting 5000 English tweets for each emoji, the text length is more than 10 and containing at least one target emoji.

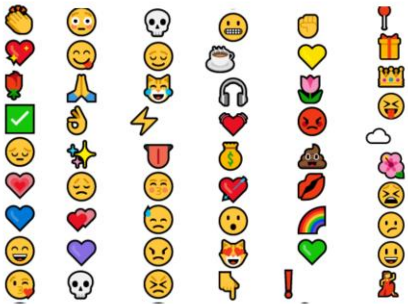

Figure 2: Selected 54 emojis

id	time	tweets	emoji
1593530060740386817	2022-11-18 09:02:27	Hahahaha......\nThe person cheated on is alwa...	
1593530057041002496	2022-11-18 09:02:26	@inScotland80 Tune like Texas x Lloyd Cole Com...	

Figure 3: Tweet Dataset Before Clean

TABLE I. DATASET FORMULATION

Dataset	# of Tweets
Raw	270K
Cleaned	108K
Train	69,120
Test	21,600
Valid	12,960

In the data cleaning stage, duplicated tweets are firstly removed from the raw dataset as some tweets are scraped down for multiple emojis. Noises in the sentence like hashtags, urls, mentions and unicodes are removed, because we think emoji should relate to the information conveyed by Tweet text only. Tweets after cleaning less than four words would be deleted since lower word count tweets contain more noise and we want to decrease the bias to model training. Figure 4 is one of the cleaned texts. We sampled 2000 cleaned tweets for each emoji to be final dataset and it is splitted into train/test/validation in the proportion of 0.64/0.2/0.16.

```
Before clean: [' @dlfjla A 你好story book scene.. congrats to99 both of you! 👏 ']
After clean: ['story book scene congrats to both of you']
```

Figure 4: Cleaned Tweet Text Example

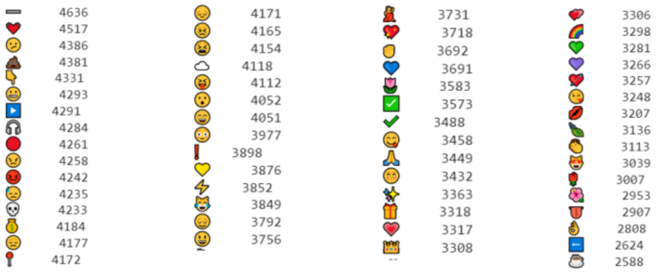

Figure 5: Cleaned Dataset

As shown in Figure 6, the open source EmoTag1200 database contains trained emotion scores for 1200 emojis under eight emotions ranging from 0 to 1. The symbolic emojis (ie.▶️) not shown in this dataset were assigned zeros for all the emotions. The new emotion dataset of 54 emojis is used to combine with the context score predicted from the model to make final emoji predictions based on the user's selection of emotion.

	emoji	emoji_num	anger	anticipation	disgus
0	👏	0	0.08	0.39	0.08
1	💔	1	0.0	0.33	0.0
2	🌹	2	0.0	0.36	0.0
3	✅	3	0.0	0.31	0.0
4	😞	4	0.25	0.22	0.31
5	💖	5	0.0	0.36	0.0
6	⬅️	6	0.0	0.0	0.0
7	💙	7	0.0	0.25	0.0
8	😄	8	0.0	0.36	0.0
9	😋	9	0.0	0.33	0.0

Figure 6:Emoji Emotion Score Dataset Example

V. ARCHITECTURE AND SOFTWARE

The final neural network model architecture is BERT. BERT has demonstrated its effectiveness in natural language processing tasks, such as text classification and question answering[6]. Pretrained BERT is a transformer based architecture which learns the contextual information using a multi-layer bidirectional transformer encode. As in Figure7, BertTokenizer is used to tokenize input tweets, and two special tokens [CLS] and [SEP] are added to the original input tokens which represent start and end of sentence respectively. The embedding input enters the BERT Model and classifier head accordingly which is stacked on top of the model.

We fine-tuned the bert-base-uncased that consists of 12 layers and 110M parameters with a classifier head by freezing the first 6 layers of the BERT model, training 5 epochs, and setting the learning rate to be 5e-5. The performance of the model is evaluated by the top-8 accuracy.

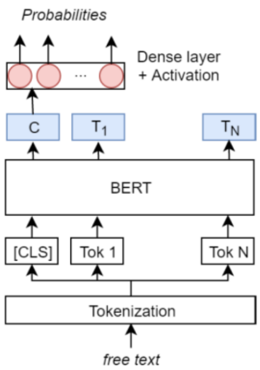

Figure 7: BERT architecture for multi-class classification

VI. BASELINE MODEL

The baseline model firstly converts the cleaned tweet text into vectors by pre-trained 840B300d GloVe embedding. The word embedding average of a sentence is passed through the forward propagation. Cross entropy loss function is computed to compare the model outputs and the one-hot encoded vector emoji label. The hyperparameters are chosen as shown in Table 2. When evaluating the accuracy, the output of the model is passed to the softmax function to produce a probability vector, which is also considered as context score. Prediction is marked as correct if the true label is among the top-8 predicted emojis. Only top-15 most probable emojis would be considered in part 2 because we believe the final predicted emoji should firstly match the context.

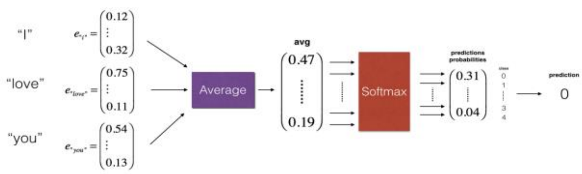

Figure 8: Baseline Architecture

TABLE II. BASELINE ARCHITECTURE

Hyperparameter	Value
Optimizer	Adam
Learning Rate	0.0001
Number of Epochs	50
Batch Size	5

VII. QUANTITATIVE RESULTS

Both top 1 accuracy (conventional accuracy) and top 8 accuracy are used to evaluate the models. By using top 8 accuracy, a prediction will be considered as correct if the true label from our datasets exists in the top 8 possible predictions. As one sentence could have multiple meanings and the model contains 54 classes, using top 8 accuracy is more reasonable. Therefore, top-8 accuracy is used as major evaluation metrics to evaluate model performances. We also measured the result by human examination, which will be illustrated in the qualitative result section.

The following tables show the quantitative results of the models. For 54 classes, the highest accuracy for both top 1 and top 8 is achieved when freezing the first 6 layers. If we only used 30 classes, the top 1 accuracy does not have a significant increase compared with the 54 classes model. Also, the goal of Emojimotion is to provide users with a variety of selections and an appropriate model to predict emojis. Therefore, the final selected model is the BERT model with a 54-class classifier head.

TABLE III. 54 CLASSES(EMOJIS) MODEL RESULTS

Model		Top 1 Accuracy	Top 8 Accuracy
Baseline		3%	24%
BERT	Not freeze	23.9%	56.3%
	Freeze all layers - except the classifier head	18.6%	50.08%
	Freeze the first 6 layers	**24.6%**	**56.73%**

TABLE IV. 30 CLASSES(EMOJIS) MODEL RESULTS

BERT	Top 1 Accuracy	Top 8 Accuracy
Not freeze	3%	72%
Freeze all layers - except the classifier head	22.7%	66%

Epoch	Training Loss	Validation Loss	Accuracy
1	3.179800	3.136890	0.536285
2	2.919900	3.059883	0.563426
3	2.559300	3.107112	0.567303
4	2.133800	3.253362	0.561285
5	1.801100	3.361941	0.554167

Figure 9: Evaluation of the best Bert model

Besides the BERT Model, we also fine-tuned the smallest version of GPT2 which has 124M parameters. According to Figure 9 and 10, the BERT model with 56.3% accuracy outperforms the GPT2 model with 55% accuracy. Therefore, our main architecture is the BERT model and the following sections are based on BERT and baseline.

Epoch	Training Loss	Validation Loss	Accuracy
1	3.166800	3.139445	0.527720
2	2.864700	3.105480	0.550000
3	2.430900	3.219656	0.546296
4	2.002900	3.409909	0.544213
5	1.682100	3.537804	0.538137

Figure 10: Evaluation of the best GPT2 model

VIII. QUALITATIVE RESULTS

The following figures show results from the baseline model and the best BERT model. For the baseline model with “Let’s grab a cup of coffee after class” and joy as input, the output contains happy face emojis. With sadness as input, the output contains emojis with negative emotion. For the BERT model with the same input sentence and joy as emotion, it outputs a coffee emoji and 2 happy faces. With sadness as emotion, it outputs coffee, a skull and a flushed face. Results for both models make sense, while the BERT model results are more reasonable, as it reflects precisely the sentence context with key word coffee and other generated emojis are more appropriate. One of the reasons for Bert good performance is due to the BERT model selection. In the progress report we used bert-tiny and the training accuracy could only reach 23%. In comparison, bert-base-uncased performed much better as this model has richer parameters good for sentence classification.

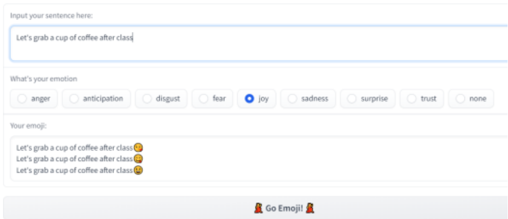

Figure 10: Baseline Model Result 1

Figure 11: Baseline Model Result 2

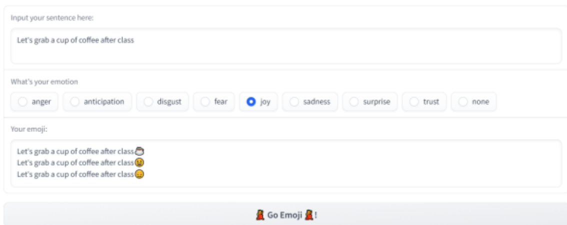

Figure 12: Best BERT Model result1

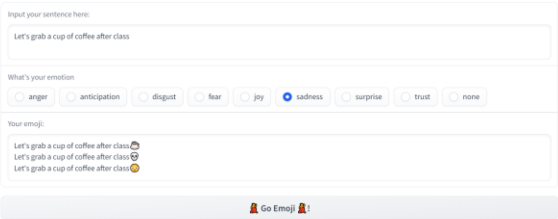

Figure 13: Best BERT Model result 2

As shown in Figure 14, for some input sentences, the top 3 predicted emojis without emotion selection would stay the same after choosing different emotions. It is because these kinds of sentences have strong emotional implication and the context meaning is obvious, therefore the high context score would outweigh the emotion score and lead to the same results.

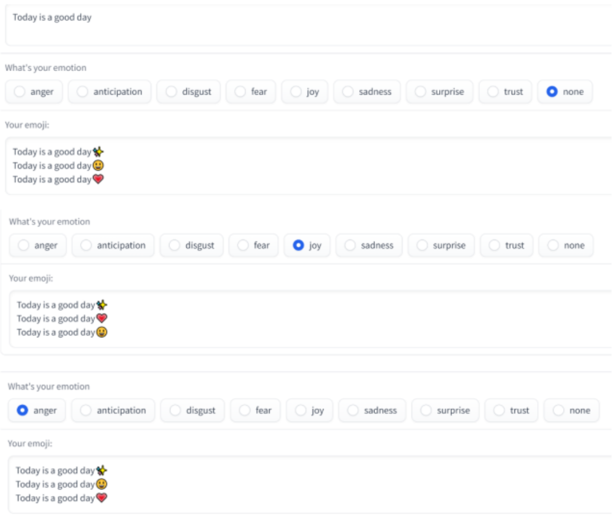

Figure 14: Example of Same Results for Different Emotion

IX. Discussion and Learning

In this project, according to the results listed in the previous sections, the baseline model top-8 accuracy(24%) is higher than we expected since the largest dimension of GloVe Embedding was used in baseline which increased word vector complexity, therefore words could be represented more accurately. Our fine-tuned Bert model reached outstanding top-8 accuracy (56%) which is very high among 54-class classification. One of the surprising findings is that our Bert model could accurately identify different K-pop groups as shown in figure 15. Each group has their own fan color(ie. BTS 💜, Super Junior 💙) and tweets posted by fans usually contain fan color of heart emoji.

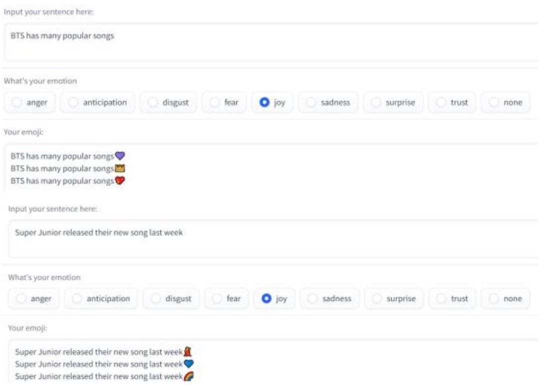

Figure 15: K-pop Groups Correct Fan Color Heart

The future improvements would include some manual work. Firstly, current emojis are selected by sample() method based on setting criteria, so similar emojis in one grouping might all be selected, for example, 💘 ❤️ 💞 💝 💗 from our emoji labels all represent red hearts and have similar meaning. The future emoji class should be selected more diversely, so the input sentence could match more appropriate emoji.

Secondly, some data in the cleaned dataset like shown in Figure14 are similar and repetitive. Most importantly, some words have no actual meanings and do not exist in English. These bad data are mostly ads which were usually posted at the same time, so according to the API timeline scraping, these tweets are collected for the same emoji class. Since they could not be filtered in the data processing, they would become the bias in the training and influence the testing accuracy. Therefore, this type of tweets require human sense to identify and delete in the future.

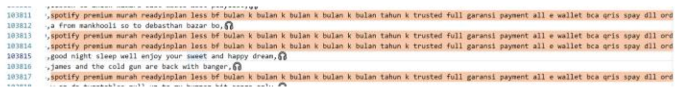
Figure 16 : Bad Data in Cleaned Dataset

Thirdly, although the purpose of Emojimotion is to predict emoji based on context and emotion, the current trained Tweet dataset only contains text and emoji labels, and we manually assigned weights to combine the context score and emotion score. Consequently, the future tweet dataset could be completed by adding an emotion label. We could get more reasonable and accurate weights of score combination through training to further increase accuracy.

X. Individual Contributions

We contributed equally to this project. Carol was responsible for scraping twitter data and building the baseline model. Yuchen was responsible for cleaning the raw dataset and building a Gradio interface. Both of us got involved in fine-tuning Bert model, training and testing the baseline model and Bert model.

References

[1] Eisner, B., Rocktäschel, T., Augenstein, I., Bosnjak, M., & Riedel, S. (2016). emoji2vec: Learning Emoji Representations from their Description. *Proceedings of the Fourth International Workshop on Natural Language Processing for Social Media.* https://doi.org/10.18653/v1/w16-6208

[2] Barbieri, F., Ballesteros, M., & Saggion, H. (2017). Are Emojis Predictable? *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*: Volume 2, Short Papers. https://doi.org/10.18653/v1/e17-2017

[3] Emojitracker. (n.d.). Emojitracker/emojitrack-rest-API: Emojitracker Rest Api. *GitHub*. Retrieved December 12, 2022, from https://github.com/emojitracker/emojitrack-rest-api

[4] How to get Tweets using Python and Twitter API. (2022, July 11). *Quantitative Finance & Algo Trading Blog by QuantInsti.*

[5] GitHub - abushoeb/EmoTag: Emoji-Emotion dataset: An emoji-centric NLP resources based on Twitter Data. (n.d.). *GitHub.* Retrieved October 31, 2022

[6] Varsheni R, S. (2021, June 29). Why and how to use BERT for NLP Text Classification? Analytics Vidhya.