Emojify This: The GSB CS Experience
Ruchir Shah, Shammi Quddus, Srish Sundram
{ruchirs, shammi, srishit}@stanford.edu

Predicting

We use a dataset of tweets with emojis to train a model that predicts the five most likely emojis for a given tweet. Tweets with emojis are used to train a bidirectional LSTM using GloVe Twitter embeddings.

Related Work
This project was inspired by the work done by Rahwan et. al at the MIT Media Lab in the Deepmoji project, where they analyzed 12.8 billion tweets to detect emotions and sentiment using emojis. Their 128 unit LSTM model had a top 1 accuracy of 77%.

Data
We used a large, publicly available set of 20kN Twitter data from November 2017. Prior to our analysis, we did a substantial amount of processing to get the data into a readable format. This included removal of following:
- all irrelevant metadata like tweet date, location, userID, etc
- URL and "RT" retweets and usernames
- non-English tweets and non-alphanumeric characters
- any tweets that did not contain emojis.

Then indexed the emojis to assign numbers as Y values for the dataset. Our dataset contains 1494 unique emojis. We then tokenized the sentences and passed each word through a pre-trained Twitter GloVe model to generate embeddings.

The complete data set contains 4M tweets, but our final, cleaned, data set includes approximately 58k9 tweets. With the frequency of most common tweets capped at 500.

We put aside 10% data for Test and trained on the remaining 90%.

LSTM Model

When calculating accuracies, we used "Top-1" accuracy, only counting a data point as accurately predicted if the top returned softmax percentage was equal to the correct emoji.

Model Hyperparameters
We experimented with two different neural network models: a Sequential baseline and a bidirectional LSTM structure, both implemented in Keras. The LSTM model was built using source code from CS230.

Our baseline model has two fully connected layers, one hidden layer and one output layer. The hidden layer uses a relu activation function with 64 hidden nodes. The output layer uses a sigmoid activation and contains 12 nodes. We used a cross entropy loss function with stochastic gradient descent over 100 epochs with a batch size of 100.

Our bidirectional LSTM has an LSTM layer with 128 nodes and a tanh activation followed by a 90% dropout layer, followed by a second bidirectional LSTM layer also with 128 nodes, then an additional 30% dropout layer, and lastly a fully connected softmax activation layer. We keep the top five probabilities from the softmax layer.

Lessons
- The bi-LSTM model improves accuracy significantly over the baseline model.
- Commonly used emojis such as heart constituted more than 25% of the emoji and by guessing heart everytime, the model was correct 95% of the time.
- Dataset has to be balanced for model to predict tweets actually connected to meaning rather than learning the most statistically efficient way to do things.

References

Sample Outputs
- CS230 is my favorite class. 🎓❤️❤️❤️❤️
- I love Stanford. 🎓❤️❤️❤️❤️
- Can we meet for coffee? 🍊❤️❤️❤️❤️

Future Work
While we were able to validate some key findings in this study, there remains significant room for growth. As a team, we were limited by time and computational resources. As we pursue this project in the future, we plan to examine the following:
1. Larger data set — try to improve our accuracy by making sure sufficiently large number of tweets per emoji.
2. Attention Layer (a) — fine-tuning the model further
3. Generative Model — Creating tweets from emojis — could we use GAN to synthesis sentiment-focused text?