Emoji Prediction Using Bidirectional LSTM

Arup Mazumder
Piriyankan Kirupaharan

December 23, 2022

Abstract

We have studied a variety of machine learning concepts throughout the course and used many of them, either directly or indirectly, in our final project. We also wanted to create a project that would intellectually and academically challenge us since we learned a lot from this course. The purpose of our final project was to predict emojis based on the context of the given text. Emojis have become a more common and significant component of modern textual inputs in the social media era, and any modern text system is now very interested in studying how they are used. We almost always use emojis to convey a certain emotion. Since we trained our model using tweets from Twitter, we classify this as a text classification problem. We obtain better accuracy and F1 results with our LSTM-RNN model.

1 Introduction

An emoticon is a single, brief line of text that typically contains punctuation. Since the 19th century, when they were originally used in joking and casual writing, emoticons have been around. According to Fahlman, to distinguish between lighthearted and serious posts, use:-) and:-(. Within a few months, emoticons were widely used, and the collection of emoticons was enlarged to include hugs and kisses by using characters from a regular keyboard [Kralj Novak et al.(2015)]. In regular internet communications for 10 years, emoticons had become ubiquitous and were now regarded as web parlance. An emoticon is a symbol that represents a facial expression, like ;-). It enables the users to convey feelings, moods, and emotions and enhances written communication with non-verbal cues. A step further, emojis were created using cutting-edge communication technologies to enable more expressive messages. Emojis are graphic symbols, or ideograms, that represent concepts and ideas in addition to facial expressions. The emoticons provide an essential piece of information when analyzing short informal texts like tweets, blogs, or comments, it turns out [Boia et al.(2013)].

We all read and write short texts on a regular basis. We rely on very well-liked platforms like Twitter, Instagram, and WhatsApp to share our interests, beliefs, feelings, and daily activities through brief texts. Besides text, Emoji are frequently used on those biggest social media platforms in today's world. Twitter tweets make up our text data, where Emojis are used in approximately 20% of all tweets. This makes it a very competitive source of text and emoji that can be incorporated into the construction of machine-learning models. However, Emojis come in a variety of forms, from facial gestures to objects, animals, and locations. We used 16 different emoji types throughout our study.

We first motivate and outline the key components of this emoji prediction task in this paper. We specifically tried different approaches to improve accuracy using the same words and emojis from tweets. To improve the efficacy of our models, we utilized various hyperparameter fine-tuning techniques. We wrote about some background, data source, and overall process in the introduction chapter. We talked about a few related studies in the second chapter. We discussed the procedures we used in the third chapter. We explained the result in the fourth chapter. The paper is concluded with the key findings derived from how this challenge was structured.

2 Related Works

Emojis are a common and pervasive form of communication currently used by virtually every social media service and instant messaging platform [Tanimu Jibril and Abdullah(2013)]. In 2013, Park and her team looked into the semantic, cultural, and social aspects of emoticon usage on Twitter. They found that emoticons are socio-cultural norms, whose meaning can change depending on the speaker's identity rather than just being used to express a single emotion or make jokes. On a sizable dataset of more than a billion Tweets from various countries and eras, they validated their findings [Park et al.(2013)]. They discovered the following year that individuals from individualistic cultures favor horizontal and mouth-oriented emoticons like :), while individuals from collectivistic cultures prefer vertical and eye-oriented emoticons like :) [Park et al.(2014)].

Emojis offer a visual and speedy method of communication by allowing us to express objects, situations, and even emotions through small images. [Barbieri et al.(2016)] used distributional semantic models to analyze the emojis used on Twitter. To embed words and emoji, they used a skip-gram model. These embeddings are then used in a similarity-matching process to generate predictions. Luda and Connie conducted a study using neural networks to predict emojis. They compared their developed LSTM-RNN model and CNN model and found higher accuracy and F1 scores than the benchmark [Zhao and Zeng(2017)].

3 Methods

3.1 Data

The data source for our project is from Kaggle Twitter dataset `https://www.kaggle.com/datasets/hariharasudhanas/twitter-emoji-prediction`. The dataset contains 70,000 records with 20 classes. Even though there is a separate validation dataset since it doesn't contain the label. The validation dataset is not used in our project. The below figure shows the distribution of records per class.

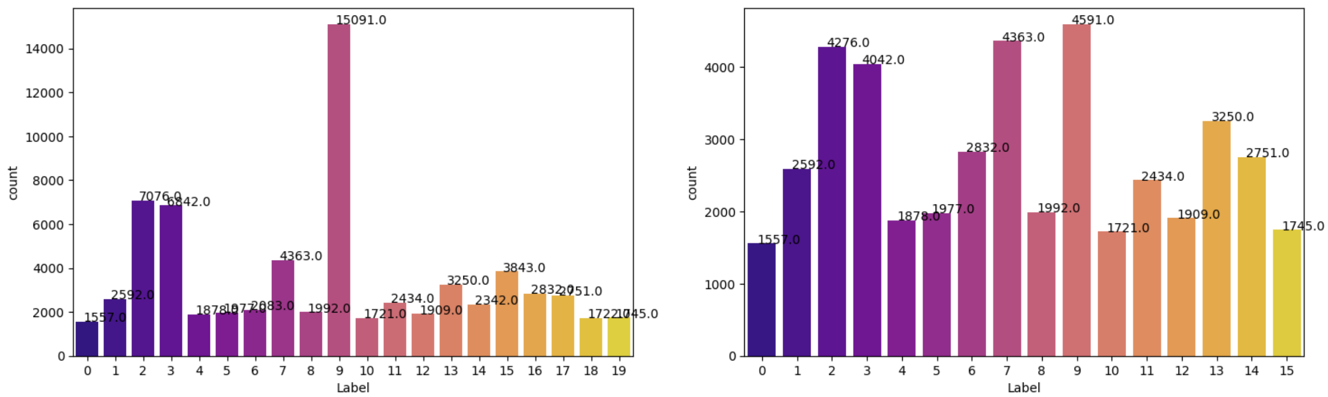

Figure 1: The distribution of the text with each class. The raw dataset in the left side, and the processed data in the right side

Since the data is not distributed normally, we have to remove certain records randomly so that the model not be biased toward a certain class. For example, class 9 had around 20% of the records. If we don't remove partial records of this class randomly, the trained model may be biased to class 9. Similarly, we had to do this for classes 2 and 3. After cleaning the dataset, we have a total record of 43,910.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
😜	📸	😍	😂	😉	🎄	😎	🔥	😘	❤	😁	🇺🇸	☀	✨	😊	💯

Figure 2: The distribution of the text with each class. The raw dataset in the left side, and proceed data in the right side

In the initial dataset, we had 20 classes, but 4 classes out of 20 had the same meaning. These 4 were not in the most used emoji list as well`https://www.makeuseof.com/top-emojis-explained-cheat-sheet/`. So we decided to remove the emoji classes as a step of pre-processing.

3.2 Model Building

A model created for one task is used as the basis for another using the machine learning technique known as transfer learning. Pre-trained models are frequently utilized as the foundation for deep learning tasks in computer vision and natural language processing because they save both time and money compared to developing neural network models from scratch and because they perform vastly better on related tasks.

GLOVE 6B 50D is a pre-trained word vector that has 6 billion words and a 50-dimensional representation. A corpus of aggregated global word-word co-occurrence statistics is used for the training and the representations that emerge highlight some intriguing linear substructures of the word vector space. We are using this already-trained model to get the embedded values for a given word in our dataset.

In feedforward, a neuron from the hidden layer is linked with neurons from the layer above it and the layer below it. The output of a neuron in such a network can only be transferred forward; it cannot ever be passed to a neuron in a higher or even a lower layer. The above behavior is different from recurrent neural networks, though. A neuron's output may very well serve as the present layer's or a preceding layer's input. In comparison to how feedforward neural networks are built, this is considerably more similar to how our brain functions. A recurrent Neural Network(RNN) is a type of Neural Network where the output from the previous step is fed as input to the current step.

Vanishing gradients are a difficulty for RNNs. When the gradient shrinks too much, the parameter changes lose their significance because the gradients convey information that the RNN uses. Long data sequences are difficult to learn as a result. As a solution to this problem, Long Short-Term Memory (LSTM) networks are capable of learning order dependence in sequence prediction problems.

We approached this problem by training different types of models.

- Single layer LSTM with GLOVE
- 2-Layer LSTM with GLOVE
- Bidirectional LSTM with GLOVE

The number of units for the LSTM was a hyperparameter. For Bidirectional LSTM, we initially had 512 units for the first layer and 256 for the second layer. For this, we received an accuracy of 43%. Then we changed the number of units of the first layer to 1024 and the number of units of the second layer to 512 units. With these hyperparameters, we got an accuracy of 50%.

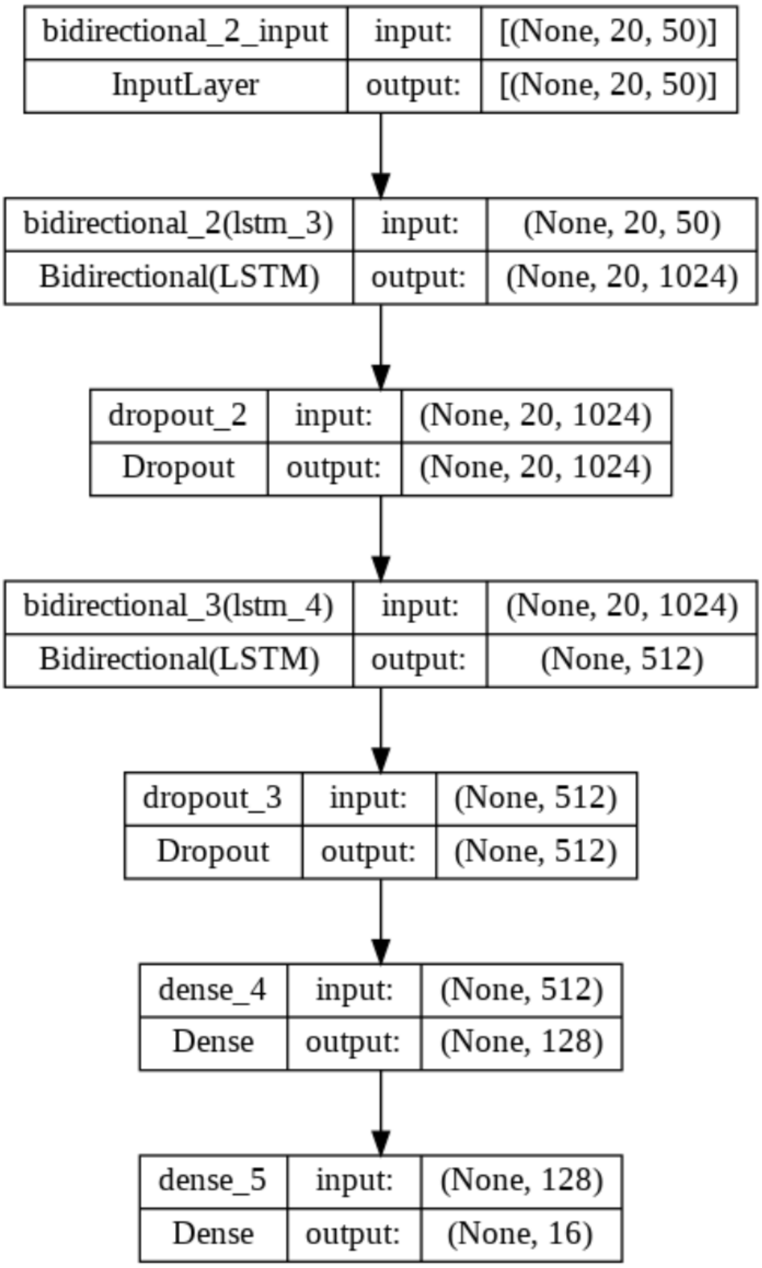

Figure 3: Bidirectional LSTM model

Each sentence in the processed data is converted to a (20*50) vector using the GLOVE embedding. The below figure shows what a single word in the sentence looks like once applied with GLOVE. Each word will have 50 vectors. We assumed the maximum number of words in a sentence is 20. So the final training matrix will be in the shape of (43,910*20*50). And the labels of each record are converted to categorical data, which the matrix shape is (43,910*16). SMOTE is a method of oversampling in which artificial samples are produced for the minority class. This method aids in overcoming the overfitting issue brought on by random oversampling. With the use of interpolation between the positive instances that are close together, it concentrates on the feature space to produce new instances. We have used this technique in our project to avoid overfitting. So the final matrix shape for text is (73456, 20, 50), and the final matrix shape for the label is (73456, 16)

```
print(embeddings_index['happy'])
print(embeddings_index['happy'].shape)
✓ 0.2s
[ 0.092086  0.2571    -0.58693  -0.37029   1.0828    -0.55466  -0.78142
  0.58696  -0.58714   0.46318  -0.11267   0.2606    -0.26928  -0.072466
  1.247     0.30571   0.56731   0.30509  -0.050312 -0.64443  -0.54513
  0.86429   0.20914   0.56334   1.1228   -1.0516    -0.78105   0.29656
  0.7261   -0.61392   2.4225    1.0142   -0.17753    0.4147   -0.12966
 -0.47064   0.3807    0.16309  -0.323    -0.77899   -0.42473  -0.30826
 -0.42242   0.055069  0.38267   0.037415 -0.4302    -0.39442   0.10511
  0.87286 ]
(50,)
```

Figure 4: Each word matrix from Glove

3.3 Web Implementation

We used standard HTML, CSS, JS, and jQuery for our web implementation that is deployed with NGINX webserver. In the frontend, there is a textarea where a user can type a sentence. There are two buttons: one is for viewing the predicted emoji, and the other is for clearing the text in the textarea.

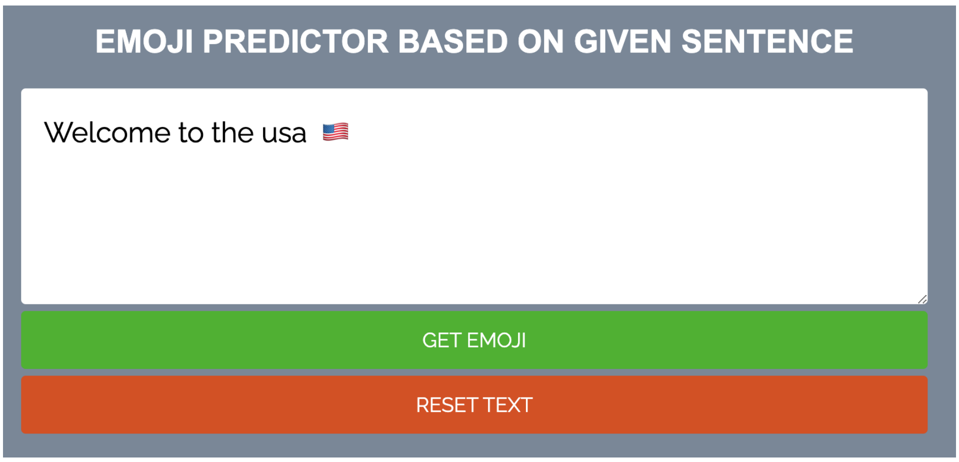

Figure 5: Web implementation: Frontend

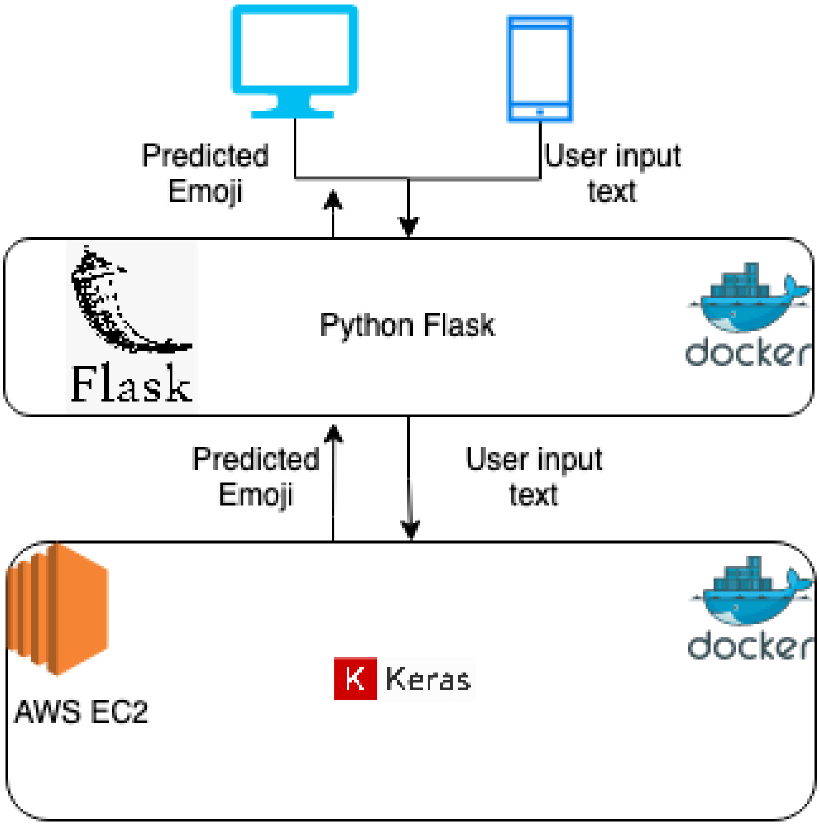

Figure 6: Deployment Architecture

For the backend, we used Python flask as a framework to develop REST API. This REST API is responsible for getting the API call from the front end. The best pre-trained model whose accuracy was the best among other models was used to predict a context-related emoji from a total of 16 emoji(classes) based on an API call. Docker was used to containerizing the entire system, and an AWS webserver was used for deployment.

4 Results

We discussed the findings from all three of these models—Single Layer LSTM, Two Layers LSTM, and Bi-directional LSTM—in this section. We compared the outcomes from other models to our base model, Single Layer LSTM, in order to establish which model performed better. The accuracy, precision, recall, and F1 score for each of the three models are displayed in table 1.

	Single layer with GLOVE	2-Layer with GLOVE	Bidirectional with GLOVE
Accuracy	0.42	0.42	0.50
Precision	0.41	0.41	0.50
Recall	0.42	0.42	0.50
F1 score	0.42	0.41	0.50

Table 1: Results from all three models

We obtained an accuracy of 0.42, a precision of 0.41, a recall of 0.42, and an F1 score of 0.42 for our base model, a single-layer LSTM. For a two-layer LSTM, we achieved an F1 score of 0.41, accuracy of 0.42, precision of 0.41, recall of 0.42, and. Finally, our bidirectional LSTM yielded an accuracy of 0.50, a precision of 0.50, a recall of 0.50, and an F1 score of 0.50.

The below figures show the confusion matrix of the three models. From the confusion matrix of Bidirectional LSTM, out of 14,692 test data, the model predicted 6,367 correctly. And we can notice higher incorrect predictions for class 6 as class 13 and class 0. And can notice higher incorrect predictions for class 9 as class 2.

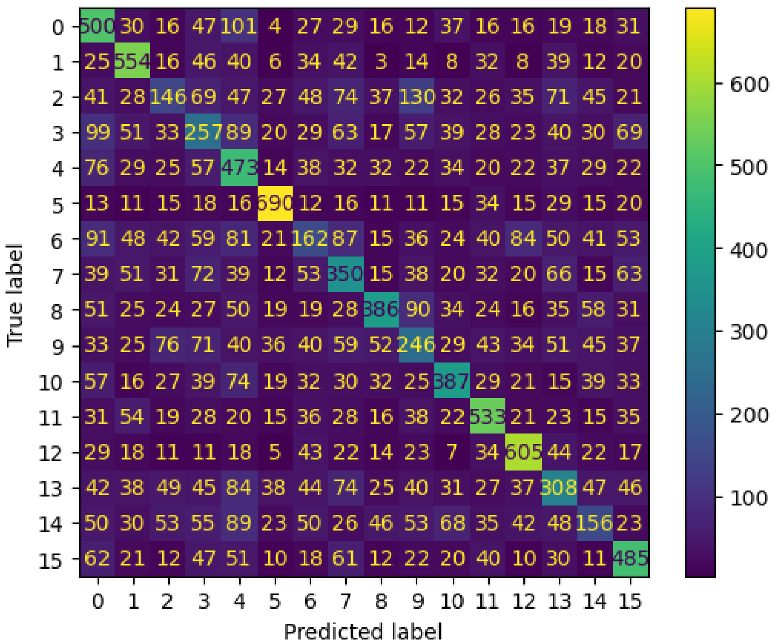

Figure 7: The Confusion Metrix for Single Layer LSTM model result

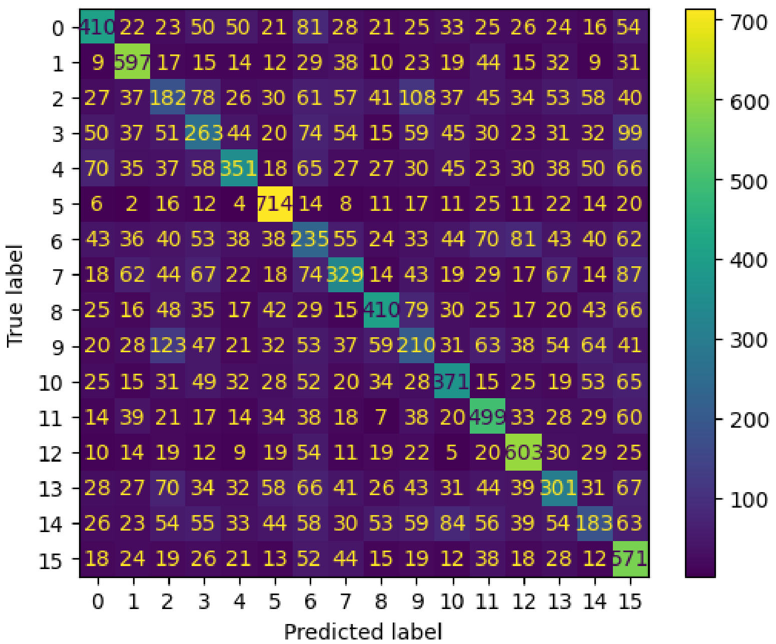

Figure 8: The Confusion Metrix for Two Layers LSTM model result

The Precision, Recall, F1-scores, and Accuracy values for each class that we obtained from the Bidirectional LSTM model are shown in the table 2 below. We can see that classes 5, 12, and 1 received the highest F1 scores, with scores of 0.75, 0.72, and 0.67, respectively.

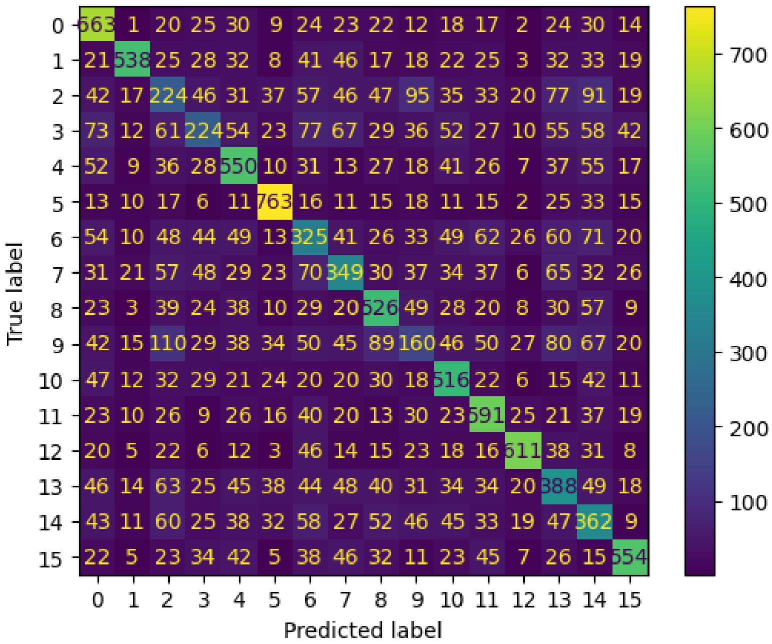

Figure 9: The Confusion Metrix for Bidirectional LSTM model result

Class	Precision	Recall	F1-Score
0	0.55	0.71	0.62
1	0.78	0.59	0.67
2	0.26	0.24	0.25
3	0.36	0.25	0.29
4	0.53	0.57	0.55
5	0.73	0.78	0.75
6	0.34	0.35	0.34
7	0.42	0.39	0.40
8	0.52	0.58	0.55
9	0.25	0.18	0.21
10	0.52	0.60	0.55
11	0.56	0.64	0.60
12	0.76	0.69	0.72
13	0.38	0.41	0.40
14	0.34	0.40	0.37
15	0.68	0.60	0.63
Accuracy			0.50

Table 2: Class based Precision, Recall, F1-scores, and Accuracy for Bidirectional LSTML Model

5 Conclusion

On social media sites like Twitter, emojis like have transformed into more well-liked in recent years. For different meanings and emotions, emojis are frequently used in combination with text in web messages. Our project aims to create a model that can correctly predict contextual emojis. We obtained 70000 labeled sentences for this task from the text dataset of Twitter tweets with emoji. At first, we had 20 labels. We preprocessed the text because there were some imbalances, which led to a reduction in label size. Approximately, 44000 sentences and 16 labels we used in our final model. We used the concept of transfer learning, pre-trained glove vector, and LSTM models. Out of the single-layer LSTM and stacked LSTM, and Bidirectional stacked LSTM, the Bidirectional model performed better.

Working with this noisy data was a significant contributor to the challenge. Emojis are interpreted differently by each person, and they are frequently combined unintentionally. Additionally, tweets have special characters and mentions, which is not applicable in real-world messaging. Cleaning the dataset to remove nonsensical tweets may help improve the classification performance. Apart from LSTM, we believe there is a high probability that different models like CNN can perform better. Also, in Glove, we have multiple embedding types like 100 dimensions and 200 dimensions. But we have only used 50 dimension vectors. We believe training the model with different dimensions may improve the accuracy.

References

[Barbieri et al.(2016)] Francesco Barbieri, Francesco Ronzano, and Horacio Saggion. 2016. What does this Emoji Mean? A Vector Space Skip-Gram Model for Twitter Emojis. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. European Language Resources Association (ELRA), Portorož, Slovenia, 3967–3972. https://aclanthology.org/L16-1626

[Boia et al.(2013)] Marina Boia, Boi Faltings, Claudiu Cristian Musat, and Pearl Pu. 2013. A :) Is Worth a Thousand Words: How People Attach Sentiment to Emoticons and Words in Tweets. *2013 International Conference on Social Computing* (2013), 345–350.

[Kralj Novak et al.(2015)] Petra Kralj Novak, Jasmina Smailovic, Borut Sluban, and Igor Mozetic. 2015. Sentiment of Emojis. *PloS one* 10 (09 2015). https://doi.org/10.1371/journal.pone.0144296

[Park et al.(2014)] Jaram Park, Young Min Baek, and Meeyoung Cha. 2014. Cross-Cultural Comparison of Nonverbal Cues in Emoticons on Twitter: EVIDENCE from Big Data Analysis. *Journal of Communication* 64 (03 2014). https://doi.org/10.1111/jcom.12086

[Park et al.(2013)] Jaram Park, Vladimir Barash, Clay Fink, and Meeyoung Cha. 2013. Emoticon Style: Interpreting Differences in Emoticons Across Cultures. *Proceedings of the 7th International Conference on Weblogs and Social Media, ICWSM 2013*.

[Tanimu Jibril and Abdullah(2013)] Ahmed Tanimu Jibril and Mardziah Abdullah. 2013. Relevance of Emoticons in Computer-Mediated Communication Contexts: An Overview. *Asian Social Science* 9 (03 2013). https://doi.org/10.5539/ass.v9n4p201

[Zhao and Zeng(2017)] Luda Zhao and Connie Zeng. 2017. Using Neural Networks to Predict Emoji Usage from Twitter Data.

6 Code Repository

GitHub: https://github.com/yankanp/emoji-predictor