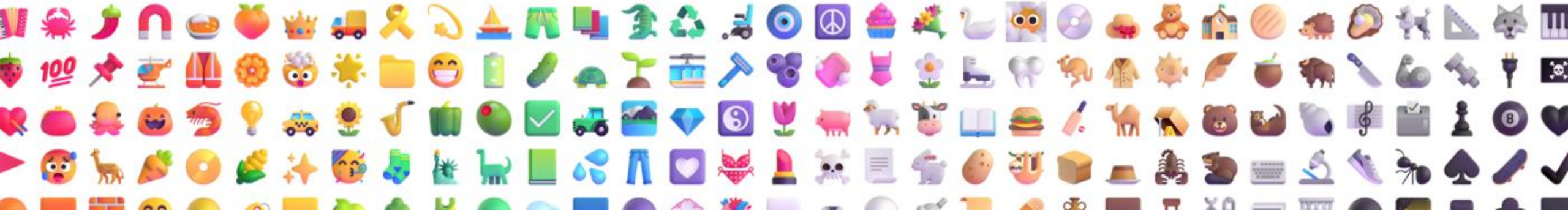
Final Project Review
[Twitter emoji prediction analysis]

Yuan Tian 0033657574
Hao Shen 0031470864
Bonan Kou 0030080798
Siyu Yao 0029599604



Introduction

- What is the problem you are trying to solve?
  Emoji prediction with tweets
- Why is it important?

  Increasing use of emojis in real life and avoiding using misleading emojis in the context
- Novel aspects

  1. Explore performance of different types of ML models

  2. Leverage sentiment as an extra feature when training the model

  3. A dynamic emoji prediction user interface

Research Question

- In this project, we proposed and examined three research questions.

  1. How do today's popular ML models perform for emoji prediction?

  2. Will additional features generated by extra models increase the performance?

     (i.e. will sentiment enhance the performance of emoji prediction?)

  3. How does the user feel when he or she can acquire a dynamic emoji feedback when typing a message.

Modeling

Neural Network with Bag of Words (Baseline)

- the frequency of each word is used as a feature for training a classifier
- Map the raw data into a matrix
  - NN size: 972*600*300*100*20 (3 hidden layers)
- We trained the hidden layers with different activation function:
  - first four hidden layers ReLU() and and the last two Sigmoid()

Neural Network with BERT

- BERT combines context embedding and several Transformers, plus it's bidirectional
- The vector BERT assigns to a word is a function of the entire sentence, therefore, a word can have different vectors based on the contexts
  - embedded each whole sentence to a vector with 768 dimension
  - a 3-layers neural network

Modeling

RNN(baseline)

- designed to work with sequential data
- uses the previous information in the sequence to produce the current output
  - we use the pre-trained word embedding and represented the whole sentence using the last output hidden state of the network

GRU

- The workflow of GRU, is the same as the RNN but the difference is in the operation and gates associated with each GRU unit
  - use similar hyperparameter as RNN
- The GRU is like a long short-term memory (LSTM) with a forget gate, but has fewer parameters than LSTM, as it lacks an output gate

Modeling

LSTM

- Similar to GRU. But there are 2 more gates 1)forget gate 2)output gate.

LSTM (with NN capturing hidden states)

- To explore if the previous hidden states are helpful to improving the performance

BiLSTM

- a sequence processing model that consists of two LSTMs
  - the size of long term memory and the short term memory are doubled

Transformer

- better than most of the architectures because they totally avoid recursion
  - we implement our model with pytorch
    - 2 hidden layers, 4 heads, and a dropout constant of 0.5 to avoid overfit.
    - use BERT to encode twitters as 768-bit vectors and use the BERT embeddings as input for the transformer model

Experimental setup:

- What are the specific experiments you are proposing

Predict with BERT with NN instead of Bag of Words; Predict with RNN; Predict with GRU; Predict with LSTM; Predict with LSTM and NN; Predict with BiLSTM; Predict with Transformer

- Explain the settings:

Dataset: From GitHub, Tweets with corresponding emojis

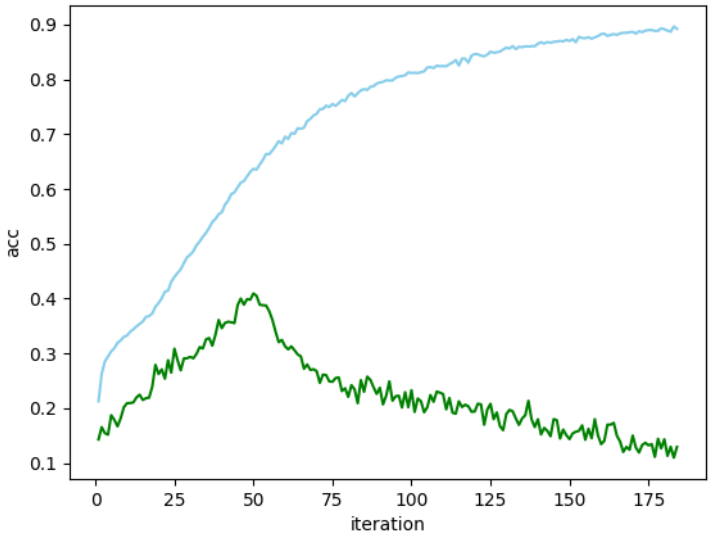
Baseline: Neural network with Bag of Words (Easiest to implement, Not including sequential information)

Models: From easy to hard based on the complexity

Experimental results

Some key points for transformer model:

- Best accuracy comes at 49th epoch, slower than other models.
- Achieve max accuracy of 0.4
- Overfit happens after ~50 epoches.



Result for transformer model

	Highest acc on train set	Highest acc on test set
NN with BoW	21.67%	20.07%
NN with BERT	97.17%	31.10%
RNN	23.42%	21.17%
GRU	41.14%	27.22%
LSTM	95.45%	30.03%
LSTM (all states)	99.97%	31.25%
BiLSTM	97.21%	29.98%
Transformer	98.3%	40.00%

Other graphs and details are shown on the reports since there is not enough space here

Discussion

- What did you learn?

  Extra input does not necessarily increase performance.

  Complex models may still have their limits.

  Complex models requires more training epochs.

- What changed compared to the proposal and why?

  Dataset: Make our own → Use existing dataset

  Focus: Application → Research

  Models: NN & BiLSTM → Multiple models

  Sentiment: Pure text → Use sentiment as extra input

  GUI: Complex → Simplified

Thank you !