STATISTICAL METHODS IN ARTIFICIAL INTELLIGENCE

TEXT CORRELATION WITH EMOJI

SEMESTER PROJECT

TA MENTOR

Satyam Mittal

TEAM MEMBERS

Akshat Maheshwari, 20161024

Kalpit Pokra, 20161134

Kushagra Nagori, 20161032

Kartik Garg, 20161219

TABLE OF CONTENTS

TABLE OF CONTENTS 1

PROBLEM STATEMENT 2

GOAL 2

INTRODUCTION 3

CURRENT STATE OF ART 3

DATASET 3

PART 1 - TWEET CLASSIFICATION BASED ON EMOJI 6

- PROBLEM STATEMENT 6
- APPROACH 6
- MODELS 7
- RESULTS 7

PART 2 - EMOJI PREDICTION FOR A SENTENCE 7

- PROBLEM STATEMENT 7
- APPROACH 7
- MODELS 8

RESULTS 10

PREDICTIONS AND GRAPHS 10

PROBLEM STATEMENT

There are 2 parts to the problem statement of the project:

1. Given a set of tweets, classify the tweets according to the emojis.
2. Given a sentence, check the sentiment analysis and insert appropriate emojis in sentence.

GOAL

1. **Classification of tweets:** Given a dataset of tweets, build various classification models so that the tweets are classified according to the correct emojis based on the One Hot Encodings of the tweets.
2. **Emoji Prediction for a sentence:** Need to suggest emojis based on the given sentence or phrase by analyzing the sentiment and predict the appropriate emoji for that sentence or phrase.

INTRODUCTION

Over the years , technology has significantly changed the way people communicate and interact with each other. The advancement in technology and the coming of the applications like Whatsapp, Facebook, Twitter, etc are the reason for this. These applications allow people to interact with each other by sending messages, or publically posting what they are feeling.

Emojis (or emoticons) are a very good way to express one's feelings in the graphical form.

For example,

- a smile can be denoted by: 😀
- disappointment can be denoted by: 😞, etc.

CURRENT STATE OF ART

We did read two research papers which were similar to our task. Below are referenced papers and approaches used in those papers -

1. https://www.aclweb.org/anthology/S18-1064 -
    a. 500k Tweets scraped from twitter.
    b. In this paper logistic regression approach is used to predict possible emoji.
    c. Max Accuracy they got in their dataset is 22%
2. Multimodal Emoji Prediction -
    a. Instagram posts with minimum one image was used in this modal.
    b. ResNets (Conv. Net) to analyze and extract feature vectors
    c. Fasttext for textual data

d. LSTM/RNN for emoji prediction
e. They created table of F1 score and Precision and tried to explain which emoji was easy to predict and which was tough to predict.

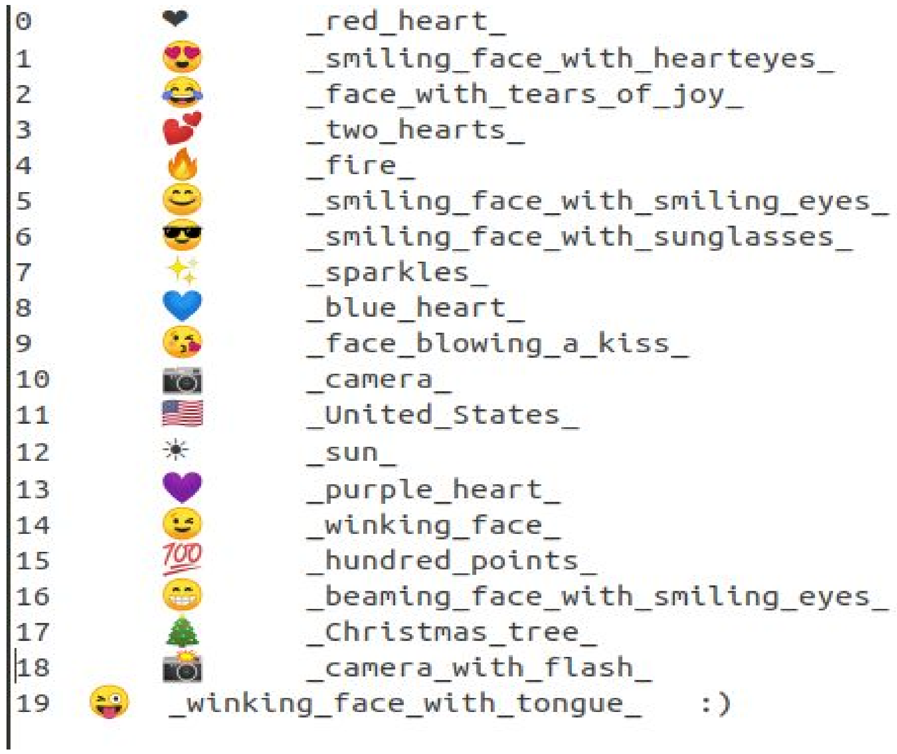
DATASET

1. For the first part, we have used the Twitter Emoji dataset from the following link: Link
    a. The dataset contains a total of 500k tweets in English. These tweets were retrieved from Twitter using API's. The dataset is in the form of 3 files, 1 containing the tweets, one containing the emoji label corresponding to tweet on that index, and the last one contains the emoji and their corresponding labels.
    b. There are a total of 20 emojis that have been used in the data.
2. For the second part, we used the dataset from the following link: Link
    a. We had to use a second dataset instead of the Twitter Emoji dataset, because we were facing some errors in the use of that dataset for sentiment analysis, such as:

        i. The @ mentions and the #-tagged elements need to be removed from the dataset tweets because creation of the word2vec models for these words is very difficult as mostly these are words that are not present in general vocabulary. That is a very laborious work, and need the exact locations till they are extending. For example, there are various Named Entities, like names of persons, places or organizations that are not present in the word embedding models, and so the creation of word vectors is not possible for them and it starts giving errors.

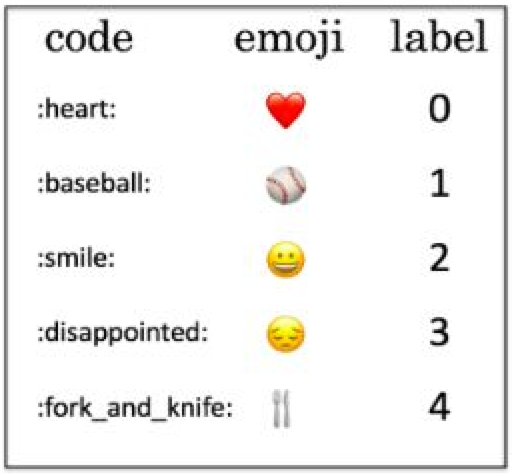
        ii. Various words that are present in the vocabulary are written as abbreviations or spelling mistakes that cannot be detected by the models. For example, the word *slam* has been written as *slamm…*, phrases like *good night* are abbreviated as *gn* or *good ni8*, etc.

b. The new dataset is a quite small dataset, which consists of around 500 sentences and emoji labels corresponding to the those sentences. The test set contains around 50-60 sentences that we can test on. The emoji set for this dataset uses only 4 types of emojis, but that can be extended to more emojis if quite sufficient cleaned up data is available.



```
0       ❤       _red_heart_
1       😍      _smiling_face_with_hearteyes_
2       😂      _face_with_tears_of_joy_
3       💕      _two_hearts_
4       🔥      _fire_
5       😊      _smiling_face_with_smiling_eyes_
6       😎      _smiling_face_with_sunglasses_
7       ✨      _sparkles_
8       💙      _blue_heart_
9       😘      _face_blowing_a_kiss_
10      📷      _camera_
11      🇺🇸      _United_States_
12      ☀       _sun_
13      💜      _purple_heart_
14      😉      _winking_face_
15      💯      _hundred_points_
16      😁      _beaming_face_with_smiling_eyes_
17      🎄      _Christmas_tree_
18      📸      _camera_with_flash_
19  😜  _winking_face_with_tongue_   :)
```

Emojis for the Twitter Emoji Dataset



code	emoji	label
:heart:	❤️	0
:baseball:	⚾	1
:smile:	😀	2
:disappointed:	😞	3
:fork_and_knife:	🍴	4

Emojis for new dataset

PART 1 - TWEET CLASSIFICATION BASED ON EMOJI

PROBLEM STATEMENT

Part-1 of the project is a classification problem. Given a dataset consisting of the tweets and the emojis corresponding to them, we need to classify the tweets based on these emojis.

APPROACH

The approach for this classification problem are as follows:

1. **Preprocessing:** The first step is data pre-processing. Pre-processing includes elimination of noises like the @ mentions, # tags, html tags, urls, non-alphabetic characters, etc.

2. **Tokenization:** The second step is tokenization, i.e., breaking of the preprocessed tweets into individual words, and generate a bag of words.
3. **Bag of Words:** This model is basically all the words that are coming in the tweets, and their frequencies. This bag of words list is then used to get all the unique keys / words that are coming in the tweets to create a vocabulary.
4. **Feature Vector Generation:** Using the vocabulary, we are generating mappings of the words with their indices. This mapping is then used to create a One-Hot Encoding, which will be used as a feature vector for our models.
5. **Model:** Finally after getting the feature vector, we will be building models and calculating their accuracies. These accuracies will be further used for comparison of the various classification models.

MODELS

- Decision Tree
- Linear SVM classifier
- Logistic Regression
- LSTM

RESULTS

- Decision Tree Classifier: 20.7%
- Linear SVM Classifier: 22%
- Logistic Regression Classifier: 26.7%
- LSTM Classifier: 38%

PART 2 - EMOJI PREDICTION FOR A SENTENCE USING SENTIMENT ANALYSIS

PROBLEM STATEMENT

Part 2 of the project is a prediction problem. Given a sentence or statement or a phrase, predict the appropriate emoji by analyzing the sentiment.

APPROACH

The approach for this part uses the method of sentiment analysis. According to Wikipedia, Sentiment analysis refers to the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information.

In this approach we will be using the word2vec model for calculating the sentiment value corresponding to a word and a sentence.

The steps that are used in this approach are as follows:

1. **One-Hot Vector creation:** We are converting the output labels (both training and testing) into One-Hot vectors so that they are suitable for training the models. The one-hot vectors are of the size y_shape[0]X24, because we are using a total of 24 emojis in the emoji dictionary for the predictions. These one-hot vectors are basically the diagonal matrices that contain 1 at the columns which correspond to the emoji index, and 0 at all other columns.
2. **Loading of the word embedding vectors:** We have used the word embedding model for the generation of the word2vec model. We have used the 50-D Stanford Glove Embedding Vectors. The read_glove function returns us 2 dictionaries, one is ***word_to_index***, which is the dictionary

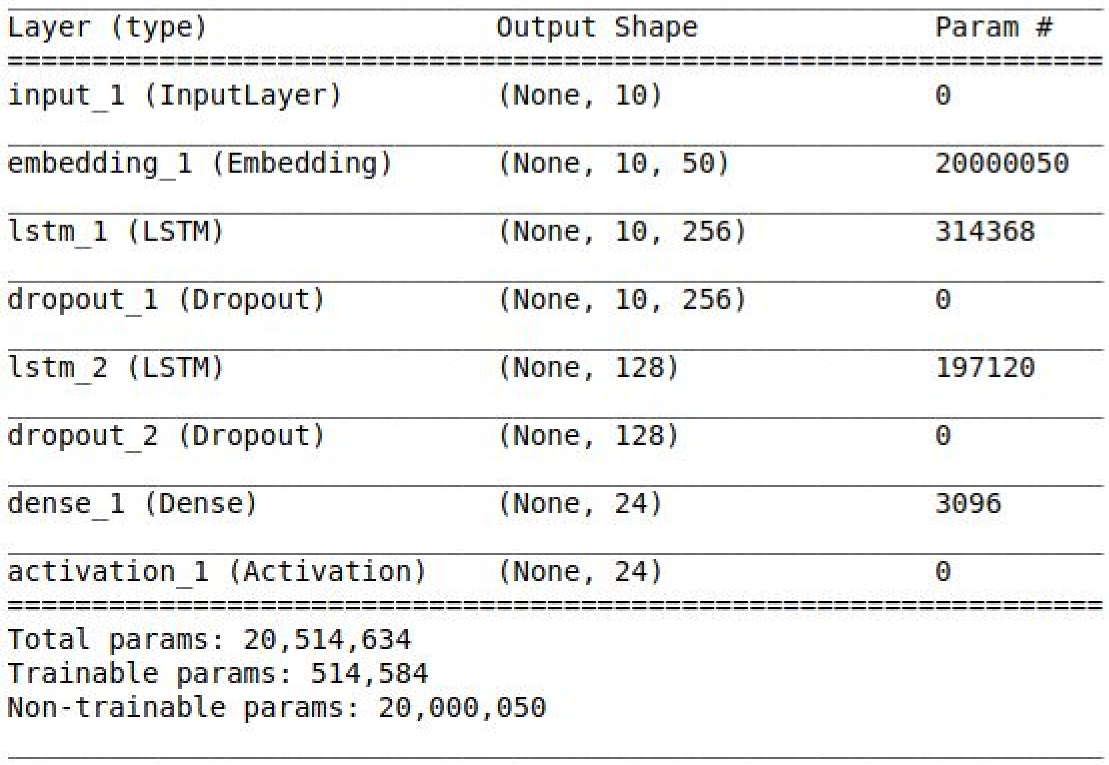
mappings from words to their indices in the vocabulary, and the other is the ***word_to_vec_map,*** which is dictionary mapping of words to their Glove vector representations.

3. **Model creation:** After getting all the parameters we need for the model, we use them to create our models, and then using these models, we can predict the emojis corresponding to the input sentences.

MODELS

We have used 2 models in this part:

1. **Iterative Model (Non-neural net model):** In this model, we are using the average of the word embeddings of all the words in a sentence. After getting this average word embedding, we are computing the forward pass, computing the cost and then backpropagating to update the softmax's parameters.
2. **LSTM Model:** Unlike the iterative model, in this model, we are using the *Embedding()* layer in Keras by initializing it with the 50-D Glove vector embeddings. In this model, instead of the average word embeddings for the sentence, we are converting the sentences to lists of indices using the word_to_index dictionary. After that, we are using 2 LSTM layers with 256-D and 128-D hidden layers and dropouts of 0.5. After that, we are using a dense layer with the softmax activation. For the optimizer, we are using the Adam's Optimizer with categorical cross-entropy loss. The model summary is as follows:



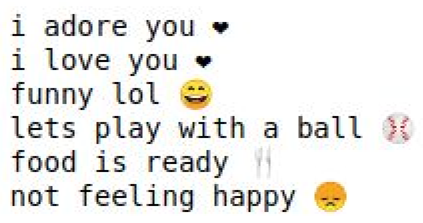
```
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 10)                0
_________________________________________________________________
embedding_1 (Embedding)      (None, 10, 50)            20000050
_________________________________________________________________
lstm_1 (LSTM)                (None, 10, 256)           314368
_________________________________________________________________
dropout_1 (Dropout)          (None, 10, 256)           0
_________________________________________________________________
lstm_2 (LSTM)                (None, 128)               197120
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 24)                3096
_________________________________________________________________
activation_1 (Activation)    (None, 24)                0
=================================================================
Total params: 20,514,634
Trainable params: 514,584
Non-trainable params: 20,000,050
_________________________________________________________________
```

RESULTS

- **Iterative Model**
    - Training Accuracy : 0.9587301587301588
    - Test Accuracy : 0.8071428571428571
- **LSTM Model**
    - Training Accuracy : 0.98 (approx)
    - Test Accuracy : 0.8171428571428571

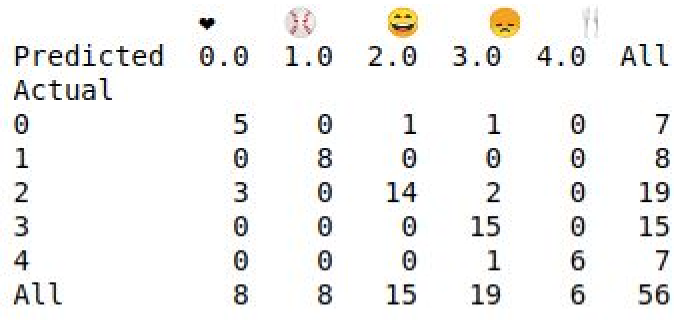
PREDICTIONS AND GRAPHS

- **Iterative Model**

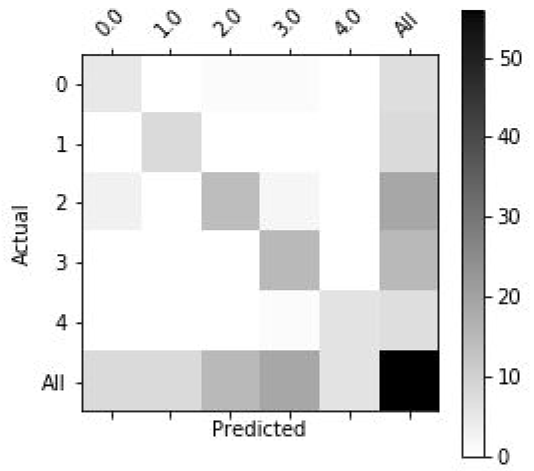


```
i adore you ❤
i love you ❤
funny lol 😄
lets play with a ball ⚾
food is ready 🍴
not feeling happy 😞
```

Predictions on some random sentences

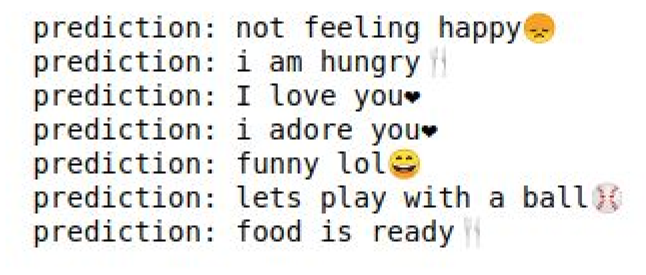


```
               ❤     ⚾     😄     😞     🍴
Predicted    0.0   1.0   2.0   3.0   4.0   All
Actual
0              5     0     1     1     0     7
1              0     8     0     0     0     8
2              3     0    14     2     0    19
3              0     0     0    15     0    15
4              0     0     0     1     6     7
All            8     8    15    19     6    56
```



Confusion matrix for the iterative model

- **LSTM Model**



```
prediction: not feeling happy😞
prediction: i am hungry🍴
prediction: I love you❤
prediction: i adore you❤
prediction: funny lol😄
prediction: lets play with a ball⚾
prediction: food is ready🍴
```

Predictions on random sentences

REASON OF USING NEW DATASET INSTEAD OF THE TWITTER DATASET

We used the new dataset, although smaller, but cleaned up for the second part of the project, because the accuracy we were getting on the Twitter dataset was around 0.20. Since this accuracy is too low so we felt a need to use more refined and cleaned dataset.

Thus, we searched for new dataset that was more cleaned up.

TEAM MEMBERS CONTRIBUTIONS

- DATA SCRAPING: Kushagra Nagori, Kalpit Pokra
- DATA COLLECTION: Kushagra Nagori, Kalpit Pokra
- REPOSITORY MAINTENANCE: Akshat Maheshwari, Kartik Garg
- MODEL COVERAGE AND SELECTION: Akshat Maheshwari, Kartik Garg
- CLASSIFIER MODELS CODINGS: Kushagra Nagori, Kartik Garg, Akshat Maheshwari, Kalpit Pokra

- SENTIMENT ANALYSIS MODELS: Akshat Maheshwari, Kalpit Pokra, Kushagra Nagori, Kartik Garg
- OUTPUT ANALYSIS: Kushagra Nagori, Kalpit Pokra
- RESEARCH PAPERS: Akshat Maheshwari, Kartik Garg