Twitter Sentiment Analysis with Emoji and Emoticon Embedding

Haozheng Ni
hn2318@columbia.edu

Chuqi Yang
cy2478@columbia.edu

Somya Singhal
ss5348@columbia.edu

Abstract

Twitter is a rich source of data for sentiment analysis, opinion mining and many other tasks. One notable feature of twitter data is the usage of emotion tokens such as emojis and emoticons. Intuitively such token express ones feeling regardless of the language used, and therefore they could be very helpful in many tasks listed above. However, previous research mainly focused on pure text and neglected the role of emoji/emoticons. In this work, we trained emoji and emoticon embeddings and developed machine learning and deep learning models on them. The best model achieves 74% accuracy and 73% F1 based on Parrots classification of emotions. We also showed that embedding trained from standard description is superior to the one trained from our data.

1 Introduction

With the rise of social media, emojis become more and more prevalent. Oxford Dictionary (2015) selected 'the face with tears of joy' 😂 as the word of year 2015, and it is estimated that over 10% of Twitter posts and more than 50% of text on Instagram contain one or more emojis in 2015 (Cruse, 2015). Since such emotion tokens could increase the precision and nuance in current fast speed communication, the popularity of emojis does not confine to certain regions or languages. Researchers show that emojis are widely seen in tweets posted by users from USA, UK, Italy, Spain and many other countries (Barbieri et al., 2016a).

Emoticon can be seen as an old-versioned emoji, and admittedly it can be hard to read sometimes because one need to tilt head to the left. However, some emoticons such as `:)`, `:-(` are still often used, and many new emoticons like `<( ^.^ )>`, `<(^_^<)`, `<(o_o<)` become popular as well.

There is no doubt that the trend of emojis and emoticons will keep increasing. Therefore, in order to better understand the conversations on social platform like Twitter, Reddit and Facebook, it is very important for researchers in natural language processing and social media analysis to acknowledge the importance of such emotion tokens, and develop methods to incorporate them in current study frameworks.

Previously the combination of word embedding and deep learning models achieved remarkable success in natural language processing tasks, one example can been seen in (Abdul-Mageed and Ungar, 2017). Therefore a natural extension would to obtain emoji and emoticon embeddings in similar ways. We tried two methods to train embeddings in this project: 1) from standard description provided by Unicode organization[1] 2) from the data (just treat emojis like normal words).

In this work, we seek to prove that emojis and emoticons indeed help to better understand the context. More specifically, we first collect data from twitter mixed with recent and historical data (see more in data section). Next, we train emoji and emoticon embeddings as introduced above. We in-

[1] http://www.unicode.org/emoji/charts/full-emoji-list.html

vestigate this problem as a multiclass prediction based on Parrots classification of emotions (Parrott, 2001), and we adopt the distant supervision that uses hashtags as labels of emotions (Mintz et al., 2009). Then we apply deep learning models including Long-Term Short Memory (LSTM) and some machine learning models together with word embedding released by Google [2].

Overall, we achieve the following targets: 1) we display the usefulness of emoji and emoticon embedding in sentiment analysis. To our knowledge, our work is the first one proving the effect of both emoji and emoticon. 2) By comparison, we demonstrate the advantages of training from standard description than from context. 3) We develop powerful deep learning models that beat many machine learning models.

The remainder of the report is organized as follows: We first discuss some related work in this topic in section 2, and then we explain the process of collecting and cleaning data in section 3. We then discuss our models in section 4, and present results in section 5. Future work will be discussed in section 6.

2 Related Work

2.1 Emotion Prediction with Distant Supervision

Sentiment analysis has been a popular topic in recent years, however, it still remains a very challenging task partly due to lack of labeled data. Traditionally researchers need to manually annotate the sentiment or make use of Amazon Turk. Apparently it is costly and inefficient, and thus its hard to obtain a large amount of labeled data. With the rise of twitter and many other social media, researchers are able to acquire huge amount of data with implicit labels. For example, many researchers turn to distant supervision, which uses hashtag in twitter as a natural label for sentiment (Mintz et al., 2009). This is quite reasonable because the hashtag given by users will be more reliable than annotation by others which might wrongly assign the sentiment.

One notable progress in sentiment analysis by distant supervision is made by Abdul-Mageed and Ungar (2017). Not only situating in psychological theory of emotion, they showed that distant supervision indeed capture the sentiment. Specifically, the annotators assigned a tag in relevant, irrelevant to specify whether they agreed the hashtag label or not. The overall agreement is 61.37%, and if they confined the hashtag appear in the last quarter of tweet, the agreement became 85.43% and 90.57% if appeared in the end.

However, this could become tricky when a tweet contains multiple emotional hashtags. They mentioned that duplication was made to reserve one hashtag per tweet, but this could incur data leakage in model training and evaluation of models. We realize this problem and further constrain that replicates from the same tweets should be assigned to the same partition during train/dev/test split of data.

2.2 Emoticon/Emoji embedding

A number of recent studies actually have been conducted to make use of emoji embeddings.

Barbieri et al. (2016a) performed emoji embedding based on Twitter data. They first applied a series of filters on Twitter data, and then skip-gram model (Mikolov et al., 2013) of different parameters were applied on the data with emojis. The quality of emoji embedding was evaluated by human. They first let 8 human participants label the similarity score of every pairs of words, and compare with the cosine scores calculated by emoji embedding. The t-SNE(Maaten and Hinton, 2008) visualization showed some clusters of emojis, which supports their final embeddings.

However, their approach can be controversial in many aspects. Firstly, the data they use to train emojis is limited by size. Different communities or languages may have subtle difference, and meaning of emojis is undertook fast development. Thus embedding trained from small corpus lack of enough generalized power. Moreover, the correctness of embeddings are hard to verify.

Eisner et al. (2016) avoided such problem by using an emoji embedding trained from description in the Unicode emoji standard table which was referred previously. Though it seem to lose interaction in context, they showed that embedding trained by this method improved the model performance than Barbieri et al. (2016b)'s work in a sentiment analysis task. This method is promising in the sense it in-

[2] https://code.google.com/archive/p/word2vec/

cludes a wider range of emojis in embedding and is very easy to implement.

Besides the research on emoji, the relationship between emoticons and sentiment polarity on social media has also been examined. Wang and Castanon (2015) addresses the following queries: prevalence of emoticons on Twitter, context of usage of emoticons, meaning of emoticons and contribution of emoticons to sentiment analysis. Through many surveys, they conclude the following: 1) a few emoticons are strong and reliable signals of sentiment polarity and one should take advantage of them in any sentiment analysis; 2) a large group of the emoticons conveys complicated sentiment hence they should be treated with extreme caution.

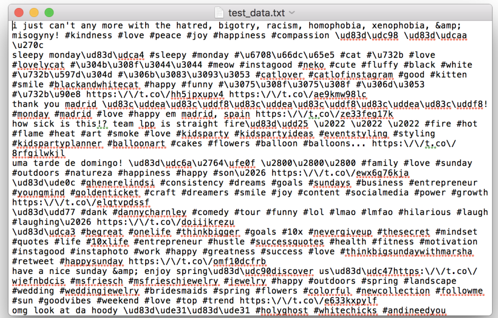
3 Data

The data of this project is from two sources, history tweets and real-time tweets.

We download history Twitter data from https://archive.org; the data is in JSON format, so we iterate over all the data and extract the tweets that contain at least one of hashtags we designed, which will be talked later, and at least one of emotions or emojis as raw data.

For real-time tweets, we create a listener in Python and use Twitter's official API to retrieve real-time tweets. the mechanism of Twitter's official API is: by setting keywords beforehand, Twitter's servers will send information of all real-time tweets which contains at least one of the keywords to the listener as a stream in JSON format.

So we first set up the keywords to be all the hashtags we designed. After obtaining the information of the real-time tweets, we extract the text of the tweets from the provided information in JSON format, then cast the string to Unicode and use regular expression to detect whether there is an emoji or emotion in the text, as long as there is at least one emoji or emoticon in the text, we will keep the tweet's text as a raw data.



```
test_data.txt
i just can't any more with the hatred, bigotry, racism, homophobia, xenophobia, &amp;
misogyny! #kindness #love #peace #joy #happiness #compassion \ud83d\udc98 \ud83d\udcaa
\u270c
sleepy monday\ud83d\udca4 #sleepy #monday #\u6708\u66dc\u65e5 #cat #\u732b #love
#lovelycat #\u304b\u308f\u3044\u3044 #meow #instagood #neko #cute #fluffy #black #white
#\u732b\u597d\u304d #\u306b\u3083\u3093\u3053 #catlover #catlofinstagram #good #kitten
#smile #blackandwhitecat #happy #funny #\u3075\u308f\u3075\u308f #\u306d\u3053
#\u732b\u90e8 https:\/\/t.co\/hh5jpxupv4 https:\/\/t.co\/ae9kmw98lc
thank you madrid \ud83c\uddea\ud83c\uddf8\ud83c\uddea\ud83c\uddf8\ud83c\uddea\ud83c\uddf8!
#monday #madrid #love #happy em madrid, spain https:\/\/t.co\/ze33feg17k
how sick is this!? team lpp is straight fire\ud83d\udd25 \u2022 \u2022 \u2022 #fire #hot
#flame #heat #art #smoke #love #kidsparty #kidspartyideas #eventstyling #styling
#kidspartyplanner #balloonart #cakes #flowers #balloon #balloons... https:\/\/t.co\/
8rfgilwkjl
uma tarde de domingo! \ud83d\udc6a\u2764\ufe0f \u2800\u2800\u2800 #family #love #sunday
#outdoors #natureza #happiness #happy #son\u2026 https:\/\/t.co\/ewx6g76kja
\ud83d\ude0c #ghenerelindsi #consistency #dreams #goals #sundays #business #entrepreneur
#youngmind #goldenticket #craft #dreamers #smile #joy #content #socialmedia #power #growth
https:\/\/t.co\/elqtvpdssf
\ud83d\udd77 #dank #dannycharnley #comedy #tour #funny #lol #lmao #lmfao #hilarious #laugh
#laughing\u2026 https:\/\/t.co\/doiijkrezu
\ud83d\udca3 #begreat #onelife #thinkbigger #goals #10x #nevergiveup #thesecret #mindset
#quotes #life #10xlife #entrepreneur #hustle #successquotes #health #fitness #motivation
#instagood #instaphoto #work #happy #greatness #success #love #thinkbigsundaywithmarsha
#retweet #happysunday https:\/\/t.co\/omf10dcfrb
have a nice sunday &amp; enjoy spring\ud83d\udc90discover us\ud83d\udc47https:\/\/t.co\/
wjefnbdcis #msfriesch #msfrieschjewelry #jewelry #happy #outdoors #spring #landscape
#wedding #weddingjewelry #bridesmaids #spring #flowers #colorful #newcollection #followme
#sun #goodvibes #weekend #love #top #trend https:\/\/t.co\/e633kxpylf
omg look at da hoody \ud83d\ude31\ud83d\ude31 #holyghost #whitechicks #andineedyou
```

Figure 1: Examples of raw data

Some examples of raw data are shown in Figure 1; from the examples, we can see the raw data is not clean enough, there are escape characters(not emojis), unrelated hashtags, reserved words(like RT which represents retweet), and mentions(in the format: @username), besides, there are many non-English tweets; so the next step is clean data: firstly, by using the package preprocess in Python, we remove all mentions and reserved words; then we go through whole text to remove unrelated escape characters and hashtags, for related hashtags, we extract them as labels; finally we determine the language of tweet with the help of NLTK package(Bird et al., 2009) in Python. After cleaning data, we keep the cleaned pure text and their corresponding hashtags and store them into a data frame, part of the data frame is shown below.

hashtag	plain_text
#blessed	how did i get so lucky? 😌
#love	so proud of you ❤🤣
#lol	so proud of you ❤🤣
#love	so, ummm.... i so a kinda want you. no joke. lik...
#elat	so, ummm.... i so a kinda want you. no joke. lik...

Figure 2: Examples of cleaned data

We can see in examples, there are duplicate texts, that is because there are two targeted hashtags exist in the text, so we duplicate the texts, and assign them different hashtags.

Totally, we collect 70000 cleaned data from two sources, 60000 of them are from real-time stream, others are from history data. Some statistics of the data we collect are shown in Table 1, Table 2 and Table 3. From Table 1, we can find out our data is imbalanced; hashtags which represent Love and Joy are much more than hashtags represent Sadness or Anger; we think it is a character of Twitter, people tends to share the love and joy moments rather than angry or sad moments. Table 2 reflects the same pattern we find in the Table 1, people tends to use emojis that are positive like red heart or heart eyes. Compare to Table 2, we find that the use of emoticons is obviously less than emojis, except the emoticons ‘:\’, which is unexpectedly frequently used by the users of Twitter. Generally, compared to other the data used in related work, the data is not enough, especially in the case that we take Neural Networks into account, but considering the limited time and equipment, this is the largest dataset we can collect, also, the target of this project is to detect the influence of emoji and emoticons to the sentiment analysis, rather than creating the state-of-art model. So, in short, the data we obtained is comparatively small, but it should be enough for us to conduct the experiment.

Following the procedures of distant supervision, we obtain tweets with emotional hashtags as labels. To filter valid tweets, we construct a list of hashtags representing each of the 22 secondary emotions proposed by Parrott (2001) as shown in the following pictures:

Labels	Counts
Love	36239
Joy	18625
Fear	2127
Surprise	1739
Sadness	1459
Anger	282

Table 1: Top 6 frequent hashtags

Emoji	Counts
❤️	10884
😍	8831
🔥	7485
😂	5022
💋	4917
⏬	3489

Table 2: Top 6 frequent emojis

Emoticons	Counts
:\	45401
:)	905
;)	245
:-)	74
8)	64
:(	35

Table 3: Top 6 frequent hashtags

Primary emotion	Secondary emotion	Tertiary emotions
Love	Affection	Adoration, affection, love, fondness, liking, attraction, caring, tenderness, compassion, sentimentality
	Lust	Arousal, desire, lust, passion, infatuation
	Longing	Longing
Joy	Cheerfulness	Amusement, bliss, cheerfulness, gaiety, glee, jolliness, joviality, joy, delight, enjoyment, gladness, happiness, jubilation, elation, satisfaction, ecstasy, euphoria
	Zest	Enthusiasm, zeal, zest, excitement, thrill, exhilaration
	Contentment	Contentment, pleasure
	Pride	Pride, triumph
	Optimism	Eagerness, hope, optimism
	Enthrallment	Enthrallment, rapture
	Relief	Relief
Surprise	Surprise	Amazement, surprise, astonishment
Anger	Irritation	Aggravation, irritation, agitation, annoyance, grouchiness, grumpiness
	Exasperation	Exasperation, frustration
	Rage	Anger, rage, outrage, fury, wrath, hostility, ferocity, bitterness, hate, loathing, scorn, spite, vengefulness, dislike, resentment
	Disgust	Disgust, revulsion, contempt
	Envy	Envy, jealousy
	Torment	Torment
Sadness	Suffering	Agony, suffering, hurt, anguish
	Sadness	Depression, despair, hopelessness, gloom, glumness, sadness, unhappiness, grief, sorrow, woe, misery, melancholy
	Disappointment	Dismay, disappointment, displeasure
	Shame	Guilt, shame, regret, remorse
	Neglect	Alienation, isolation, neglect, loneliness, rejection, homesickness, defeat, dejection, insecurity, embarrassment, humiliation, insult
	Sympathy	Pity, sympathy
Fear	Horror	Alarm, shock, fear, fright, horror, terror, panic, hysteria, mortification
	Nervousness	Anxiety, nervousness, tenseness, uneasiness, apprehension, worry, distress, dread

Figure 3: Examples of cleaned data

4 Methods

There are two approaches for the classification of the tweets to the sentiments. One machine learning approach is supervised learning technique and the other approach is unsupervised learning technique. Unsupervised learning means that the input data contains only the data points and no target labels and mainly consist of clustering like K-means, hierarchical and dimensionality reduction techniques like PCA, SVD and others. The main reason for the good accuracies behind the model is correct selection of features for the supervised learning techniques for classification of emotions. We are going to use supervised machine learning approaches for the sentiment classification using emoji and emoticons.

We will have two datasets: training and test datasets. Number of machine learning techniques are used for the purpose of classification like Naive Bayes, maximum entropy and support vector machines. So, we

collect the training dataset and train our classifiers on the collected training dataset. Next, we need to do the feature selection.
For simple baseline classifiers, we trained Random Forest(Liaw,Wiener and news,2002), SGD(Bottou, 2010) and Xgboost(Chen and Guestrin,2016) which are traditional classifiers for the purpose of classification. Random Forest classifier use the technique of combining many classification trees. After growing each of the classification trees, each tree gives vote for each of the class. The Random Forest choses the class label having the maximum number of votes among all the classification trees. Xgboost is the most widely used algorithm for classification now a days. The main goal of the algorithm is for increasing the speed and performance of the classification algorithm. Xgboost implements the gradient boosting decision tree algorithm. Boosting is an ensemble technique in which we keep adding the models one after the other so as to rectify the errors made by the previous models. We sequentially keep adding the models until no further improvements can be made. One example of boosting is the AdaBoost algorithm. In Gradient boosting, we create new models that predict the residuals or errors of prior models. We add together all the models to make the final prediction for the classification purpose. It is uses a gradient descent algorithm to minimize the loss when we add the new models.

This approach supports both regression and classification predictive modeling problems.
We used the 300 dimensional word embedding combined with the same length 300 dimensional emoji and emoticon embedding as feature vector for the above baseline models. We used the strategy of one vs all for the purpose of classification.
Recurrent neural network(Medsker and Jain,2001) is a neural network used for sequence modeling. At each of the time step, input vector and hidden state vector are given to the RNN and it produces another hidden state. While learning long range dependencies, recurrent neural networks run into the problem of exploding and vanishing gradients. Therefore LSTM(Hochreiter and Schmidhuber, 1997) are used to deal with the problem of vanishing and exploding gradients. The main classifiers for the purpose of classification in our work were LSTM and BiLSTM. LSTM is a variant of recurrent neural network which deals with the problem of vanishing and exploding gradients. LSTMs solve the problem of RNNs by augmenting RNN with memory cell. BiLSTMs are bidirectional LSTMs in which we can feed the data once from beginning to the end and once from end to beginning. The equations for the LSTM are:

$$i_t = \sigma(W^i x_t + U^i h_{t-1} + b_i) \tag{1}$$

$$f_t = \sigma(W^f x_t + U^f h_{t-1} + b_f) \tag{2}$$

$$o_t = \sigma(W^o x_t + U^o h_{t-1} + b_o) \tag{3}$$

$$\tilde{c_t} = \tanh(W^c x_t + U^c h_{t-1} + c_o) \tag{4}$$

$$c_t = f_t o c_{t-1} + i_t o \tilde{c_t} \tag{5}$$

$$h_t = o_t o \tanh(c_t) \tag{6}$$

To feed the input, we use 300 dimensional representation of word vector and emoji and emoticon vector.

5 Experiment

5.1 t-SNE of embedding

Before training the model, we first analyzed the embeddings of emojis and emoticons; however, because the embeddings we use is 300-dimensions, it is impossible to project the embeddings to 2-dimensions to observe, so we apply the t-Distributed Stochastic Neighbor Embeddin(t-SNE)(Maaten and Hinton, 2008), which is a technique for dimensionality reduction, to the embeddings, in theory, similar objects are more probable to cluster and dissimilar objects are more probable to be away from each other after reduction.
We applied t-SNE to both emojis and emoticons embeddings trained from description and corpus and reduce the dimension from 300 to 2. The results are shown in Figure 4 and Figure 5:

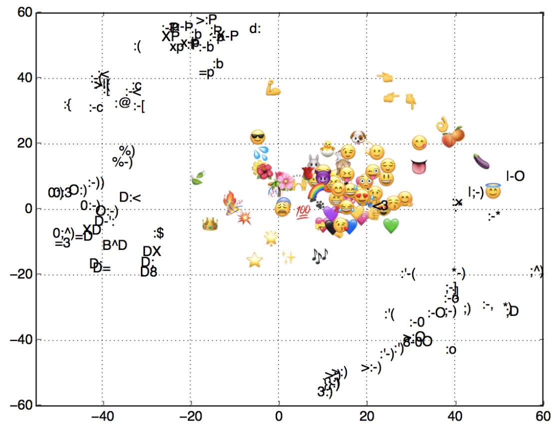


Figure 4: t-SNE of embeddings trained from description

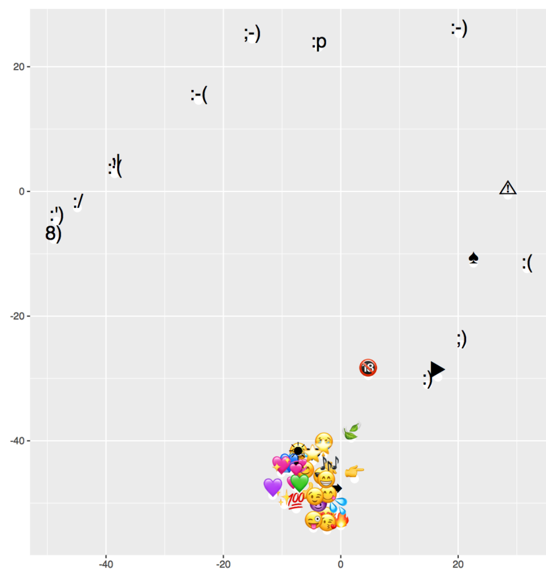


Figure 5: t-SNE of embeddings trained from corpus

We can observe that the embeddings trained from description is better than the embeddings trained from corpus. For embeddings trained from description, there are several obvious emotions' clusters like: joy, surprise, sadness, also similar emojis tend to get together, for example, trees, flowers and leaves are all in one cluster, peachs, oranges and eggplants are also close to each other. However, all smiley faces cluster together even though some of them have different meanings. For embeddings trained from corpus, compared to the embeddings trained from description, these embeddings do not make much sense, almost all emojis cluster together and all emotions randomly spread out.

So we conclude that the embeddings trained by description is better than the one trained by corpus. This accords with the result of another paper(Eisner et al., 2016).

5.2 Model Results

As explained earlier, we use 6 primal emotions defined by Parrot's theory as labels, and perform multiclass classification by LSTM, BiLSTM, Random Forest, SGD classifier and Xgboost.

For LSTM and BiLSTM, we tried several parameters and found out the one with 500 hidden unit, 16 batch size, dropout (Srivastava et al., 2014) with rate 0.5 is the best. 10 Epochs were used in training and Adam (Kingma and Ba, 2014) with 0.001 learning rate was deployed for optimization. Negative log likelihood function was loss function in the model. We implemented above settings in Pytorch.

We split data into 0.75 training (n=45957), 0.05 development (n=2419), 0.2 test set(n=12095).

Results with different embedding source Results with embeddings trained from different sources are listed in Table 4. The features in each model are 300 dimensional for each word, emoticon and emoji in the sentence. Description stands for embedding trained from standard description and Corpus means embedding trained from twitter data we collected. Acc means accuracy score. As we can observe in the table, embedding from standard description in general is dominating the one from corpus, which agrees with the conclusion in (Mikolov et al., 2013).

	Description		Corpus	
Model	Acc	F1	Acc	F1
BiLSTM	**0.73**	**0.73**	0.68	0.69
LSTM	**0.74**	**0.73**	0.70	0.70
Random Forest	**0.68**	**0.64**	0.65	0.63
SGD	**0.63**	0.60	0.60	0.60
Xgboost	0.64	0.58	0.64	**0.59**

Table 4: Comparison of different embedding sources

There are many reasons for such counter-intuition result, as people may think embedding from context learns the interactions. As we discussed in Section 2, one drawback of training from corpus is that it can only see small number of emoji and emoticon during training and cannot generalize to unseen tokens. What's worse, the surrounding words of emojis can be noisy sometimes, and many of such tweets come with videos or images that are actually the main topic. Thus learning emoji and emoji embedding from tweets is hard.

Results with different combinations of embedding: To test our hypothesis that emoji and emoticon are helpful in understanding context, we perform studies with different combinations of embeddings. The results are listed in Table 5. word+emoticon+emoji means we use embeddings for all of them in the models, while word+emoticon means we set the embedding for emojis as 0 (300 dimensions of course). It is the same to remove all emojis. Similar we have word+emoji and word only. Here we only present the results with embedding trained from standard description, because all scores for corpus are smaller, as we have discussed in last paragraph.

We can make following observations in Table 5:

- Including emoji and emoticon improves the model performance for all the models, which supports the usefulness of emotion tokens. For LSTM, adding emoji and emoticon improves F1 and accuracy for 7%.

- In general LSTM displays dominating performance in all settings, which again shows the power of deep learning.

- If we separate emoji and emoticon, emoji brings more improvement over emoticon. One way to explain this is emoticon, limited by its variety, can not express as rich emotion as emoji. However, this could also result from small number of emoticons we have in data.

Result for Individual Emotion: The prediction result for each emotion category is listed in table 6, together with the number of data in the test set. Since random forest outperforms other machine learning models in last section, for display simplicity we only compare random forest with LSTM. We can clearly see the labels are imbalanced (as we already showed in Section 3). In general LSTM performs better when there are more data, and the prediction could be noisy when data is not sufficient. This observation also lies in one drawback of deep learning model, which is requirement of large data.

Table 6 also shows the partial reason of error. In general models perform well for Love and Joy classes, and mostly the error comes from minor classes. Therefore it lead to one further work, which is to gather more data in minor classes.

	LSTM		Random Forest		
Category	Acc	F1	Acc	F1	count
Love	**0.711**	**0.793**	0.689	0.762	7428
Joy	**0.722**	**0.545**	0.716	0.504	3725
Sadness	0.971	**0.475**	**0.976**	0.169	292
Anger	0.948	**0.261**	**0.996**	0.01	56
Surprise	**0.974**	0.292	**0.974**	**0.323**	348
Fear	**0.995**	0.125	0.971	**0.384**	426

Table 6: Comparison of different emotion types

5.3 Semantic Consistency of Embedding

Mikolov et al. (2013) found out the word vectors trained from CBOW or skip-gram preserves certain consistency, in the sense the result of add/subtract operations on word vectors is meaningful. The famous example is king - man + woman $\approx$ queen. This could serve as good verification of validity of embeddings.

Similarly we would like to test on our emoji and emoticon embedding. We only experiment on embedding from standard description because its good model performance makes this experiment more promising. The result is shown in Figure 6. Magically the emoji embedding preserves the consistency for word vectors. We again have king - man + woman $\approx$ queen. It also displays certain generalization power as hamburger - bread $\approx$meat This not only supports the correctness of embedding training process, but also implies that the emoji embedding is compatible with word embedding. However, the results for emoticon are not very neat, partly because there are limited numbers of emoticon types. Nonetheless, we still have meaningful result such as happy face + sad face - happy face = sad face.

6 Future Work

The data collected from the micro blogging websites are very short in text size and very informal. Therefore, as future work, we plan to get more data and train our models used in this work on more data. We plan to obtain at least 2 million data with which we

	BiLSTM		LSTM		Random Forest		SGD		Xgboost	
combination	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
word+emoji+emoticon	0.73	**0.73**	**0.74**	**0.73**	0.68	0.64	0.63	0.6	0.64	0.58
word+emoticon	0.68	0.69	**0.69**	**0.7**	0.64	0.61	0.58	0.57	0.63	0.56
word+emoji	0.7	0.7	**0.72**	**0.72**	0.66	0.64	0.61	0.53	0.66	0.6
word	0.65	**0.66**	**0.68**	**0.66**	0.65	0.62	0.56	0.5	0.63	0.56

Table 5: Comparison of models in different embedding settings

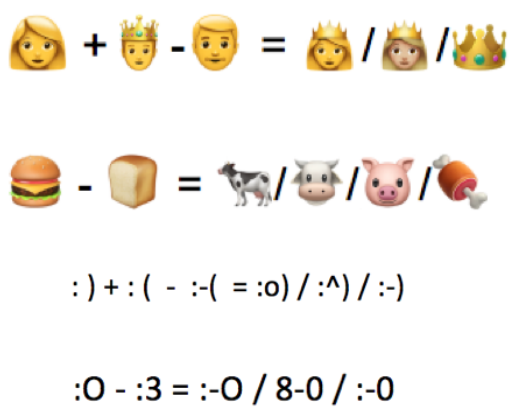


Figure 6: Semantic consistency for emoji and emoticon embedding

are hopeful to get better results. We can also aim to work with more sophisticated algorithms and architecture in which we can connect words and emojis using lexical meaning.
Some possible models include adding attention layer after multiple layers of LSTM, which essentially will stress helpful emojis or words. Another idea for future work involves incorporating sociolinguistic aspects of the emojis. One more addition which we can do in our model in future is using secondary and tertiary emotions instead of just using primary emotions for the model. Another very potential idea for future research work is to investigate the other sources of emotion indicators like emoji and emoticons and able to extract more information. We can also try to work with the more cleaner data in the future so that we are able to get better accuracies for the classification. We can in future also try to combine machine learning method with opinion lexicon method so as to improve the accuracy of the classification.

7 Conclusion

In this project we perform emotion prediction on twitter data with emoji and emoticon embedding. We prove our hypothesis that emoji and emoticon are helpful in order to better understand social media conversations. We train emoji and emoticon embedding from different sources and show that embedding from standard description brings more improvement to models compared with embedding from twitter data, and we discuss possible reason for such result. We also show that the deep learning models are better than machine learning models given sufficient data.

8 Individual Contribution

8.1 Writeup

- Haozheng Ni (hn2318): Write introduction, related work, half of experiment and conclusion
- Chuqi Yang (cy2478): Data and t-SNE
- Somya Singhal (ss5348): Write Methods and future work.

8.2 Project

- Haozheng Ni (hn2318): Implement models and run on GPU
- Chuqi Yang (cy2478): Twitter data collection, data clean, apply t-SNE to word embeddings.
- Somya Singhal (ss5348): Data Collection, selecting emotion model and making list of hashtags corresponding to each emotion where we use hashtags as classification labels, implementation of code for training emoji and emoticon embedding on skip gram neural network and output the emoji and emoticon vectors to be used further down in the pipeline alongside word vectors.

References

Muhammad Abdul-Mageed and Lyle Ungar. 2017. Emonet: Fine-grained emotion detection with gated recurrent neural networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 718–728.

Francesco Barbieri, German Kruszewski, Francesco Ronzano, and Horacio Saggion. 2016a. How cosmopolitan are emojis?: Exploring emojis usage and meaning over different languages with distributional semantics. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 531–535. ACM.

Francesco Barbieri, Francesco Ronzano, and Horacio Saggion. 2016b. What does this emoji mean? a vector space skip-gram model for twitter emojis. In *LREC*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Joe Cruse. 2015. Emoji usage in tv conversation.

Oxford Dictionary. 2015. word of the year 2015.

Ben Eisner, Tim Rocktäschel, Isabelle Augenstein, Matko Bošnjak, and Sebastian Riedel. 2016. emoji2vec: Learning emoji representations from their description. *arXiv preprint arXiv:1609.08359*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

W Gerrod Parrott. 2001. *Emotions in social psychology: Essential readings*. Psychology Press.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Hao Wang and Jorge A Castanon. 2015. Sentiment expression via emoticons on social media. In *Big Data (Big Data), 2015 IEEE International Conference on*, pages 2404–2408. IEEE.