Studying the Emoti(c)ons of Twitter Users

CS 145 Data Mining
Professor Sun Yizhou

Group Number: 21
Group Name: Tweet Emojis 💩
Group Members: Mark Tai (504399847), Nathan Zhang (304605928), Jai Srivastav (604605757), Fan Hung (804319873), Michael Xiong (404463570)

Final Report
December 10, 2017

1 Abstract

Our goal was to find and predict the general happiness of individual tweets. We approach this problem by first training our neural network to detect positive or negative emoji's in tweets. This will allow us to examine all tweets and make hypotheses on the happiness index of users. In our work, we extend beyond studying individual tweets and also study reactions to certain words that would generally garner an either deeply positive or negative emotion. We hope that our analysis would give us results can that could eventually be used to simulate general public response concerning any tragic or happy event, potentially ranging from presidential scandal to a holiday celebration.

2 Introduction

In today's world with applications like Twitter and Facebook, people consistently share their thoughts with the general public. Because of this, there is a significant amount of data that can be used to describe the emotions of people on an everyday basis. Many of their tweets use emojis or emoticons that give a strong indication of the happiness level of the words that they are using. We hope that this emoji-filled data will give our machine learning algorithms insight on what words or phrases generally constitute happiness or sadness.

On Twitter, it is common for users to respond to certain events that occurred and give their own personal opinion followed by a hashtag and the name of the event. This allows us to easily mine data for responses to a certain event. We hope to explore the happiness of both these specific response tweets and the overall happiness of the users after a certain event.

Twitter data has some unique problems. In tweets, syntax is much more fluid, and often, full statements are not written out. There is also a vocabulary of abbreviations and slang that is not found in typical language. Misspellings are also extremely common. One of our main solutions to this variability is the use of neural networks for classification as well as representing tweets as vectors built from context.

The rest of our paper will explain the process of how we approached quantifying our data and how we will use machine learning to analyze it. Finally, we will provide our results and analysis on the data.

3 Problem Definition and Formalization

In our project, we developed an algorithm to determine the sentiment of a certain tweet. Using this, we identified the overall happiness levels of users on Twitter. Later, we hoped to see user sentiments towards certain topics, such as Trump and UCLA.

Mood detection is a classification problem. We are trying to classify tweets as either happy or sad, and the simplest way to approach this is setting extreme happiness as a 1 and extreme sadness as a 0. However we will not treat this as a binary problem. We will instead use a continuous scale from 0 to 1 to express the level of happiness that exists.

Another option that we had considered was a binary representation of emotions but with a wider range of emotions. We would have true or false for happiness, sadness, disgust, anger, love, etc. We preferred our option continuous output would be easier to visualize in our final graphs.

We also hoped to find how users felt about certain people or just general topics. By analyzing thousands of tweets with words like “morning” or “Trump”, we found how the general population feels about these topics.

4 Data Preparation

To find and prepare data for our happiness predictor, we used emojis on Twitter to classify happiness and sadness, using the resulting data to train a neural network. Data was collected by a crawler that scraped English tweets, and classified them with the existence of emojis representative of the two types of emotions. These labeled tweets were then sanitized for links, stop words, and spelling mistakes. Additionally, we both removed capital letters and stemmed the tweets, reducing words to their most basic forms. This included conjugating verbs, removing plurals, and other normalizations. While this was not an essential process, due to word2vec being robust to various forms of words, we stemmed in order to be as complete as possible.

After we had a normalized repository of tweets, we converted the entire corpus of data to doc2vec vectors. This gave us an embedding of the Twitter vocabulary that would allow us to convert any test Tweet into a feature vector as input into our neural net. This was an incredibly powerful tool as it recognized relations in context. For example, “king” and “queen” would be very similar, and would differ by approximately the vector “woman”. Also, vectors for tweets such as “how are you doing” and “what are you doing” were close in cosine distance and far from the vector for “I like hamburgers”. In other words, the meaning of our tweets is captured within the vectorized representation of our words.

The vectorization process, roughly speaking, runs as follows. All the words seen are represented by a one-hot encoding, and all the tweets are represented by a one-hot encoding. A neural network is then trained to predict which words will appear, given the tweet. While building the network, a specific layer in the neural is designated to be the vector representation of the tweet. The output of this layer is restricted in number of nodes to match the dimensionality of the vectors, as requested (in our case, we use 100 dimensions). Then, when the vector for a new tweet is requested, new words and the new document are added to the on hot code, and after a few gradient descent steps for neural network training are run, the output of the vector representation layers is returned.

Our resulting dataset consisted of roughly 60,000 tweets converted to doc2vec vectors supported by our embedding, as well as labels determined by the emojis found in the original tweets. We made sure to remove the emojis from the vector representations of tweets so as to avoid teaching the neural net the correlation between emojis and emotion. We separated our data into testing and training sets.

5 Method Description

5.1 Data Preprocessing

Our data preprocessing methods are described above, and resulted in a doc2vec embedding used to generate our feature vectors labeled as either positive or negative examples. We simply randomly partitioned this data into training and testing sets in order to evaluate our model. In addition to the 60,000 tweets used to train and evaluate, we also crawled additional sets of data containing specific keywords. These sets included “Christmas”, “Trump”, “LAFD”, among other words. We used these datasets to test our final trained model. For example, we expected the datasets with tweets containing “Christmas” to be generally happy, while the sets with tweets containing “LAFD” to be largely negative.

5.2 Neural Network Training

We chose a neural network because of its flexibility. Our neural network takes a single vectorized tweet as an input and outputs a predicted happiness score from 0 to 1 (0 being completely unhappy, 1 being completely happy). In training, the labels of the data was binary, but we would like a continuous predictor of happiness vs. unhappiness rather than a binary classifier, so we did not threshold the sigmoid activation function (except when comparing against other binary classifiers and computing confusion matrices and quality measures, for these, for simplicity, we used a threshold of 0.5). This is because a continuous value is more useful in ranking many tweets in their happiness, whereas binary classification is less informative.

To train our neural network, we used the scikit-learn library to interpret all of our data. Scikit-learn allowed us to customize the number of fully connected hidden layers, number of nodes in each, and other gradient descent related hyperparameters we would want to change. For simplicity,we did not vary gradient descent parameters. Gradient descent technique used by sklearn in training did not use a static learning rate, so we assume that simplification is safe. We tried different network layouts and found out that two layers of 30 hidden perceptrons gave us the best results. The data and more details for this step can be found later in 6.1.

5.3 Keyword-Based Happiness Analysis

After training and selecting an optimal neural network model, we applied it to extract knowledge from additional tweets. An additional 11 sets of data were crawled, each associated with a specific keyword. The keywords used were:

1. LAFD - We expected the recent fires to produce significantly negative tweets.

2. UCLA - We expect mixed tweets about UCLA due to their inadequate response to the fire.

3. Wedding - We expect a happy event to receive a high happiness index.

4. LAPD - We expect that LAPD would be surrounded by negative tweets.

5. Trump - We expected high variance in the tweets, due to the divisive nature of politics.

6. Dead - We expect death to be strongly associated with sadness.

7. Bitcoin - Due to the recent bitcoin crash, we expect largely negative tweets.

8. Morning - We included this term as a relatively neutral term.

9. Coco - Coco was a very well received animated film, and we expect positive tweets.

10. Christmas - We expected Christmas to be universally positive.

11. Thanksgiving - We expect Thanksgiving to be just like Christmas.

12. Thor - We do not expect a correlation for this term.

13. Reputation - This is a neutral term, so we expect no trend with this data.

14. Terry Crews - We expected variance in these tweets, due to the mix of his celebrity status, and the sexual assault allegations surrounding Terry Crews.

15. Justice League - This is a popular movie, so we expect a positive happiness index.

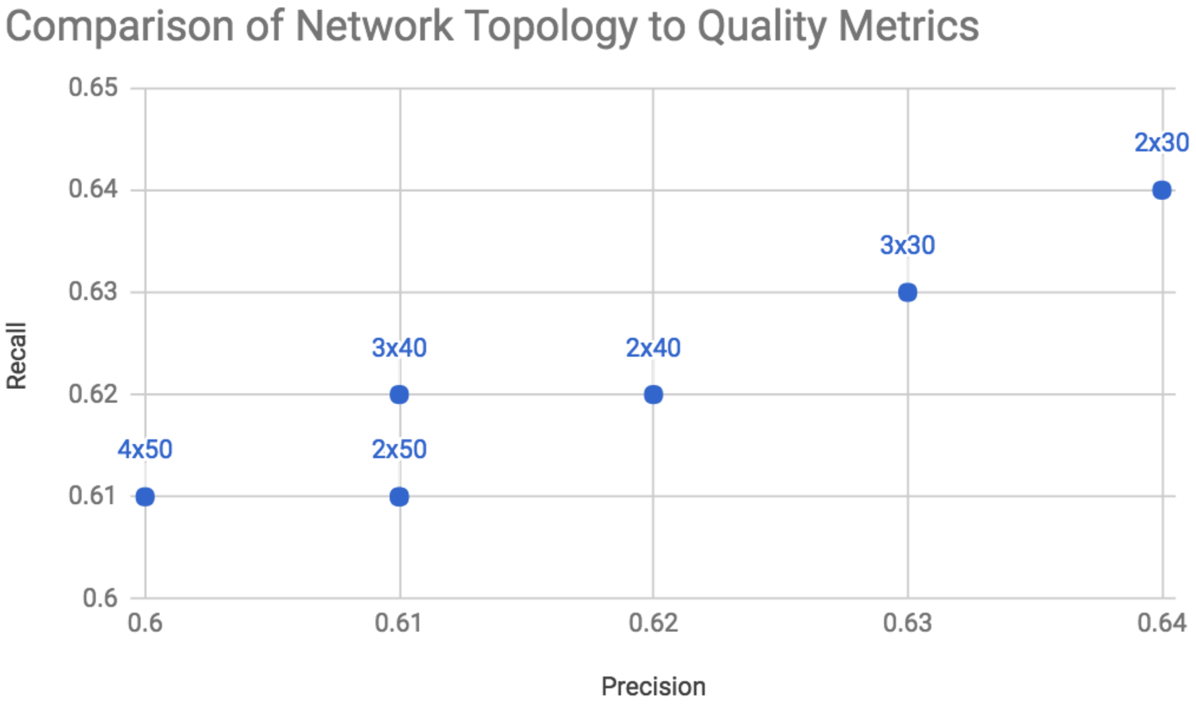
16. Birthday - Like wedding, this is a happy event, and we expect a similarly happy rating.

6 Experiments Design and Evaluation

6.1 Hyperparameter Selection and Neural Network Training

The main goal of our project was to predict whether a happy face or sad face is more applicable to a given tweet. To evaluate this, we measured both precision and recall for many neural nets trained. Through trial and error, we tuned various hyperparameters to arrive at an optimal model. These hyperparameters included activation function, number of hidden layers, and number of neurons per layer.

In our testing, we found that activation function did not significantly impact the results, and there was no clear correlation between activation function and classifier error. Therefore, we chose ReLu (AKA Rectifier, or ramp function), as it provided marginally better results than the others tested, which included sigmoid, and hyperbolic tangent. While the activation function did not make a significant impact, the network topology did. Varying both the number of hidden layers, as well as the number of neurons within each layer significantly impacted our results. Below is a graph showing the resulting precision and recall for different network topologies:

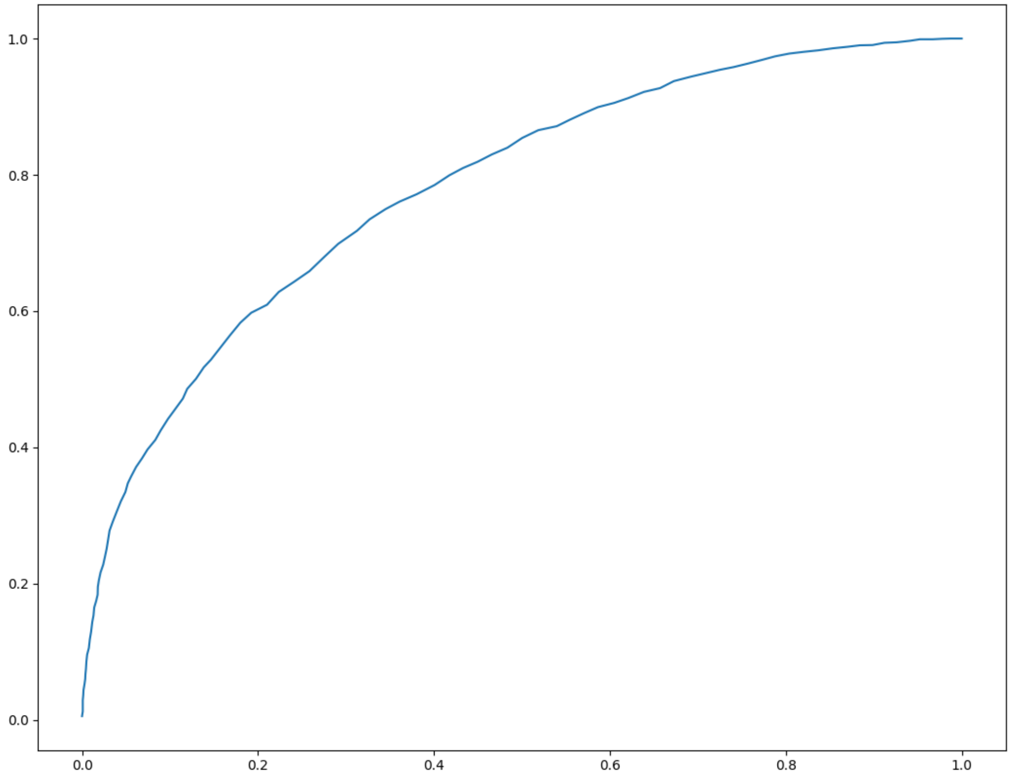


Here, we represent as the number of fully connected layers x number of nodes per layer. For example, network of 4 fully connected hidden layers with 50 nodes per layer is represented as as “4x50”.

We can see that a deeper and larger neural net does not result in a better classifier. In fact, our deepest and widest neural net actually performed the worst. Instead, we found that a smaller neural net with two layers and 30 units per layer performed the best in both performance and recall. This is likely due to the fact that larger neural nets tend to overfit the data, learning a relationship that is much more complex than reality.

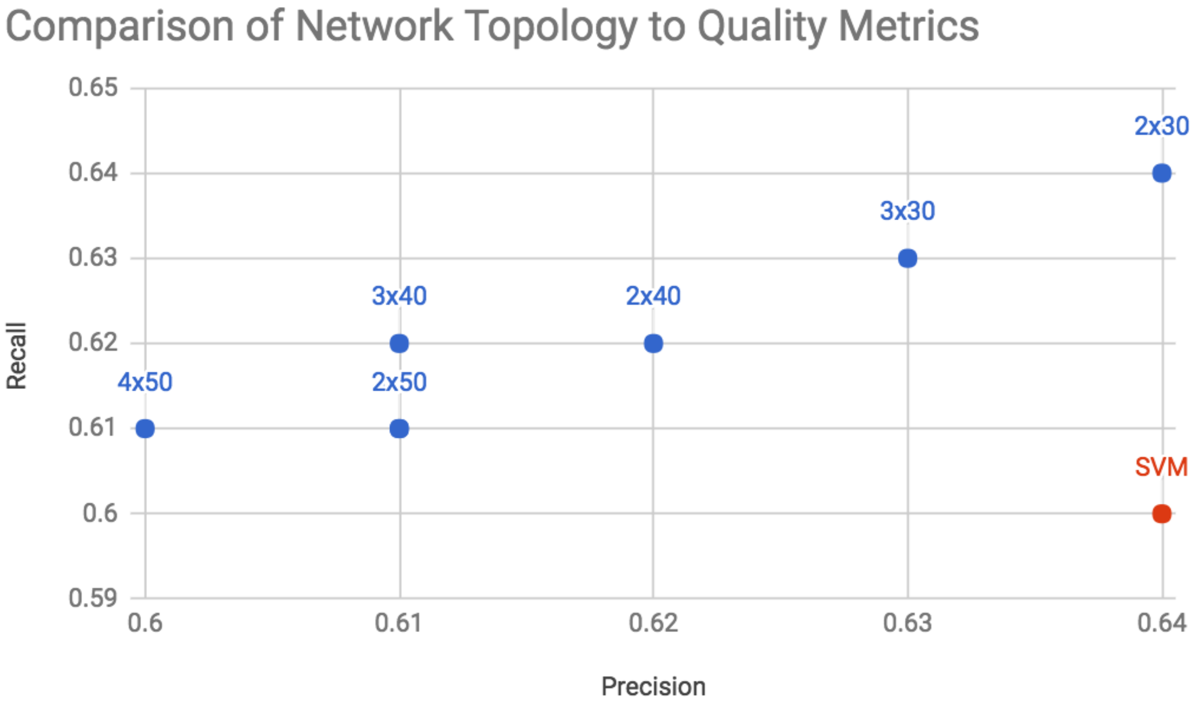
After selecting our optimal model, we further evaluated it by plotting a ROC curve for different values of our classification threshold. As can be seen, the classifier clearly does noticeably better than random chance, which would be indicated by a diagonal line with slope 1.

ROC Curve of Optimal Classifier



6.2 Comparison Against Other approaches:

In order to obtain a relative comparison of our model, we trained a SVM with a radial basis function kernel on the same dataset. Below, we can see that an SVM trained on the same data produces weaker results than our neural net.



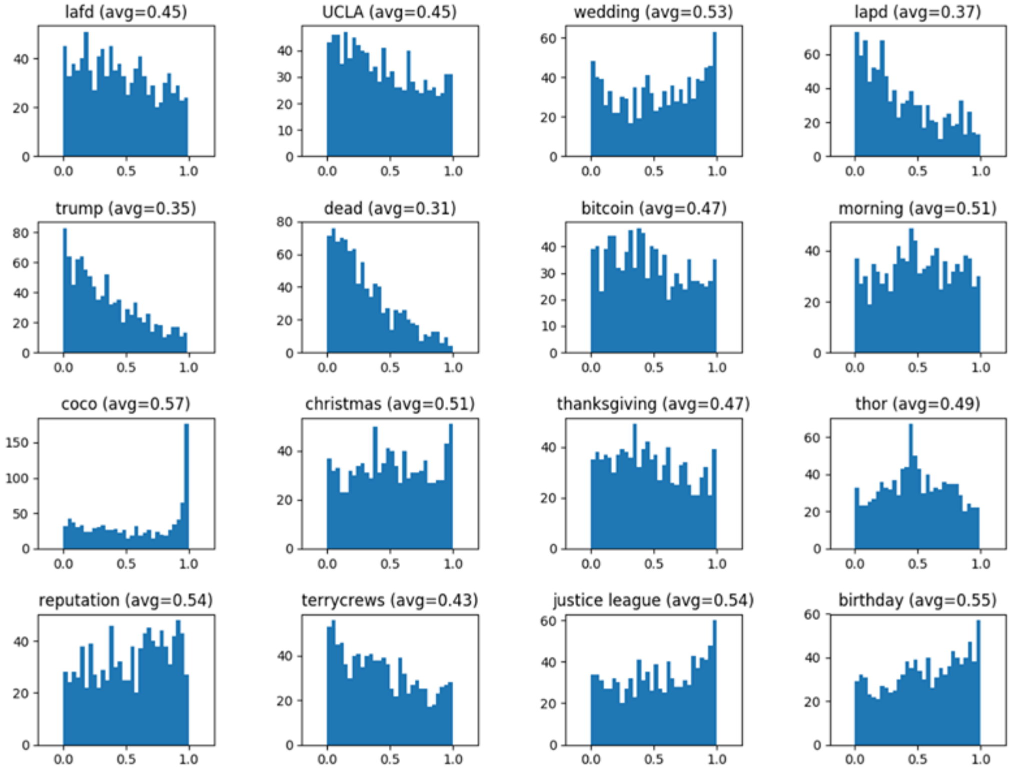
In fact, the SVM performed worse than the weakest neural net topology and activation function. In a problem such as classification of natural language, we would expect that the flexibility of a neural net to excel, while the linear constraints of SVM to struggle in this setting.

Also, not only was the SVM a weaker performer than the neural net, it was also significantly more difficult to train. Beyond 10,000 data samples, the model training began to scale very poorly, resulting in extremely long training times, over 10 minutes to train on our 60,000 examples, while the neural network took approximately 1 minute. Furthermore, because the complexity of training a SVM is worse than quadratic in the number of examples, the performance costs of SVM would be even higher for larger datasets. In contrast, the training complexity of a neural net is approximately linear in the number of examples (assuming a small topology, which we used).

6.3 Computation of Happiness of Twitterverse with Keywords

Finally, as an application of our model, we crawled an additional 1000 tweets containing one of 16 selected keywords. Below, we show histograms of the data, with the frequency plotted against the happiness index for each term.

Histograms of Sentiment of Query Terms



Applying our neural net classifier to these datasets reveals some very interesting trends. As might be expected, “LAPD”, “LAFD”, and “dead” exhibit clear negative patterns. These terms are often used in conjunction with negative events, and are therefore associated with sadness. Strangely, “UCLA” and “Terry Crews” also exhibited negative trends. While this is unexpected, it can be explained by recent events, such as the UCLA fires, UCLA final exams, and the sexual assault allegations made by Terry Crews. An interesting extension of our work would look at how these histograms changed over time.

Another interesting observation can be made about the histogram of Donald Trump. Certainly a divisive figure, we expected great variance in the histogram for “Trump”. However, it exhibits a clear negative trend. This may be indicative of the generally liberal nature of the Twitterverse, where most users tend to disapprove of Donald Trump, while relatively fewer express support.

Terms with positive trends included “birthday”, “wedding”, “justice”, and “Coco”. All of “birthday”, “justice”, and “wedding” are intuitively associated with happy events, and received similarly happy ratings. Shockingly, vast majority of tweets were extremely positive regarding the movie Coco, indicating that our neural network was very confident that a tweet containing the word “Coco” was overwhelmingly positive. This can be explained by the relative uniqueness of the term. Therefore, there would be very low noise for this term, allowing our classifier to draw a very clear conclusion about this term. Similar to “Terry Crews” and “UCLA”, it would be interesting to monitor the histogram of Coco over time, as the movie becomes less popular. However, given the time constraints, we were not able to verify this.

The term “Bitcoin” was fairly neutral, with a slight negative shift. This can be attributed to the recent bitcoin crash, causing bitcoin buyers to lose money. We would expect the trend to remain similar, but orient slightly positive in the case of a bitcoin rise. Given more time, exploring how these trends vary over time in cases such as Bitcoin could be very revealing.

Another surprising result were the terms “Christmas” and “Thanksgiving”. Being holidays, we expected largely positive tweets, yet these were both neutral, very similar to “morning”, and “reputation”, which we expected to be neutral. Overall, there was no clear trend for these terms.

We also looked at the most positive and negative tweets for each of our query terms and found many pretty prototypical positive and negative tweets. However, there were also tweets here that show that using emojis for sentiment has some flaws. Emojis may have pretty complicated interpretations, and sentiment itself is not always clear cut. For example, the “thanksgiving” term’s negative tweet example may be very well be interpreted by a human as tweet with positive sentiment, as it thanks the writer’s friends and family for their support.

Most Negative and Positive Tweets by Query Term

Query Term	(sentiment) and example most negative tweets	(sentiment) and example most positive tweets
lafd	(0.00025) returning home wildfire information fd nf (0.00537) sending thoughts prayers affected fires california please stay safe venturacounty venturafire	(0.99064) gang thanks lafd (0.99004) fortunate los angeles pros heroes working round clock save l
ucla	(0.00019) ucla stole twice today making improvements (0.00231) michigan wolverines overcome point deficit beat ucla ot	(0.99912) need go ucla lonzo (0.99555) welovela ucla wolverines overcome point deficit beat ucla ot sportsroadhouse
wedding	(0.00101) lmao wedding sh** messy rhoa (0.00124) crazy years now might ur wedding ur kids birthday rn just regular sh***y day	(0.99908) wedding santorini gem via marryme greece (0.99869) alright since everyone's getting engaged married least can invite wedding open
lapd	(0.00083) vcr malcom gave speech	(0.97580) amazing friend lapd k officer

	funeral man killed lapd years ago still se	m scared death d
trump	(0.00011) trump tweets unhinged way cnn getting date wrong yesterday please remember something else happened	(0.97773) according aides trump watches hours tv every day regularly fails show work even afte
dead	(0.00074) walking dead stressing	(0.98147) dead fish celebration christian screaming one top fears
Bitcoin	(0.00055) everyone turns trader since bitcoin numbers	(0.99860) switched bitcoin cash low fees bulls**t free
morning	(0.00155) sunday morning routine	(0.99934) waking every morning wondering re coming cold just wake now
coco	(0.00665) cried watching coco	(0.99995) coco amazing movie go watch ll cry happiness
christmas	(0.00079) hi happy holidays m lonely hoping someone will tweet back ive single yrs daughter needed long	(0.99952) m trying bake christmas cookies loml wya
thanksgiving	(0.00009) xxox happy thanksgiving everyone m thankful family friends thank endless support l	(0.99681) football game played future get picked way home thanksgiving
thor	(0.00042) will silence accord shall thor snarls harshly sla (0.00103) thing thor ragnarok delightful love delighted m sick endless dr	(0.99156) spotted truly fantastic thor fanart russo's today
reputation	(0.00876) december boygroup brand reputation rankings	(0.99880) brand reputation male idol groups month december wanna one bts exo seventeen https
terrycrews	(0.00004) men need hold men accountable via	(0.99490) keep going
Justice league	(0.00003) main cast members justice league just round table plenty chairs	(0.99878) latest coco justice league repeat box office nothing new movienews movietvtechgee
birthday	(0.00238) happy birthday great coach mentor friend	(0.99843) happy birthday favorite bbboy go still love

7 Related Work

While we limited our work to classifying individual tweets, and extending this capability to tweets surrounding specific events or people, there still exists many areas of exploration for future work. For example, analyzing tweets from specific individuals could reveal trends or events in their life. This type of analysis would be much more fine grained that the high level general trends we explored in our project. We would be able to see how their mood was affected over time, and how they expressed that in their tweets.

In addition to more finely analyzing individuals' tweets, we could also increase our granularity of classification. In this experiment, we implemented binary classification of happy vs. sad, but many other emojis exist, and are indicative of other emotions. For example, confused emojis may be slightly correlated with sadness, but could potentially be better classified as a discrete emotions. These changes would have required more extensive crawling, and more complex models, which were outside of the scope of our work.

8 Conclusion

We successfully achieved our goal of building a neural network to identify sentiment in tweets. In addition to this, we were able to apply the neural network to analyze tweets containing specific keywords to reveal trends of users in the Twitterverse.

Some other analysis we had hoped to do, which was described in our midterm report, included analyzing certain user trends in happiness and overall Twitterverse change in sentiment from tragic events. However, we were unable to accomplish this given our current resources because the Twitter API only allowed us to go back seven days. Both of these tasks required a large amount of time-based analysis and we realized that only having access to the last 7 days of data made it essentially impossible to gain meaningful results. They would however, be possible if we slowly accumulated data over time, and then organized and plotted them.

Because of incredibly powerful Python libraries which were already built for us, we were able to quickly build a neural network and make meaningful discoveries on user sentiments on Twitter. With more time and access to more and better data, we believe that our project could become a powerful predictor of the reaction of humans to certain events.

9 References and APIs

- Doc2vec
    - http://arxiv.org/pdf/1405.4053v2.pdf
- Word2vec
    - https://www.tensorflow.org/tutorials/word2vec
    - https://arxiv.org/abs/1301.3781
- Gensim (big ML library that includes word2vec and doc2vec)
    - https://radimrehurek.com/gensim/models/word2vec.html
    - Includes other word vector embedding strategies
- Another word vectorizer
    - https://fasttext.cc/
- CMU's tweet parser
    - http://www.cs.cmu.edu/~ark/TweetNLP/
- Stanford's probablistic parser
    - https://nlp.stanford.edu/software/lex-parser.shtml
- Other twitter sentiment analysis APIs and approaches
    - http://www.ijcaonline.org/research/volume125/number3/dandrea-2015-ijca-905866.pdf
    - http://www.ijcaonline.org/research/volume125/number3/dandrea-2015-ijca-905866.pdf
    - https://www.tweetsentimentapi.com/
    - https://www.softwareadvice.com/resources/free-twitter-sentiment-analysis-tools/
- From Slides:
    - http://www.hlt.utdallas.edu/~kirk/publications/robertsLREC2012_2.pdf
    - https://mislove.org/twittermood/
    - Johan Bollen et al., Modeling Public Mood and Emotion: Twitter Sentiment and Socio-Economic Phenomena, ICWSM'11