

Empathy Emoji Chat-bot via BERT, LSTM and AutoML

Han Yu, Lubo Han

Abstract

Emojis are widely used in social media, blogs, and instant messages to express users' emotions and intentions. In this report, we present our analysis protocol and preliminary results to predict emojis based on open text in English by using the [“English Tweet, with Emoji”](#) dataset from Kaggle.

Sample dataset and notebooks(WiP) can be accessed at [GitHub:XCS224U-Project-Emoji-ChatBot](#)

1 Introduction

Our project idea is that given input texts to Chat Bot, based on the prediction of the sentiment and the topic discovered in the input text, the Chat Bot will return an Emoji which “Emphasizes” the topic with empathy. The problem we are trying to solve can be used for sentiment understanding of the text input, and it can also be used to build an emoji recommendation system to enrich text-based sentences.

2 Related Work

Sequence-to-sequence(seq2seq) models are widely used in deep-learning to convert sequences of Type A (e.g., speech in French) to sequences of Type B (e.g., translated text in English). Recurrent Neural Network (RNN) based seq2seq models have gained a lot of traction since its introduction in 2014 ([Sutskever et al., 2014](#)), and they have been widely used to develop deep neural network models to power a variety of NLU tasks, such as machine translation, text summary, speech recognition, question-answering system, and video captioning. The performance of the seq2seq models was further improved with the addition of the attention mechanism ([Luong et al., 2015](#)) to capture the interdependence between the input and output sequences, and also within the input sequence.

In paper “Attention Is All You Need” ([Vaswani et al., 2017](#)), researchers from Google introduced a new seq2seq model, named “Transformer”. While “Transformer” still follows a classical encoder+decoder structure, it eschews recurrence and instead relies entirely on attention mechanisms to draw global attention between input and output. The nature of the transformer’s architecture also enables more parallelism computations to achieve the new state-of-the-art results in machine translation when using the WMT 2014 English-to-German and WMT 2014 English-to-French dataset.

The “Transformer” introduced in “Attention Is All You Need” ([Vaswani et al., 2017](#)) has been further used in Google BERT framework ([Devlin et al., 2018](#)). The BERT framework involves two steps to build new language representation models across different tasks: namely pre-training and fine-tuning. During the pre-training phase, the model is trained on unlabeled data over two unsupervised pre-training tasks (Masked Language Modeling and Next Sentence Prediction). During the fine-tuning phase, the BERT model is first initialized with the pre-trained parameters, and then all the parameters are fine-tuned using labeled data from the downstream tasks. In summary, BERT is a unified architecture across different NLU tasks, and there is minimal difference between the architecture of the pre-trained language representation model and the final models that are fine-tuned for a specific task. Architectures like BERT demonstrate that unsupervised learning (refer to the pre-training step in BERT) is going to be a key element in many language understanding systems, especially for low resource tasks (e.g., lack of labeled training data or limited computational resources). The “two-steps” frame introduced in BERT is very similar to the “transferring learning” concept which has been widely applied in the field of computer vision. For instance, a pre-trained deep learning model

```

Line0: <START> O
Line1: No O
Line2: object O
Line3: is O
Line4: so O
Line5: beautiful O
Line6: that O
Line7: under O
Line8: certain O
Line9: conditions O
Line10: it O
Line11: will O
Line12: not O
Line13: look O
Line14: ugly O
Line15: Oscar O
Line16: Wilde O
Line17: u O
Line18: RT :red_heart:
Line19: ... O
Line20: <STOP> O
Line21:
Line22: <START> O
Line23: Cant O
Line24: expect O
Line25: different O
Line26: results O
Line27: doing O
Line28: the O
Line29: same O
Line30: thingdoing O
Line31: stuff O
Line32: different O
Line33: from O
Line34: now O
Line35: on :person_shrugging:
Line36: [] O
Line37: :female_sign:
Line38: O
Line39: <STOP> O

```

Sample Raw Tweet Data

preprocess

tweet	emoji
id	
0 CeeC is going to be another Tboss What is 45 million Naira	(face_with_tears_of_joy)
1 This gif kills me Death is literally gushing towards you and you really gon do a whole 3point turn	(weary_face)
2 LOVE TEST Raw Real JaDine	(purple_heart)
3 I swear we dont gotta look it finds us	(face_with_tears_of_joy)
4 We would like to wish everyone a very Happy New Year and all the best in 2018	(party_popper)

Sample Cleaned Tweet Data

Figure 1: Clean Raw Dataset

could be fine-tuned for a new task on the [ImageNet](#) dataset and still give decent results on a relatively small labeled dataset.

Social network sentiment analysis has been drawing great attention due to its wide practical applications in business and society. Compared to mining textual contents, emoji is only lightly studied in sentiment analysis on social media platforms such as Twitter, Facebook, and Instagram. An interesting aspect of emoji is that it can be used in both positive and negative contexts(i.e irony, jokes, sarcasm). A novel way for Twitter sentiment analysis was proposed ([Chen et al., 2018](#)), where bi-sense emoji is analyzed in both positive and negative sentimental tweets individually with an attention-based long short-term memory network (LSTM), which outperforms the state-of-the-art models.

With the proliferation of animated GIF as a way to express emotions on social networks, interesting design of experiments and new benchmarks have been reported in [EmotionGIF 2020 challenge](#), where various latest NLU/NLP techniques are applied to predict the GIF response categories for unlabeled tweets: for example, pre-trained post-BERT models and hybrid BERT models are used to achieve the tasks with very good performance ([Bi et al., 2020](#)).

Inspired by those prior arts, we plan to explore different approaches to build a multi-class emoji classifier to predict emoji, given input text in English. We will fine-tune a BERT based

multi-class classifier to explore the state-of-the-art BERT framework; we will also explore those well-established seq2seq models by designing a LSTM-based multi-class classifier. Last but not least, we will leverage Google Cloud AutoML to auto-train a multi-class emoji classifier using the same training dataset. The model generated by Cloud AutoML will be used for benchmark studies.

3 Data

Raw dataset can be downloaded from Kaggle: “[English Tweet, with Emoji](#)”. The original files for this dataset were four archives from Archive Team: The Twitter Stream project. This Kaggle dataset reformatted these files, selected all the English-language tweets with at least one emoji, along with other pre-processing including: removing hashtags, URLs, mentions.

We cleaned and transformed the original dataset so that the dataset can be further used for model development and evaluation (e.g., purged all the tweets with multiple unique emojis, and removed duplicated tweets), see Figure 1. We planned to share our cleaned dataset on Kaggle so other machine learning practitioners can use our dataset and work on interesting NLU/NLP tasks in the future.

4 Models

There are several state-of-the-art models developed by the NLU/NLP communities which can be di-

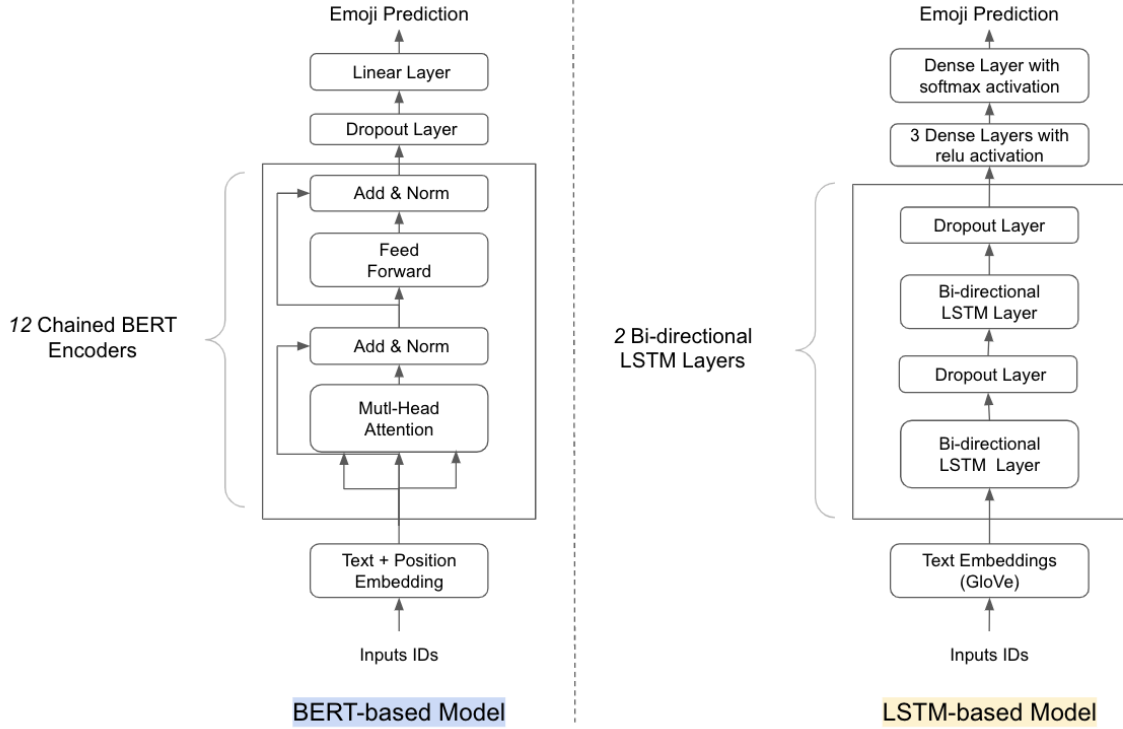


Figure 2: Emoji Classifier: BERT-based Model and LSTM-based Model

rectly used to build a emoji classifier through learning from a large corpus of texts. Those include LSTM, GRU, BERT, etc. We leveraged those existing “building blocks” to develop our own models and we have also fine-tuned the models through design of experiments. We have also trained models by using Google AutoML for bench-marking purposes.

4.1 BERT

Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018) is a two-steps modeling tuning framework developed by Google and open sourced in 2018. BERT has proven to be able to produce state-of-the-art results in many of the NLP tasks.

The Encoder used in BERT is an attention-based architecture that was introduced in “Attention Is All You Need” (Vaswani et al., 2017). We have leveraged the advances of transfer learning by combining a pre-trained BERT model (bert-base-cased) with a dropout layer and a linear layer, to fine-tune a multi-class emoji classifier.

4.2 LSTM

Long Short-Term Memory (LSTM) is an artificial Recurrent Neural Network (RNN) architecture

used in the field of deep learning. LSTM networks have good performance on classifying, processing, and making predictions based on time series data such as neutral language dataset. Bi-directional LSTM with Global Vectors (GloVe) (Pennington et al., 2014) embedding are among some of the best methods for building neutral language classifiers using RNN (Staudemeyer and Morris, 2019).

4.3 AutoML

Cloud AutoML is a suite of machine learning products that enables developers with limited machine learning expertise to train high-quality models specific to their business needs. It relies on Google’s state-of-the-art transfer learning and neural architecture search technology.

Google AutoML provides automated hyperparameter tuning for model training. Once model training is completed, developers can access the models via Cloud AutoML client libraries and build apps for batch/online predictions.

5 Experiments

5.1 Exploratory Data Analysis

As shown in figure 3, there are 49 unique emojis in our dataset, and their occurrences vary a lot. Popular emojis such as “face_with_tears_of_joy” and

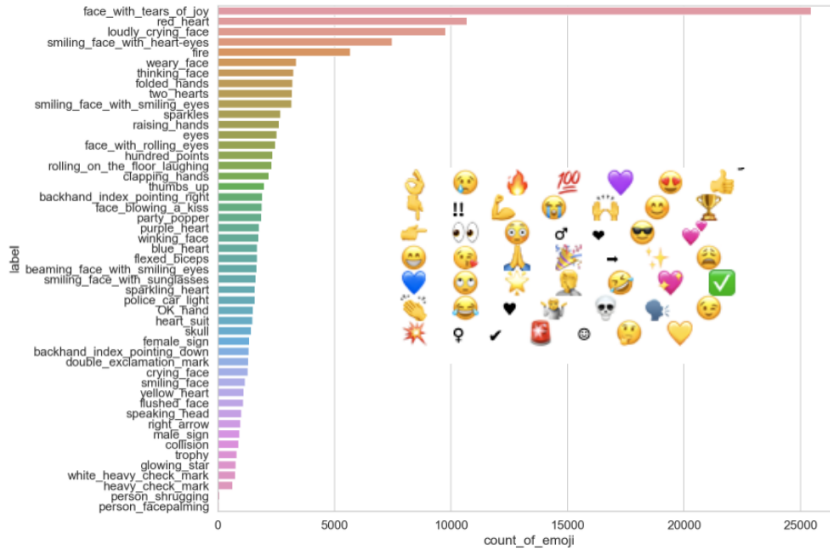


Figure 3: Unique Emoji and Distribution

“red_heart” have much more data points than the other less popular emojis, and presumably model performance will bias towards classes that dominate the training dataset.

We did a baseline training on raw data points for 16 classes with LSTM-based model, as in table 1, model trained with raw dataset tend to predict towards the popular “face_with_tears_of_joy” class which has very high recall, but low precision. The unweighted macro-F1 score is not high in this case (see the study in Data Re-sampling).

precision	recall	f1-score	rows in test
0.57	0.92	0.70	8966/19738 (45%)

Table 1: Results for “face_with_tears_of_joy” (LSTM, 16 emoji classes)

5.2 Data Re-sampling

To avoid model prediction bias toward high occurrence emojis, we explored several approaches to balance emoji classes.

The first approach is to sample a fixed size for each class using pandas.DataFrame.sample library. To get enough samples for low occurrence emojis, we initially set “replace=True” which allows sampling the same row more than once. But later we found an issue with this balancing approach: some rows are duplicated in re-sampling, and hence duplicated rows exist among training, validation, and test datasets. This gave us unrealistic good validation and test accuracy (>0.50).

The second approach is to down-sample all emoji classes to the size of the class with minimum occurrence. This works for the purpose of balancing, but potentially drops a lot of useful information from high occurrence emoji classes.

A variation of the second approach is to down-sample high occurrence emoji classes to the median of all classes, and keep the low occurrence as-is. Through our experiments, we found the median down-sample approach gets us the best overall macro-F1 score.

In LSTM-based model training, we also tried Synthetic Minority Over-sampling Technique (SMOTE) (Chawla et al., 2011). A SMOTE step is added after the median down-sample step to generate synthetic rows for low occurrence emoji classes. But the experiment result shows that this only helped on validation accuracy (validation set has synthetic rows), but not on test accuracy (test set has no synthetic rows). This is likely because that mathematically generated data points are not able to reflect the real world tweets and emojis.

Based on the studies, we used balanced-median re-sampling approach for LSTM-based models.

Re-sampling Approach	Macro-F1	Accuracy
Baseline	0.24	0.52
Balanced-Min	0.27	0.28
Balanced-Median	0.31	0.32
Balanced-Median-SMOTE	0.29	0.30

Table 2: Results with different re-sampling approaches (LSTM, 16 emoji classes)

5.3 Build Dataset for Training, Validation, and Testing

The balanced dataset generated from the “re-sampling” has been further split into training, validation, and testing dataset ($51,780 * 90\% = 46,602$ for training, $51,780 * 5\% = 2589$ for validation, and $51,780 * 5\% = 2589$ for testing) via stratified sampling strategy. This guarantees that training, validation, and testing dataset will all be balanced dataset. (Percentage in LSTM-based model training is slightly different: 81% training, 9% validation, 10% testing.)

5.4 Model Setup

5.4.1 BERT-based Model

In order to use BERT to build a machine learning model, text input data needs to be pre-processed as follows:

- Add special tokens to separate sentence: for example, using [SEP] for ending of a sentence, [CLS] for start of a sentence, [PAD] for padding sequence of tokens to the same length, and using [UNK] for tokens that are not in the training set of BERT pre-trained models.
- Pad input sequence of tokens to be of the same length.
- Create array of 0s (pad token) and 1s (real tokens) called attention mask.

Since BERT works with fixed-length sequences, we use the distribution plots of the tokens to choose the max length. Tokens for each tweet are generated using BERT tokenizer provided from the [hugging-face transformer library](#).

Based on the distribution of token lengths generated from our input dataset (see Figure 4), we selected 60 as the max sequence length to pre-process input sequence of tokens.

A pre-trained BERT model (12 layers, 768 hidden states, 12 heads, 110M parameters, pre-trained on lower-cased English text) is selected to build our Emoji classifier. Figure 2 illustrates the key building blocks in our model: in a BERT encoder, numerical representation of each word in the corpus is a combination of position encoding and token embedding; the embedding of each sample is then fed into a total of N encoder blocks (in our model, $N=12$) which are chained together to generate the encoder’s outputs; the outputs from the encoder

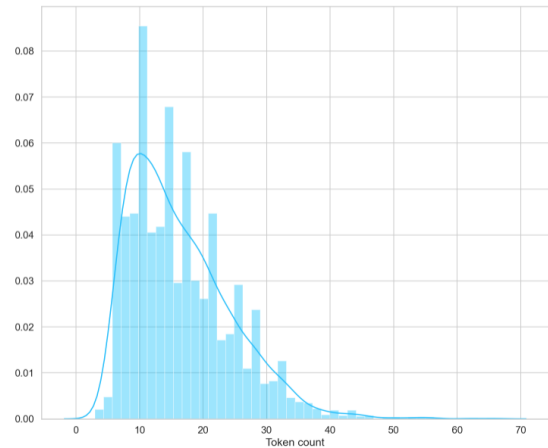


Figure 4: Distribution of Token Length using BERT Tokenizer

is then forward to a Feed-Forward Network. The BERT encoder can then be augmented with other layers to build a classifier (e.g., in our model, we added a dropout layer and a linear layer on top of the BERT encoder).

5.4.2 LSTM-based Model

LSTM-based model takes tweet text embedding and emoji class labels as input. The tweet pre-processing includes:

- Split the raw tweet text to lower case tokens using nltk TweetTokenizer library.
- The generated tokens are then converted to embeddings using the 50-dimension 27B-token GloVe vectors pre-trained on Twitter dataset. (Twitter-based GloVe embeddings perform slightly better than wiki-based ones in this task.)
- Use random vector for token that does not have corresponding GloVe embedding.
- Pad input sequence of tokens to be of the same length. According to the token distribution in Figure 5, we chose 30 as the fixed length for LSTM-based model.

The emoji class label is one-hot encoded with TensorFlow `keras.utils.to_categorical` library so that it is compatible with the model fitting interface.

As shown on the right side of Figure 2, the LSTM-based model is composed with two stacked bi-directional LSTM layers for the classification task. The idea of bi-directional LSTM is straightforward, it duplicates the recurrent layer in the

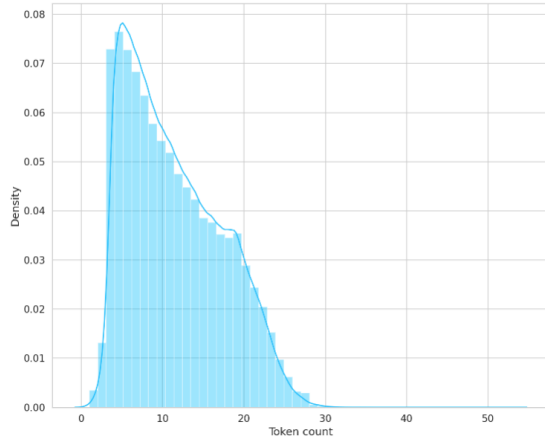


Figure 5: Distribution of Token Length using Tweet Tokenizer (LSTM)

networks, and provides the input sequence as-is as input to the first layer and provides a reversed copy of the input sequence to the second. Bi-directional LSTM performs better in text/speech sequence prediction problems because the context of a sentence is usually useful in interpreting the meaning.

A **Dropout** layer follows each bi-directional LSTM layer, which prevents over-fitting by setting input units to 0 with a configurable frequency at each step during training time.

Dense layers follow the second LSTM-Dropout layers. Dense layers compress the LSTM output to the dimension of emoji classes using kernel weights matrix, and an activation function is applied element-wise. Softmax activation function is used in the final dense layer, which not only maps outputs to [0, 1] range, but also maps each output in a way that the total sum is 1. The output of Softmax is therefore a probability distribution. In emoji prediction and evaluation, we use the numpy argmax function to get the max probability category as the prediction result.

5.4.3 Cloud AutoML

Cloud AutoML needs user to import labeled training dataset in order to use the AutoML service, and no special data pre-processing is required. By default, Cloud AutoML will use 80% of the uploaded dataset for training, 10% for validation, and 10% for testing. See Figure 15 for our imported dataset on Cloud AutoML, and how imported dataset has been split for training, validation, and testing purpose.

5.5 Model Training

5.5.1 BERT-based Model

Since we have formalized our problem as a multi-class classification problem, we selected **Cross Entropy Loss** as the loss function, **AdamW** with learning rate $2e-5$ as the optimizer, 64 as batch size, and 10 epochs to train the model. The model training process was ran on a Google Cloud virtual machine instance (32 vCPU, 120 GB RAM). It took about 5 hours to complete running the all the training steps. See Figure 6 for the training history.

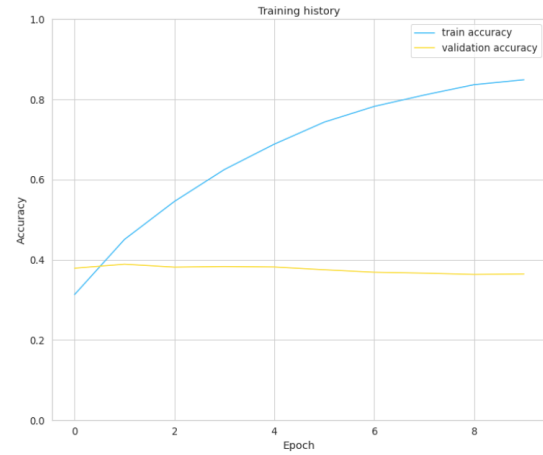


Figure 6: Model Training: BERT-based Model

5.5.2 LSTM-based Model

For LSTM-based model, we used **TensorFlow Adam Optimizer**, and **Categorical Cross Entropy** loss function since this is a multi-label classification task. The training metrics we use is accuracy which is internally using categorical accuracy. See Figure 7 for the training history for 10 epochs in 64 batch size.

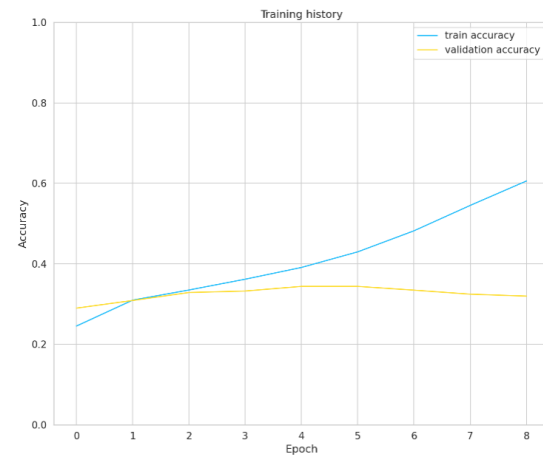


Figure 7: Model Training: LSTM-based Model

5.5.3 Cloud AutoML

The model training process which uses the same dataset for both BERT-based model and LSTM-based model is completed overnight. Once training is done, a notification will be sent to the user via email, and user can go to Google Cloud Console to review the model evaluation report and test the model prediction results online.

5.6 Model Evaluation

5.6.1 BERT-based Model

2589 Tweets split over 15 different emoji classes are used as testing dataset to evaluate the performance of model. The evaluation result on test dataset is 0.36 macro-F1 score and 0.36 accuracy. See Figure 8 for the detailed model evaluation report.

	precision	recall	f1-score	support
backhand_index_pointing_down	0.43	0.46	0.44	182
blue_heart	0.19	0.16	0.17	176
crying_face	0.28	0.31	0.29	167
face_with_rolling_eyes	0.26	0.21	0.23	179
face_with_tears_of_joy	0.24	0.26	0.25	175
flexed_biceps	0.37	0.39	0.38	172
flushed_face	0.32	0.36	0.34	166
folded_hands	0.42	0.42	0.42	170
hundred_points	0.44	0.44	0.44	160
party_popper	0.53	0.51	0.52	173
police_car_light	0.62	0.61	0.62	172
thinking_face	0.50	0.39	0.44	183
thumbs_up	0.36	0.38	0.37	170
two_hearts	0.25	0.25	0.25	171
weary_face	0.27	0.29	0.28	173
accuracy			0.36	2589
macro avg	0.36	0.36	0.36	2589
weighted avg	0.36	0.36	0.36	2589

Figure 8: Test Dataset Classification Report of BERT-based Model

5.6.2 LSTM-based Model

15 emoji classes with over 9200 tweets (10%) are used as testing dataset to evaluate the performance of the LSTM-based model. The evaluation result on test dataset is 0.32 macro-F1 score and 0.34 accuracy. See Figure 9 for the detailed evaluating report.

5.6.3 Cloud AutoML

The same 2589 Tweets split over 15 different emoji classes are used as testing dataset to evaluate the performance of model generated from Cloud AutoML. The evaluation result on test dataset is 0.59 macro-F1 score and 0.59 accuracy. See Figure 10 for the detailed model evaluation report.

6 Analysis

- The model performance for “blue-heart” is very low (F1 score: 0.17, 0.08 in two models),

	precision	recall	f1-score	support
backhand_index_pointing_down	0.35	0.28	0.31	457
blue_heart	0.28	0.04	0.08	594
crying_face	0.29	0.18	0.22	471
face_with_rolling_eyes	0.24	0.42	0.30	720
face_with_tears_of_joy	0.25	0.24	0.25	720
flexed_biceps	0.31	0.24	0.27	552
flushed_face	0.25	0.06	0.10	345
folded_hands	0.42	0.45	0.44	720
hundred_points	0.43	0.43	0.43	720
party_popper	0.51	0.64	0.57	578
police_car_light	0.45	0.64	0.53	454
thinking_face	0.47	0.29	0.36	720
thumbs_up	0.32	0.33	0.32	720
two_hearts	0.27	0.46	0.34	720
weary_face	0.31	0.32	0.32	720
accuracy			0.34	9211
macro avg	0.34	0.33	0.32	9211
weighted avg	0.35	0.34	0.33	9211

Figure 9: Test Dataset Classification Report of LSTM-based Model

	precision	recall	f1-score	support
backhand_index_pointing_down	0.64	0.60	0.62	182
blue_heart	0.59	0.43	0.49	176
crying_face	0.58	0.54	0.56	167
face_with_rolling_eyes	0.57	0.50	0.53	179
face_with_tears_of_joy	0.52	0.51	0.52	175
flexed_biceps	0.56	0.58	0.57	172
flushed_face	0.58	0.57	0.57	166
folded_hands	0.65	0.58	0.61	170
hundred_points	0.57	0.64	0.60	160
party_popper	0.65	0.74	0.69	173
police_car_light	0.66	0.85	0.75	172
thinking_face	0.64	0.63	0.64	183
thumbs_up	0.61	0.55	0.58	170
two_hearts	0.54	0.56	0.55	171
weary_face	0.48	0.55	0.52	173
accuracy			0.59	2589
macro avg	0.59	0.59	0.59	2589
weighted avg	0.59	0.59	0.59	2589

Figure 10: Test Dataset Classification Report of AutoML Model

and it can be easily misclassified with “two-hearts”(as shown in the confusion matrix, 26% of the samples with True Label “blue-heart” is misclassified as “two-hearts”), which is as expected. Both “blue-heart” and “two-hearts” can be used to express love, affection, pleasure, or happiness; “blue-heart” was originally created to show support for Autism awareness, and then it has been used to show trust and loyalty of love to someone. So both emojis can be used in the same context, and it is a challenge for machine to learn the subtleties among various “heart” emojis.

- As shown in the BERT model classification report, “face-with-rolling-eyes” also have a low F1 score (0.23), and we can see from the confusion matrix that “face-with-rolling-eyes” can be easily misclassified with “crying-face”, “face-with-tears-of-joy”, and “weary-face”. “face-with-rolling-eyes” is commonly used to express moderate disdain, disapproval, frustration, or boredom, within a tweet that conveys a negative sentiment, similar to “crying-face”(for frustration) and “weary-face”(for

boredom). “face-with-tears-of-joy” is a very popular emoji, and it is commonly used to underscore a joke, acknowledge a funny comment, or soften a sarcastic remark. So in some context, “face-with-tears-of-joy” can also be used in tweet to soften the sarcasm or hide the negative sentiment.

- We see “face-with-tears-of-joy”, which is a very popular emoji used on social media, is very likely to be misclassified with our models. This is because “face-with-tears-of-joy” can be used to express, soften, or hide emotions in many contexts (e.g., people use politeness as a way of being sarcastic, which does not event match the causality/sentiment surfaced in the text).
- Distinct emojis, like “party_popper” and “police_car_light”, have better F1-score (>0.5 in all models) as they have clear meaning and are not likely to be used interchangeable with others.

7 Conclusion and Future Work

Some tweets in the testing dataset are misclassified by the model, mostly probably because they are very specific to a certain situation, relationship, or culture. It requires a diverse and comprehensive knowledge which currently cannot be processed quite well with natural language understanding.

Some tweets can be used interchangeably in many contexts, for example, “double-hearts” and “blue-heart”. However, our current evaluation method is not smart enough to evaluate that. This problem potentially can be resolved by reformulating the original problem as a multi-label classification problem and by implementing a customized evaluation method (for example, for emojis that are likely to be used interchangeable, if both “double-hearts” and “blue-heart” are predicted with high probabilities, then as long as one of them match the true label in original tweet, we can consider the model prediction result is correct).

For future work, we would like to explore other models and train on larger dataset, and also implement a customized evaluation method to evaluate model performance. We believe additional context information outside of the tweet itself can also help the model to predict the emoji and sentiment better, that being said, more comprehensive dataset generation work is needed in the future. Last but not

least, more development work will be needed in order to deploy the model to be used by a real on-line “Emoji Chat-bot” as a web service, see a very preliminary demo as illustrated in Figure 18.

References

- Ye Bi, Shuo Wang, and Zhongrui Fan. 2020. [A hybrid bert and lightgbm based model for predicting emotion gif categories on twitter](#).
- N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. 2011. [Smote: Synthetic minority over-sampling technique](#).
- Yuxiao Chen, Jianbo Yuan, Quanzeng You, and Jiebo Luo. 2018. [Twitter sentiment analysis via bi-sense emoji embedding and attention-based lstm](#).
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. [Bert: Pre-training of deep bidirectional transformers for language understanding](#).
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. [Effective approaches to attention-based neural machine translation](#). *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [GloVe: Global vectors for word representation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.
- Ralf C. Staudemeyer and Eric Rothstein Morris. 2019. [Understanding lstm – a tutorial into long short-term memory recurrent neural networks](#).
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. [Sequence to sequence learning with neural networks](#).
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#).

Appendix

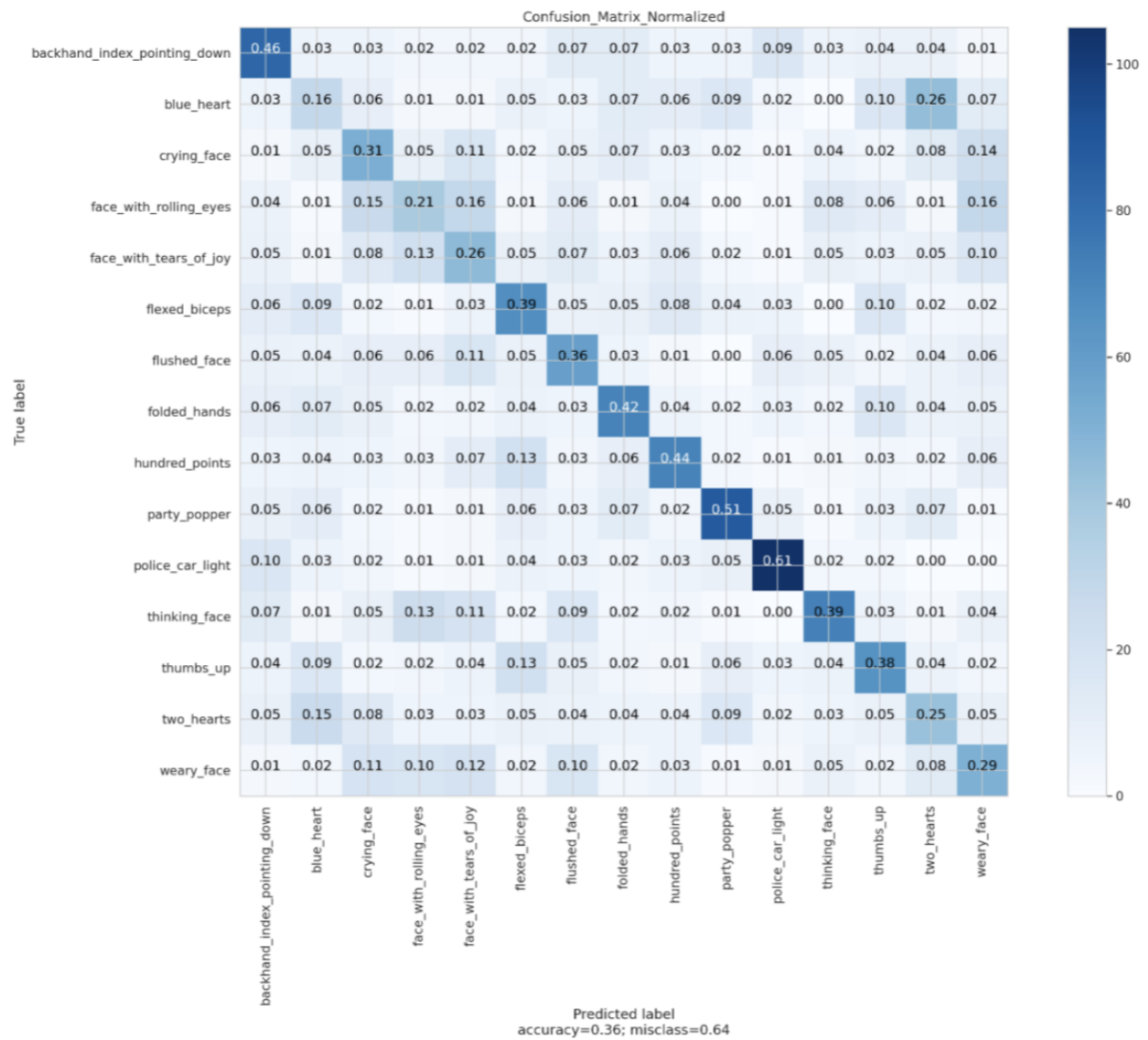


Figure 11: Confusion Matrix of the BERT-based Model

tweet_texts	emoji	predicted_emoji
Niggas lie so much i bet eve aint even eat that fucking Apple	😬	😬
Are you Because you fasho mugged me yesterday	😬	😬
Can you really call him a boyfriend if he has to climb on a chair to change the light bulbs	😬	😬
I am living here amp now There is nothing but the present moment	💙	😬
I have to go out for a bit but Ill be back soon MOUCHES	💕	💕
I know your making money moves	💯	💯
Good morning lovelies Have a blessed day	💕	💙
fellas is it gay to woo your bros hip thrust	😬	😬
Such a thoughtful gift off my auntie Miss you Dad	💙	💕
Thank you We had a great season im excited for our future	💪	💙
I was suppose to watch A Quiet Place w mi amigas but I couldnt make it home on time	😬	😬
To the Press see below This is the type of behavior that has emboldened There is no place for	👉	👉
Listen Im doing this lmao	😬	😬
THANK YOU Sooner Nation What a day amp weekend It takes ALL of us to be great See you in September	💯	🎉
Cutest Freshmen couple Heritage	😬	💙
Legendary gif Love me right mv when he killed everyone in 1 second h	😬	😬
Reply below or quote and please include EXOs tags in your tweet and Vote for for	👉	👉
Have you voted for today Voting for is until May 8 only Keep em votes coming httpst	💙	🙏
how do u even console a depressed friend	😬	😬
Thank you for your acknowledgment	🙏	🙏
My week on Twitter 16 Mentions 875 Mention Reach 34 Likes 27 Retweets 148K Retweet Reach See yours with	🎉	🎉
Im stepping over shit i use to trip off of	💯	💯
Look at the producers they cant mess this up	🙏	😬
gym shit	💪	💪

Figure 12: Sample Original Emojis in Tweets v.s. Predicted Emojis Generated from the BERT-based Model

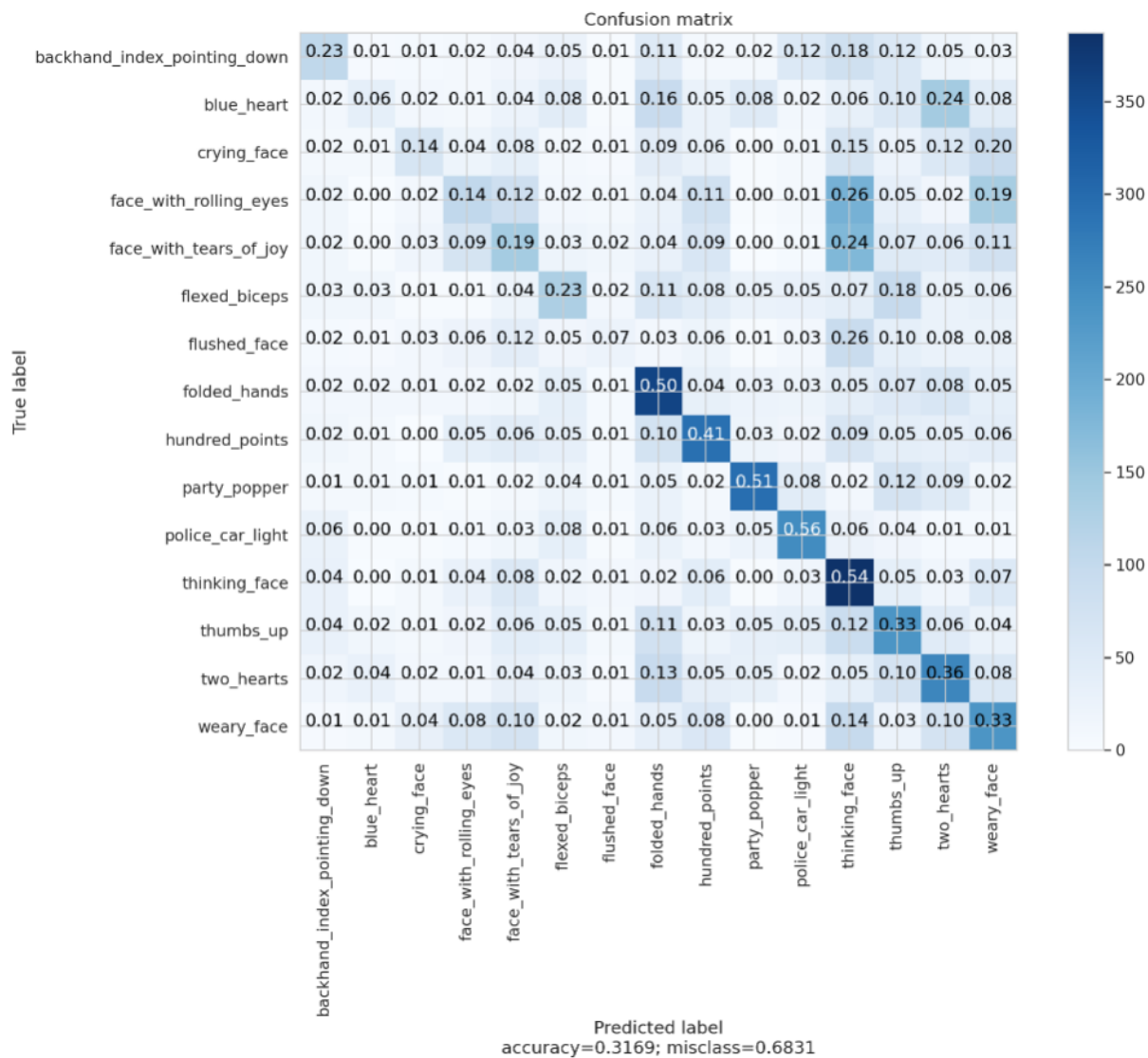


Figure 13: Confusion Matrix of the LSTM-based Model

	tweet_texts	emojis	predicted_emojis	predicted_probability
	Toy boy place is available Details soon on	🎉	👍	0.244556
	Thanks mate all the best for the play offs	👍	👍	0.659613
	Updated 2903 9PM KST Melon 4 Genie 3 Naver Bugs OUT Mnet 1 Soribada	📺	📺	0.986358
	i dont Fw nobody i just be doing me Fr	💯	💯	0.589659
	Phew Enjoy the sir Cracking job done	👍	👍	0.876758
	so much foreign money into RNC and trump and GOP campaigns	📺	😬	0.245482
	It was a BBC programme Thats probably why its so good	👍	👍	0.341807
	Always and I live here too lol	💙	😬	0.272030
	MOCK DRAFT MONDAY See who the experts are favoring at pick 6	📺	📺	0.613125
	Lol sorry had to tweet almost that whole song Cardi took words out of my mouth amp put them in this song	😬	😬	0.376253
	while yall out being hoes Im home being faithful	🙏	💯	0.514710
	BTCUSD Digital Currencies Stage A Comeback via	📺	🎉	0.247864
	Movie at 715 Should be ready in an hour Still need to shoot by my Bruvha house	😬	😬	0.179174
	Ayooo Who tf is this	😬	😬	0.232329
	My week on Twitter 1 Mention 2 Retweets 126K Retweet Reach 4 New Followers 1 Tweet See yours with	🎉	🎉	0.998074
	All Glory To GOD	🙏	🙏	0.801402
	I want some hibachi	😬	😬	0.751314
	This should be illegal whew	😬	😬	0.230436
	PARENT ALERT Like Follow and Retweet for a chance to a set sixmonth subscription to Epic Magazine ONE WINNER	📺	📺	0.977081
	If you want your COUNTRY BACK get out and vote these people OUT Indiana WILL DO our part Mark it down Donnelly is	📌	📺	0.400934

Figure 14: Sample Original Emojis in Tweets v.s. Predicted Emojis Generated from the LSTM-based Model

IMPORT	ITEMS	TRAIN	EVALUATE	TEST & USE
All items	51,780	Filter table		
Labeled	51,780	<input type="checkbox"/> Items		Labels
Unlabeled	0	<input type="checkbox"/> Thats how I know Im growing		hundred_points
Training	41,430	<input type="checkbox"/> Even tho you said fuck me its never fuck you		hundred_points
Validation	5,175	<input type="checkbox"/> Thanks mate will do my best		thumbs_up
Testing	5,175	<input type="checkbox"/> Yall my mom met Dale Murphy		face_with_rolling_eyes
		<input type="checkbox"/> Were equally terrified and intrigued by this bad boy		thinking_face
Filter labels		<input type="checkbox"/> Greek Freak against Brad Stevens		thinking_face
backhand_index_poi...	3,452	<input type="checkbox"/> Im going on a diet wanna feel better about myself		flexed_biceps
blue_heart	3,452	<input type="checkbox"/> Jameson Taillon threw a changeup so dirty that it moved twice		flushed_face
crying_face	3,452	<input type="checkbox"/> What do you mean were almost out of treats x		crying_face
face_with_rolling_ey...	3,452	<input type="checkbox"/> With the BREAKING NEWS of Michael Cohens office being raided amp Mueller is involved be prepared to...		police_car_light
face_with_tears_of_j...	3,452	<input type="checkbox"/> gotta bother baby before she leaves me for the night		face_with_rolling_eyes
flexed_biceps	3,452	<input type="checkbox"/> No more double txting nomore trippin nomore nhn it is what it is mindset		flexed_biceps
flushed_face	3,452	<input type="checkbox"/> Bae really 1 month I can say hun		hundred_points
folded_hands	3,452	<input type="checkbox"/> Wishing a very happy birthday to the GOAT Have a great year ahead amp Thank you for being who you ar...		party_popper
hundred_points	3,452	<input type="checkbox"/> I so agree to this sometimes just let things be peoplejust let things be can someone once think abou...		folded_hands
party_popper	3,452	<input type="checkbox"/> Share This amp Take Time To Vote		backhand_index_pointing_down
police_car_light	3,452	<input type="checkbox"/> Lets help on decreasing the gap REPLY with your favorite lyrics of EXOs song with the hashtags HWAIT...		police_car_light
thinking_face	3,452	<input type="checkbox"/> Wow Your view is normally bang on but mate we have been embarrassing tonight and Spurs head n s		flushed_face
thumbs_up	3,452	<input type="checkbox"/> Proud proud proud of you my friend		flexed_biceps
two_hearts	3,452	<input type="checkbox"/> Life is all about up and down d best feeling is always at d top		flexed_biceps
weary_face	3,452			

Figure 15: labeled dataset uploaded to Cloud AutoML









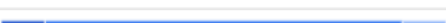





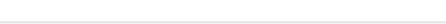
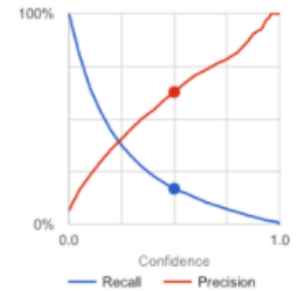
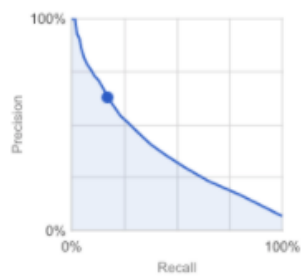
Labels ↑	Annotations	Train	Validation
backhand_index_pointing_down	 3452	2762	345
blue_heart	 3452	2762	345
crying_face	 3452	2762	345
face_with_rolling_eyes	 3452	2762	345
face_with_tears_of_joy	 3452	2762	345
flexed_biceps	 3452	2762	345
flushed_face	 3452	2762	345
folded_hands	 3452	2762	345
hundred_points	 3452	2762	345
party_popper	 3452	2762	345
police_car_light	 3452	2762	345
thinking_face	 3452	2762	345
thumbs_up	 3452	2762	345
two_hearts	 3452	2762	345
weary_face	 3452	2762	345

Figure 16: Training and Validation Dataset Split for Cloud AutoML

All labels

Test items	5,175
Precision ?	62.84%
Recall ?	17.02%

Use the slider to see which confidence threshold works best for your model on the precision-recall tradeoff curve.
[Learn more about these metrics and graphs.](#)



Confusion matrix

This table shows how often the model classified each label correctly (in blue), and which labels were most often confused for that label (in gray). If you have more than 10 labels, this table only includes the 10 labels with the most incorrect predictions.

True label	Predicted label	backhand_index_pointing_down	blue_heart	crying_face	face_with_rolling_eyes	face_with_tears_of_joy	party_popper	police_car_light	thinking_face	two_hearts	weary_face
backhand_index_pointing_down	49%	4%	5%	2%	7%	7%	14%	6%	4%	2%	
blue_heart	6%	29%	7%	2%	5%	9%	7%	2%	29%	3%	
crying_face	4%	4%	43%	6%	9%	1%	3%	6%	10%	13%	
face_with_rolling_eyes	5%	2%	5%	33%	17%	-	3%	11%	2%	23%	
face_with_tears_of_joy	7%	2%	5%	13%	38%	3%	2%	14%	5%	11%	
party_popper	3%	4%	2%	-	1%	73%	13%	1%	3%	1%	
police_car_light	9%	2%	2%	2%	0%	5%	77%	2%	1%	-	
thinking_face	6%	2%	4%	15%	10%	2%	3%	50%	2%	6%	
two_hearts	5%	17%	8%	2%	3%	10%	4%	2%	40%	10%	
weary_face	2%	5%	14%	15%	11%	1%	0%	3%	8%	40%	

Figure 17: Model Evaluation Report generated from Cloud AutoML

```
input_text = "I feel so upset that this small town in Italy hit so bad by COVID-19..."
pred_score, pred_class = automl_predict_emoji(prediction_client, model_full_id, input_text)
```



```
input_text = "Take it easy and we will go through this together."
pred_score, pred_class = automl_predict_emoji(prediction_client, model_full_id, input_text)
```



```
input_text = "Thank you sweet!"
pred_score, pred_class = automl_predict_emoji(prediction_client, model_full_id, input_text)
```



```
input_text = "TRUMP Administration was Not a Presidency IT WAS A CRIME SPREE"
pred_score, pred_class = automl_predict_emoji(prediction_client, model_full_id, input_text)
```



```
input_text = "Hey guys You can make a donation to the hospital here"
pred_score, pred_class = automl_predict_emoji(prediction_client, model_full_id, input_text)
```



Figure 18: A rudimentary demo of a Emoji Chat-bot by leveraging the deployed model from Cloud AutoML