ECE324 Final Report: Emoji Predictor

2023.04.13

Jim Yang | 1006913761
Mary Cheng | 1006675714
Yimiao Yuan | 1007050227

Abstract

As people spend more time on the use of social media platforms to express themselves and their feelings, the need to convey emotions, feelings, and thoughts clearly on the Internet (eg. messages, posts, and comments) has led to increasing usage of emojis. However, people who cannot choose the appropriate emoji to go along with their messages or texts, for example, people with autism or dyslexia, may experience difficulties properly expressing themselves.

This report proposes a neural network model that takes in an input sentence and outputs 3 emojis that are possibly a good representation of the sentence's meaning. The proposed model architecture consists of a pre-trained BERT model with linear and dropout layers. Our approach reaches an accuracy of 37.61% for targets within the top 3 predicted emoji. The proposed model differs from the Baseline model (accuracy of 27.14%) by approximately 10% in accuracy.

1. Introduction

The usage of emojis has increased as people spend more time expressing themselves and sharing their opinions on social media platforms. Emojis have become significant in people's conversations online as they can deliver a visual representation of emotions. Nowadays, there are approximately 4000 available emojis on a typical mobile device. This has presented a challenge for people to quickly and correctly choose the emotion that can properly express their feelings. Therefore, this project is to address the question of how to use machine learning to build a model that predicts the most suitable emoji to go along the given block of text. The model will output three top predicted emojis for the users to choose from.

The final report will highlight the work that has been done in the past months. It will be divided into five sections, which are Related Work, Methodology, Results & Discussion, Broader Impact & Ethical Implications, and Conclusion. The report first summarizes the related work, followed by a detailed description of the methodology, including data collection, pre-processing, processing and model implementation of the BERT model and a Baseline model. Then the report provides the results of two models and discusses the comparison between the models. It will also analyze the limitations of the present work. Broader Impact & Ethical Implications are explained and finally we conclude the report.

Here is the Github Repository Link: https://github.com/marycx/ECE324-Project-Emoji-Predictor

2. Related Work

2.1 Paper I — *Federated Learning for Emoji Prediction in a Mobile Keyboard [1]*

2.1.1 Summary

The paper aims to reduce the latency of the auto-generation of emoji and to diversify the prediction of emoji output. The research team used a pre-trained Coupled-Input-and-Forget-Gate, a variation of Long-Short-Term Memory, to perform sentimental prediction. Furthermore, the probability of predicting frequently used emojis is high, regardless of the content. To overcome the issue, the probability of each emoji (P) is scaled by the empirical probability of the emoji (P): (**α** is a scaling factor that is determined empirically through experiments)

$$S_i = \frac{\widehat{P}(emoji=i|text)}{P(emoji=i)^{\alpha}} \qquad \text{(Eqn 1)}$$

2.1.2 Useful Incorporations

This paper provides useful insights in the effect of different emoji frequencies on the model accuracy. The processing data step should have ways to mitigate the effect of the different emoji frequency distribution. To incorporate this important finding in the team's final program, the number of unique emojis from collected data reduced from 1274 emojis to 50 emojis. This significantly reduced the possibility of overfit of the model and the reasoning is explained in the data collection section.

2.2 Paper II — *Context-Aware Emoji Prediction Using Deep Learning [2]*

2.2.1 Summary

The paper aims to personalize the auto-generation of emoji by analyzing the time and location of the conversation on existing models, such as the cousin and Lstm algorithms. The dataset is separated into 15 sentiment classes. Each data input also includes the time, location, and text messages.

2.2.2 Useful Considerations

This paper states that LSTM is a possible choice. Furthermore, the main takeaway in this paper is that they added location and time as input features to improve accuracy. Many papers used text as the only feature. Thus, it gives us the idea that the project needs to pay attention to the information that is kept inside the input data, so that the model has the necessary information to make better predictions.

2.3 Paper III — *Using BERT for Emoji Prediction [3]*

2.3.1 Summary

The third paper used a pre-trained BERT model to predict emoji. This paper introduced the idea of Multi-label Classification, given d and E, predict whether each $e \in E$ is properly connected to d. Different from the multi-class classification setting, given $d \in D$ to predict the $e \in E$ which best associates with d, the probability score of each emoji is independent from the rest of the emojis under the multi-label classification setting. Then, emojis with low probability scores are eliminated from the final predictions. To predict emojis based on input texts, the authors fine tuned a pre-trained BERT model on the dataset to adapt BERT model to the social media domain.

2.3.2 Useful Considerations

This paper suggested using the newly developed BERT model. The BERT model is significantly different from the Lstm model mentioned before, as it uses attention and transformers to build its architecture. After careful consideration, the BERT model was chosen as the model to implement in predicting emojis. Furthermore, the team currently uses the Multi-class classification. In the future, the team will research more about the Multi-label classification trying to improve the accuracy of the current model.

2.4 Analysis

Both BERT and LSTM are still used currently for text classification. The BERT model is good for large corpus training as it is not sequential. LSTM on the other hand, is more suitable for smaller corpus. Having an accurate context understanding of input texts is important in emoji prediction, since the context offers a clear semantic meaning of words within the corpus. Better semantic meaning equals better emoji prediction. Thus, BERT is superior in emoji prediction.

3. Methodology

This section describes the development and process of the emoji prediction system. There will be four subsections which are Data Collection, Data Pre-Processing, Data Processing, and Model Implementation & Training.

3.1 Data Collection

Data was collected using Twitter Scraper from Apify [4]. 161805 tweets in total were collected from 996 Twitter accounts. Those were then outputted as a CSV file with only one column that contains the content of the tweets. A sample data can be seen in Figure 1.

```
 ['Get 👟SNEAKY👟 for St Patty’s day 🍀 and for the rest of the year with the Skatter Bead available @Nordstrom! #shoesdaytuesday @kpcollections
https://t.co/loJGedhUSx'],
 ['Virgin Mobile Mod Club Toronto 2013 https://t.co/0V9edMtrCK'],
 ['Indian Premier League 2012 https://t.co/ehdV0O8ltA'],
 ['ProSieben show 2010 https://t.co/VJUAZsKtt5'],
 ['EMAs 2009 https://t.co/LSbKMsi7f3'],
 ['2009 Hello Katy tour https://t.co/9n9tKlF2Uv'],
 ['KatyCats, get ready to own a piece of my history! 👗 💃 FINALLY the Nostalgia Store powered by @Theta_Network is coming Friday 3/24, featuring
REAL stage-worn cutie costumes &amp; props from some of my past performances! 😍🙌\n\nA thread of some of the lewks here 🧵:'],
 ['It may be Oscar Sunday but don’t worry there’s still a lil taste of #idol tonight! Tune in after the awards show for a special 1 hour bonus ep
😘 (then see ya again at our usual place + time next week!) https://t.co/oHSN4oC0Fv'],
 [' 🔔 T-minus 1 month until Easter! 🔔 get ur 🐰 baskets ready with @kpcollections x @PEEPSBrand 😊 https://t.co/7Z69Px8m7z https://t.co
/LElzkHsjF3'],
 ['That feeling when the first styles from my Spring/Summer collection drops on @aboutyou_com 😍 The full collection is coming on March 29th. 🌷
```

Figure 1. The raw data scraped from twitter accounts

The data was collected from several celebrity accounts (from Github [5]) since it would maximize the validity of the dataset. Celebrities’ tweets are usually more formal and proper in language and emoji usage. Therefore, situations, such as sarcasm, can be largely avoided (for example, the sentence meaning is negative, yet the person puts a smiling face at the end of it). Moreover, the included celebrities are of different ages, sex, and racial groups to eliminate any bias or patterns of using emojis.

3.2 Data Pre-Process

Data pre-process was done primarily using Python modules, re and spacy. The pre-processed data file is used by both the BERT model and the baseline model.

During the first stage, the general invalid entries, such as entries without emoji and non-English, were removed to increase computational efficiency. Then, text and emoji were separated into two columns in each entry. The target emojis are encoded using integers. Afterwards, text length shorter than 3 words was removed, because the trade-off of learning from these short phrases is less than the information loss overall.

The next stage is text pre-processing. In order to have a higher accuracy of the model, more content should be retained in the input text. In the final version, input text contains no lemmatization and only non-space whitespace characters, such as newlines, tabs and carriage returns were deleted. The input data keeps the abbreviated words such as I’m, I’ve, and bday, since the BERT encoder can handle these words. Furthermore, hashtags and website links are kept to add more information to the input data. Figure 2 shows a sample of pre-processed text.

```
'To My Forever Valentine! I know you ain’t keen on this day, (Like a lot of ya’ll out there) BUT, I love it and that’s why you celebrate It, for m
e, which makes me adore you more! Just wait til you see what the day Has in store for you! I LOVE YOU MR.WEATHERSPOON! ❤️❤️🤎❤️❤️🤎🤎 https://
t.co/PHNSHiBFF3'
```

Figure 2: Pre-processed text

Initially, there were 1274 unique emojis within the data. However, many of the emojis only occurred a few times, thus making them outliers (Figure 3). These emojis with few occurrences may cause overfitting. For example, when the training dataset only contains one sample for a specific emoji, the model will only memorize the characteristics of the single sample input for the targeted output. It cannot learn the general pattern of predicting that targeted output, which consequently leads to overfitting in validation and testing. To improve the performance of the model and mitigate overfitting, only the 50 most frequent emojis were included in the dataset (Figure 4). The dataset was updated accordingly, keeping only the sentences that contain the chosen 50 emojis. The number of classes is chosen to be 50 due to the trade-off between eliminating as many classes with few samples and a total number of training samples. This results in 50 classes to use for the classification of emojis.

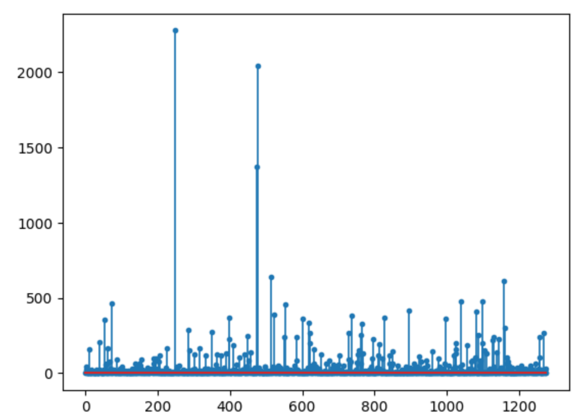

Figure 3. Frequency distribution with 1270 classes

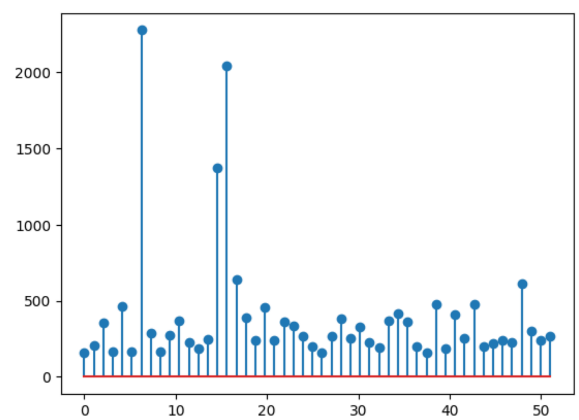

Figure 4. Frequency distribution with 50 classes

In the updated version, the CSV file contains 16200 data for model training, with one column of text with minimal preprocessing text as input and one column of emoji as targets. Each row is an independent sample that includes text in one column and target emoji in another. (Figure 5).

	TEXT	Emoji
0	Here to cure ur Sunday sads and bluezzz with #...	1
1	KatyCats, get ready to own a piece of my histo...	2
2	It may be Oscar Sunday but dont worry theres s...	3
3	T-minus 1 month until Easter! get ur baskets r...	4
4	That feeling when the first styles from my Spr...	2

Figure 5. Pre-processed data

3.3 Data Processing

3.3.1 BERT Model [9]

The feature of the current model only contains pre-processed text as inputs and encoded emojis as targets. The text in the data is tokenized with the help of BERT tokenizer functions. Then the tokenized inputs are transformed into torch tensors.

The data loading function returns four categories. Text is the first category. It is the torch tensor with the tokenized words. The second category is input IDs. It is the output of the BERT encoder, and it stores the ids for each input token. The third category is attention masks. It is also the output of the BERT encoder, and it tells the model token coming from the input sentence. The last category is target, and it contains torch tensors that hold the target emojis for each input text. The model takes 70 percent of data into training, 15 and 15 percent for validation and testing. It also uses a batch size of 16 for training.

3.3.2 Baseline Model

The baseline model takes in the average of all word embeddings in a sentence as input and encoded emoji as the target.

The collected Twitter data is first tokenized using nltk.sent_tokenize and nltk.word_tokenize. Then the word embeddings of all words are achieved using the word2vec model in the Gensim library [7]. Gensim word2vec model is first trained with our group's corpus data, with a vector size of 1000 and a window of 5. After training the word2vec model, two nested for-loops are implemented to loop through each sentence, then loop through each word in the sentence, to convert it into word embeddings. Eventually, the average of the word embeddings in each sentence is calculated using NumPy.

The model takes in averages of word embeddings and outputs encoded emoji. It takes 80 percent of data into training, 10 and 10 percent for validation and testing.

3.4 Model Implementation & Training

3.4.1 BERT Model

Model Architecture [6]

The processed inputs mentioned from the previous section are loaded into the pre-trained BERT-based-cased model. There is an added dropout layer with a dropout rate of 0.3 applied to the output of the BERT model. Then, a linear layer, acting as the classification layer, takes the output of the dropout layer and has an output dimension of 50. Lastly, the sigmoid layer produces the probability distribution of each emojis, which is built in the loss function. The model uses the standard cross entropy loss as the loss function and AdamW as the optimizer. To implement the L2 regularization with lambda = 1×10^{-4}AdamW optimizer has the additional parameter of weight_decay while the loss function is not modified.

Training

Adjusted from the previous model, there are 50 distinct emojis in the dataset. Thus, there are a total of 50 classes of emojis for the model to classify. The highest training accuracy is 62.04% at the tenth epoch. The model has an accuracy of 22.18% at the third epoch if the most probable emoji is the same as target emoji. The accuray for target within the top 3 predicted emoji is 38.19% at the second epoch.

Similar to the standard training process, the current training process for the model involves forward passes and backward propagations. The validation accuracy started to plateau at the third epoch, so the model only went through 10 epochs due to early stopping. Observing the trend of data, the data seems to overfit with the current model, shown in (Figure 6 and Figure 7). The validation losses for both validation methods are higher than the training loss. At the same time the training accuracy is significantly higher than the validation accuracy and still increasing.

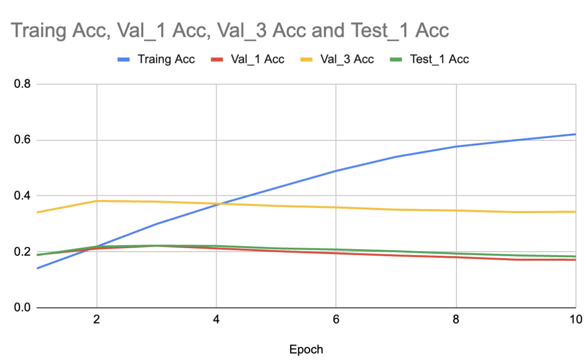

Figure 6. Accuracy trend for three datasets

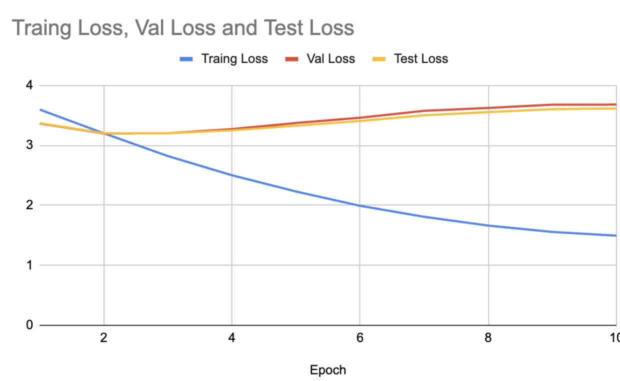

Figure 7. Loss trend for training and validation

3.4.2 Baseline Model

Model Architecture

The model architecture of the baseline model consists of 4 linear layers with dropout function and RelU activation functions. The input linear layer takes an input size of 1000 (since the word embedding size is 1000) and an output size of 50, which corresponds to the probability for each emoji. Each linear layer is passed to the ReLU activation function and then the dropout function. The dropout layer is applied with a dropout rate of 0.2. The model uses the Adam optimizer and Cross-Entropy loss function.

Training

The model training is set to have 1800 epochs with a 0.00005 learning rate. In each epoch, the x_data is first inputted into the model. Then the output of the model and the ground truth are compared using the loss function. Using backpropagation, we first compute the gradient of the loss using 'loss.backward()' and then update the parameters using 'optimizer.step()'. Also, in each epoch, we call 'optimizer.zero_grad()'.

The training accuracy at 1800 epochs is 12.75% and the validation accuracy for 1 emoji is 11.67%, while the training accuracy at the beginning is 8.52% and the validation accuracy is 8.46%. The accuracy for targets within the top 3 predicted emoji at 1800 epochs is 25.07%. It is clear that the model does learn some of the patterns and increases its performance. Figure 8 shows the accuracy trend for training, validation for 1 emoji, and validation for the top 3 emojis. Figure 9 shows the training and validation loss curves. It can be observed that the validation loss is slightly higher than the training loss, which seems normal since the model is trained using the training set.

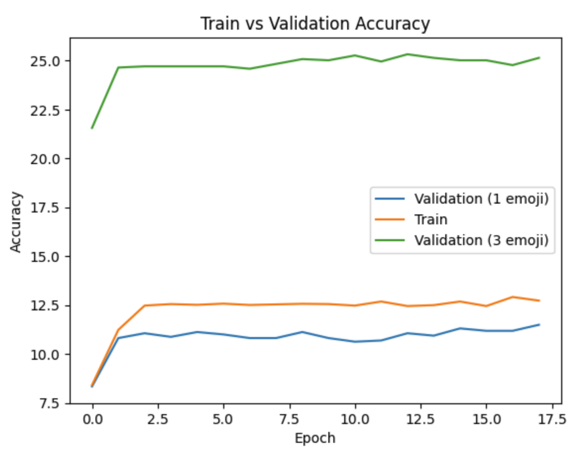

Figure 8. Accuracy trend for three datasets

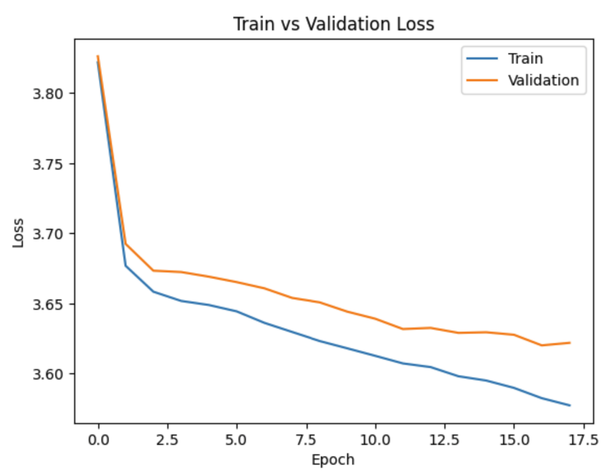

Figure 9. Loss trend for the training and validation

4. Test Results & Discussion

4.1 BERT Model

The test accuracy for the BERT model is 22.22%, and the accuracy for targets within the top 3 predicted emoji is 37.74%.

To address the overfitting problem of the model, various modifications have been made to the model with changes in data pre-processing to address the overfitting issue. Firstly, the cased model will learn the difference between words with capital case letters and words with lowercase letters. The difference in words captures different emotions within text samples, which is useful to predict the sentiment of the sentences for emoji prediction. Secondly, the additional dropout layer, with p=0.3, is another attempt to avoid overfitting the training data. With dropout, the neurons cannot rely on a single input for learning, since it could be dropped out on future training iterations. Lastly, L2 regularization is also used to combat overfitting. In addition, we have also tried adding more linear layers and dropout layers after the pre-trained BERT model in hope of reducing overfitting. However, the model accuracy reduces in this case.

Most semantic classification such as movie reviews only has three or five classes. However, the emoji prediction has fifty classes; therefore, it is much harder to classify than with 5 classes. In addition to validating the most probable prediction with the target, another validation method of checking the top 3 most probable predictions with the target is used. This validation method accuracy doubles the single most probable prediction accuracy and yields close to 40% accuracy. From Figure 11 below, the second most likely prediction contains the target for 2 out of 4 samples. This shows that the model has a good understanding of the relationship between emojis and input texts since the target emoji is likely within the top 3 most probable predictions. In the Figure 10 below, the “Predictions from model” is a 2D list. It contains information from four text samples in the test dataset. Each row represents the first or second or third most likely prediction of the model. Each column represents the sample index.

```
Target emojis [39, 10, 33, 5]
Predictions from model [[39, 10, 12, 40], [16, 40, 7, 35], [0, 38, 8, 10]]
```

Figure 10. The target of samples and the prediction of the model

```
Prediction Sample 1
Target prediction 🎶 Most likely 🎶 Second likely 🤩 Third likely 💋
Target prediction 🎉 Most likely 🎉 Second likely 🥳 Third likely 🙏🏾
Target prediction 🤍 Most likely 😁 Second likely 🙏🏻 Third likely ‼️
Target prediction 🖤 Most likely 🥳 Second likely 💖 Third likely 🎉

Prediction Sample 2
Target prediction 🙏🏻 Most likely 😁 Second likely 🙏🏻 Third likely 😘
Target prediction 😁 Most likely 📸 Second likely 😁 Third likely ✨
Target prediction 😎 Most likely 😎 Second likely 🙏🏾 Third likely 💯
Target prediction 🎉 Most likely 🔥 Second likely 🖤 Third likely ❤️
```

Figure 11. The sample prediction index turned into emojis

The first reason for having the large discrepancy of most probable prediction vs top 3 prediction is that some emojis appear much more frequently than others as seen in Figure 4. This results in the model having a great understanding of the few frequent emojis because the model has more chances to learn them due to the higher number of appearances in the training set. On the other hand, the model has less chance to learn the less frequent emojis, so the model will memorize the few cases included in the training set. Thus, the model does not have enough information to generalize the learning for the less frequent emojis. In the end, the model will predict a higher probability for more frequent emojis than less frequent emojis due to memorization of the less frequent emojis. Secondly, despite having more than 16000 examples, there is not enough data to cover all emojis. BERT is a powerful model, so feeding too little data will cause memorization. The memorization is shown through the modeling achieving 62% training accuracy with less than 20 validation accuracy.

4.2 Baseline Model

The test accuracy for the baseline model is 12.40%, and the accuracy for targets within the top 3 predicted emoji is 27.14%.

The model does not show signs of overfitting. It may be underfitting a bit, however, it does not seem to be the major issue. The lower accuracy is primarily due to the relatively simple model architecture. Before tuning, the previous baseline model only had only two linear layers and RelU activation functions, which resulted in lower accuracy and slightly overfitting. Therefore, two more linear layers and the dropout layers are added to the model to combat these deficiencies. They have proven to be effective in terms of reducing overfitting and increasing the model performance.

A fraction of the test outputs are printed out for analysis. In Figure 12, each row of the tensor is the top 3 predictions for each input. The first column is the output with the highest probability, and so on. The comparison between ground truth and model outputs for those inputs can be seen in Figure 13. The predictions from the model are mainly labels 10, 12, 16, and 18, which are emojis in Figure 12. From Figure 15, we can see that these correspond to the emojis with higher frequencies in the dataset (overall 16200 emojis). We can see that the model is correctly predicting 3 out of 5 cases. This suggests that the model is learning since it does not constantly predict the most frequent emoji. However, its learning is limited since it still is affected by the mass number of several particular emojis. Due to the simple model architecture, this is relatively hard to improve further.

```
topk_indices[201:206]

tensor([[12, 16, 18],
        [12, 18, 10],
        [12, 10, 18],
        [12, 18, 16],
        [12, 10, 18]])
```

Figure 12. Fraction of output of model

	Model Output			Ground Truth
	First likely	Second likely	Third likely	
Data 1	12	16	18	24
Data 2	12	18	10	2
Data 3	12	10	18	10
Data 4	12	18	16	18
Data 5	12	10	18	12

Figure 13. Compare ground truth and model output

```
print(l[9],l[11],l[15],l[17])

['10', '🔥'] ['12', '❤️'] ['16', '🙏'] ['18', '😂']
```

Figure 14. Encoded emoji

```
[28] all_emoji.count('❤️')
     2275

[29] all_emoji.count('🔥')
     1370

[30] all_emoji.count('🙏')
     613

[31] all_emoji.count('😂')
     2042
```

Figure 15. Frequencies of emoji

4.3 Comparison between Baseline Model and BERT Model

To test the performance of the BERT model, two benchmark models are created to compare the result with the BERT model. The first model uses the most frequent emoji in the dataset to make all predictions. In the training procedure, the model will always predict the most frequent emoji within the dataset. This model has a prediction accuracy of 14.04%. The accuracy is high due to one outlier emoji '❤️' that appears much more frequently than other emojis. The second benchmark is the baseline model, with a prediction accuracy of 12.17%.

Comparing these benchmarks, the BERT model performs better than both the most frequent emoji prediction model and the baseline model. Firstly, the BERT model performs better than the simple baseline model by 10 percent. The BERT model performs better than the random generating model by 4 percent in validation accuracy. This shows that the BERT model indeed learned from the training epochs by having the best accuracy. As for training accuracy, the BERT model has 62.04% accuracy with only 10 epochs, which is significantly higher than all the other two models. The high training accuracy shows that the BERT model has great potential when there is a large amount of data. With a database with millions of training samples with even distribution of frequency between emojis, the BERT model is likely to have high validation accuracy as well.

5. Broader Impact & Ethical Implications

Emoji is a graphic language that visualizes emotions and thoughts, allowing people to express what they want to say more directly and clearly. This is a great help for people who have difficulty choosing or expressing themselves in words. Communication through emojis can improve the efficiency of Internet and computer-mediated communication and allow people to feel more delicate.

This project makes it possible to write text with emojis through machine learning. Also, this research project can be used in many other ways, and different uses can make it work in different ways.

For example, this model can be added to an internal company platform and can be used as a tool and a way to understand and assess the mental health of employees. By allowing employees to communicate with the model, the output emojis can be used to effectively and directly understand the employee's mental state. It allows colleagues and practitioners to observe negative signals and risks early and intervene promptly, thus improving overall productivity and health.

Also, this model can help autistic, dyslexic, and deaf people to better understand the emotions of others. They are a group of people that may be able to understand the meaning of words, but not the emotions of others. Through the output of emojis in this model, they can better understand the emotions and meanings of the sentences and can convey their thoughts and emotions through this model.

However, human language is complex and diverse, and a single sentence in different contexts often conveys different meanings. Machines are often unable to interpret the hidden meanings of statements well, and thus may predict wrong and hurtful emojis. This drawback is that using machine learning to predict emotions is risky and could be used as a tool to hurt others. For example, by changing the prediction algorithm so that the model is always on the negative side, people are led to become negative and closed-minded, leading to more serious mental health problems.

6. Conclusion

This project aims to use machine learning to build a model that predicts the emoji to best represent the given text. The raw data is pre-processed and the top 50 most frequent emojis and their corresponding text are kept. After pre-processing, data is tokenized to load into the BERT model and baseline model using different techniques.

The BERT model has been implemented and used as the main predicting model. It takes 70% of data for training, 15% for validation and 15% for testing. The model architecture contains another dropout layer, a linear layer, and a sigmoid layer added onto the pre-trained BERT model for making the prediction. With a batch size of 16, the training accuracy for the BERT model is 62.04%, and the test accuracy is 22.22%. Its testing accuracy for targets within the top 3 predicted emoji is 37.61%. To address the overfitting problem, more data should be collected since BERT is a powerful model and feeding too little data will cause memorization problems.

Two benchmark models are used to analyze and compare the performance with the BERT model. The first model always predicts the most frequent emoji in the dataset, and has an accuracy of 14.04%. The second model uses word encodings and gets a testing accuracy of 12.40% and its testing accuracy for targets within the top 3 predicted emoji is 27.14%. According to the accuracy results, it can be concluded that the BERT model performs better than both the most frequent emoji prediction model and the baseline model.

This project makes it possible to write text with emojis through machine learning. Implementing this model in different areas can help different people. However, since the machine lacks human emotion and understanding, it can sometimes misinterpret statements and thus predict the wrong expressions, allowing some unsuspecting people to take advantage of this shortcoming to hurt others.

There is still scope for improvement of this model. Due to the limitation of collecting comprehensive datasets, the current model's accuracy has space for improvement. If the dataset contains millions of entries, BERT will have the capacity to learn them and improve its performance. In addition, we can add other features to the model, such as time, location, user's age and so on, to get more user-specific emoji recommendations.

Reference

[1] Ramaswamy, S., Mathews, R., Rao, K., & Beaufays, F. (2019, June 11). *Federated learning for emoji prediction in a mobile keyboard*. arXiv.org. Retrieved January 31, 2023, from https://arxiv.org/abs/1906.04329

[2] *(PDF) context-aware emoji prediction using Deep Learning - Researchgate*. (n.d.). Retrieved February 1, 2023, from https://www.researchgate.net/publication/358171937_Context-Aware_Emoji_Prediction_Using_Deep_Learning

[3] *Emoji Prediction: Extensions and benchmarking - arxiv.org*. (n.d.). Retrieved March 28, 2023, from https://arxiv.org/pdf/2007.07389

[4] *Twitter scraper · Apify*. Apify. (n.d.). Retrieved March 26, 2023, from https://apify.com/quacker/twitter-scraper

[5] 262588213843476. (n.d.). *Top 1000 celebrity twitter accounts (Twitter,domain,name,type)*. Gist. Retrieved March 26, 2023, from https://gist.github.com/mbejda/9c3353780270e7298763

[6] *Sentiment analysis with BERT and transformers by hugging face using pytorch and python*. Curiousily. (n.d.). Retrieved March 27, 2023, from https://curiousily.com/posts/sentiment-analysis-with-BERT-and-hugging-face-using-pytorch-and-python/

[7] Ganesan, K. (2021, July 9). *Gensim word2vec tutorial: An end-to-end example*. Kavita Ganesan, PhD. Retrieved April 13, 2023, from https://kavita-ganesan.com/gensim-word2vec-tutorial-starter-code/#.ZDjKsPbMK3A