Sentiment Analysis of Tweets

Adarsh Ranjan

Abstract

These days people spend many hours on social networks and share their opinion, therefore social networks have become a great source to gather information about their opinion and sentiment towards different topics. Twitter is a such a social network website. This project deals with analyzing sentiments behind the tweets, whether they are about a person, product, movie, organization or people's everyday lives.

1. Introduction

Microblogging websites have evolved to become a source of varied kind of information. This is due to nature of microblogs on which people post real time messages about their opinions on a variety of topics, discuss current issues, complain, and express positive sentiment for products they use in daily life. In fact, the companies which manufactures such products have started to poll these microblogs to get a sense of general sentiment for their product. Many times these companies study user reactions and reply to users on these microblogs. One challenge is to build technology to detect and summarize an overall sentiment.

In this paper, we look at one such popular microblog called Twitter and build a model for classifying tweets into positive, negative and neutral sentiment.

The twitter data represents a true sample of actual tweets in terms of language use and content. In this paper we also introduce two resources which are available: 1) a hand annotated dictionary for emoticons that maps emoticons to their polarity and 2) a dictionary collected from web with frequently used contractions in English sentences for their expansions.

2. Data Characteristics

Twitter is a social networking and microblogging service that allows users to post real time messages, called tweets. Tweets are short messages, restricted to 140 characters in length. Due to the nature of this microblogging service, people use acronyms, make spelling mistakes, use emoticons and other characters that express special meanings.

Following is a brief terminology associated with tweets.

- Emoticons: These are facial expressions pictorially represented using punctuation and letters; they express the user's mood.
- Target: Users of Twitter use the "@" symbol to refer to other users on the microblog. Referring to other users in this manner automatically alerts them.
- Hashtags: Users usually use hashtags to mark topics. This is primarily done to increase the visibility of their tweets.

Our dataset consisted of 21,465 tweets and contained tweet id, polarity and the tweets. The training data was imbalanced as we had 9064 positive tweets, 3387

negative tweets and 9014 neutral tweets. Since we do not require tweet id, we remove it from our training data. We have 42.27% positive, 41.99% neutral and 15.78% negative tweets.

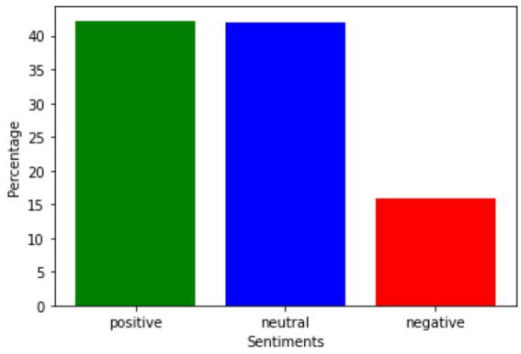

Figure 1: Percentage of each type of tweets

3. Resources and preprocessing of Data

In this paper we introduce two new resources for pre-processing the tweets: 1) an emoticon dictionary and 2) a contraction dictionary. We prepare the emoticon dictionary by labeling some emoticons listed on Wikipedia with their emotional state. For example, ":)" is labeled as positive whereas ":=(" is labeled as negative. We assign each emoticon a label from the following set of labels: extremely positive, extremely negative, positive, negative, and neutral. Some frequent contractions like *I'm* was replaced by *I am* with the help of contraction dictionary.

As a pre-processing step, tweets must be cleaned. First, the unicodes were replaced by the characters they represent. Most frequent unicodes were "\\u2019" and "\\u002c" which were replaced by " , " and " ' " respectively. Taking advantage of regular expressions, all the links, urls, targets and hashtags were removed. Then the contractions were expanded and emoticons were replaced by the emotions they denote with the help of dictionary. We also treated elongations (that is, the repetition of a character) by removing the repetition of a character after its second occurrence (for example, happpyyyy would be translated to happyy). Finally we removed all stopwords present in tweets. The output labels were label encoded and then converted to one-hot.

Once the tweets were preprocessed, they were tokenized using the keras preprocessing text tokenizer and were converted to sequences. These sequences were padded to the length of the longest tweet. We used the pretrained Word2Vec word embeddings to learn the contextual relations among words in training data.

4. Background

In this section we will on the basic model constructs of Recurrent neural networks and Convolutional neural networks.

4.1. RNN for Sentiment Analysis

Recurrent neural networks (RNN) are one of the most popular architectures used in NLP problems because their recurrent structure is very suitable to process the variable-length text

A simple strategy for modeling sequence is to map the input sequence to a fixed-sized vector using one RNN, and then to feed the vector to a softmax layer for classification. But, a problem with RNNs is that during training, components of the gradient vector can grow or decay exponentially over long sequences. This problem with exploding or vanishing gradients makes it difficult for the RNN model to learn long distance correlations in a sequence.

Long short-term memory network (LSTM) and Gated Recurrent Unit (GRU) can specifically address this issue of learning long-term dependencies. In our case, it is possible that a Bidirectional LSTM or GRU could allow us to capture

changing sentiment in a tweet. For example, *the movie was terribly exciting* expresses a positive sentiment but the word *terribly* denotes a negative polarity. Here the word *exciting* changes the meaning of word *terribly.*

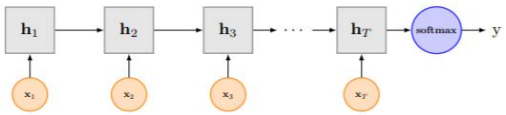

Figure 2: RNN for Sentence Classification.

4.2. CNN for Sentiment Analysis

Intially designed for image recognition, CNNS have become an incredibly versatile model used for a wide array of tasks. CNNS have the ability of recognizing local features inside a multi-dimensional field. The intution behind using CNNS on text relies on the fact that text is structured and organized. As such, we can aspect a CNN model to dicover and learn patterns that would otherwise lost in a feed forward network.

5. Methods

This section will be proposed of the models that were used. Since its a multilabel classification problem we use the softmax activation function in the final layer of every model. We used the categorical cross entropy loss function. We separated some data from the data set to form the validation set and rest the training set. Hyperparameters were tuned on the validation set.

5.1. LSTM Model

We used Bidirectional LSTM for our model. The model combination consists of Bi-LSTM layer which will receive an Embedding layer as its input. This Bi-LSTM layer was followed by another Bi-LSTM layer. We added a dropout of 0.1. The output was then passed to a fully connected layer with softmax activation function. Optimizer used was RMSprop with a learning rate of 0.001. We trained for 20 epochs with a batch size of 64. We used learning stop so that the model doesn't overfit to the training data.

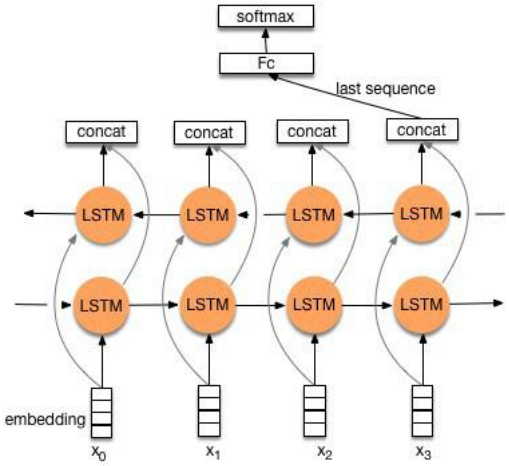

Figure 3: Bi-LSTM for Sentence Classification.

5.2. GRU Model

Here, we used Bidirectional GRU for our model. The initial embedding layer were fed to the bidirectional GRU and a droput of 0.1 was added. We passed the output to a fully connected layer with softmax activation function. RMSprop was te oprimizer used with a learning rate of 0.001. We trained for 20 epochs with a batch size of 64. We used learning stop so that the model doesn't overfit to the training data.

5.3. CNN Model

The CNN model initially consisted of an embedding layer which was passed to a Conv1d layer with 20 filters and kernel size equals to 3 with a relu activation function. Its output was then pooled and fed to a GlobalMaxPooling1D layer. A dropout of 0.2 was added and its output was passed to

a fully connected layer with softmax activation function. We used the optimizer to be RMSprop with a learning rate of 0.001. We trained for 20 epochs with a batch size of 64. We used learning stop so that the model doesn't overfit to the training data.

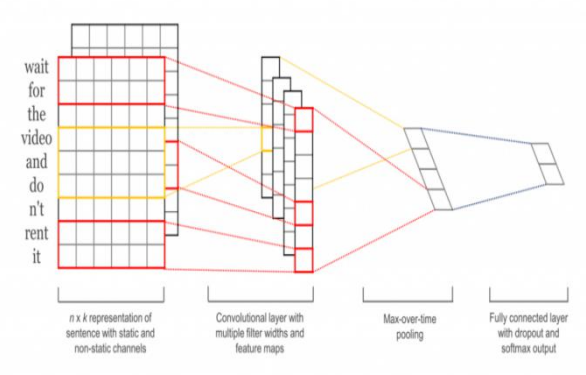

Figure 4: Kim, Y. (2014). CNNs for Sentence Classification

5.4. LSTM-CNN Model

In LSTM-CNN model we added a spatial dropout of 0.4 to the initial embedding layer which was followed by a BI-LSTM layer. The output of Bi-LSTM layer was fed to a Conv1D layer with 64 filters and a kernel size of 3 with a tanh activation function. The output of Conv1D layer was then pooled and fed to a GlobalMaxPooling1D layer. We added a dropout of 0.6. The output of dropout layer was passed to a dense layer with relu activation function. Later, this output was passed to a fully connected layer with softmax activation function.

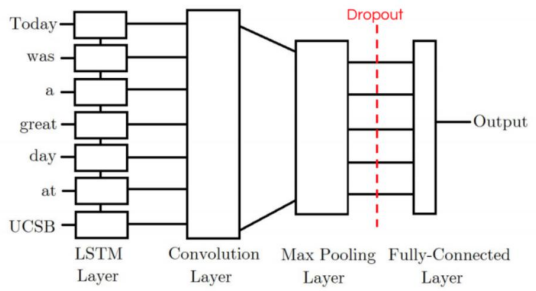

Figure 5: LSTM-CNN for Sentence Classification

6. Result

In this section we will compare the different models. We trained the model and measured there average F1 score which were found to be:

Model	Average F1 Score
LSTM	0.6813
GRU	0.6676
CNN	0.6635
LSTM-CNN	0.7058

We find that the LSTM-CNN model works the best for the given dataset with a score of 0.7058. The LSTM model we used has a score of 0.6813. GRU model has a F1 score of 0.6676 and for the CNN model we have a F1 score of 0.6635.

7. Conclusion

In this paper, we discussed the importance of social network analysis and its applications in different areas. We focused on Twitter and implemented four models that aim to achieve better performance on sentiment analysis of tweets. Our LSTM-CNN model gives a better performance than the other models. We tentatively conclude that sentiment analysis for Twitter data is not that different from sentiment analysis for other genres.

8. References

Y. Kim, "Convolutional Neural Networks for Sentence Classification," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), pp. 1746–1751, 2014.*

P. M. Sosa and S. Sadigh, “Twitter Sentiment Analysis using Combined LSTM-CNN models,” *Academia.edu*, 2016

Isabel Segura-Bedmar, Antonio Quiros and Paloma Martınez, “Exploring Convolutional Neural Networks for Sentiment Analysis of Spanish tweets,” *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 1014–1022, Valencia,* 2017