Proposal :Tweet Sentiment Extraction

Springboard Capstone 2 Project Proposal;
Pavan Poosarla

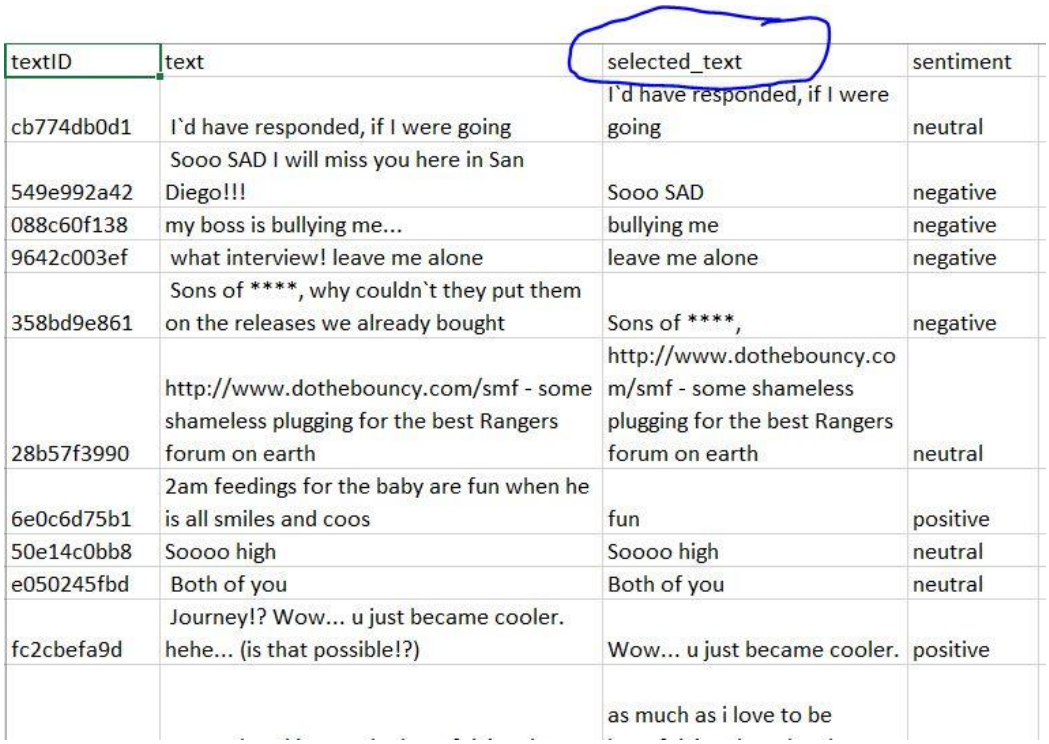
Link : https://www.kaggle.com/c/tweet-sentiment-extraction/data

Introduction

For the second capstone project, the problem we attempt to solve is an extension of the classical sentiment analysis problem for twitter data. In this problem, we would like to pick out the exact phrase of the tweet that informs the sentiment, given the sentiment. The inspiration and data for this problem is obtained from the above Kaggle Competition

Problem Statement

Given a tweet and the sentiment associated with it, determine the best words in the tweet that capture the sentiment
For example, in the snip of the dataset below, predict the column that is annotated with blue circle



textID	text	selected_text	sentiment
cb774db0d1	I`d have responded, if I were going	I`d have responded, if I were going	neutral
549e992a42	Sooo SAD I will miss you here in San Diego!!!	Sooo SAD	negative
088c60f138	my boss is bullying me...	bullying me	negative
9642c003ef	what interview! leave me alone	leave me alone	negative
358bd9e861	Sons of ****, why couldn`t they put them on the releases we already bought	Sons of ****,	negative
28b57f3990	http://www.dotheboucy.com/smf - some shameless plugging for the best Rangers forum on earth	http://www.dotheboucy.com/smf - some shameless plugging for the best Rangers forum on earth	neutral
6e0c6d75b1	2am feedings for the baby are fun when he is all smiles and coos	fun	positive
50e14c0bb8	Soooo high	Soooo high	neutral
e050245fbd	Both of you	Both of you	neutral
fc2cbefa9d	Journey!? Wow... u just became cooler. hehe... (is that possible!?)	Wow... u just became cooler.	positive
		as much as i love to be	

Motivation

This is a further value addition from sentiment analysis when the company/ government or person interested in sentiment analysis wants to have ‘human-readable’ descriptions associated with the sentiment in question. This model will also have value summarizing larger pieces of text while preserving the sentiment associated with it. For example, this can be used to process customer feedback text to summarize the large body of feedback text while capturing the reason for given rating

Data Sources

Data sources used will be in the link below. In addition to this, it may be required to supplement the dataset with other freely available data from twitter API
LInk : https://www.kaggle.com/c/tweet-sentiment-extraction/data

Initial Approach

The initial approach for attacking the problem is as follows
1. Data Wrangling : Use regex on tweets to capture URL’s , substitutions for expletives(***, F***, etc.). We would also need to correct typos (potentially separate intentional misspelling from unintentional), etc. However, we need to do this without losing the original text

2. Exploratory Data Analysis : We need to identify which words appear strongly in positive and negative connotations. We will need to treat the words that can be used in both contexts differently

3. Analysis : Plan to use a two part approach :

a. classify all words into two categories - 1. Which have strong sentiment and which do not : One potential way to do this is to use Naive Bayes on bag-og-words vectorizer on the dictionary
b. Next, we need to use word2vec or some other formulation that accounts for context of the words to predict the relative importance of different vectors in the sentence given the sentiment.

Data Wrangling

The raw dataset consists of four columns, one is a unique textID for each data point. One of the column is the full text of the tweet. The remaining two columns are for sentiment associated with the tweet and the selected text from the full tweet that expresses the said sentiment.
The first step in data cleaning is to look for and deal with missing values in dataset. We see that one row has the tweet and selected text missing. This row is dropped, which leaves us with 27480 lines of training data
The next step is to create a cleaned text dataset with the following cleaning steps

1. Remove leading and lagging white space
2. Remove URL's in the text and extract them into a different column consisting of list of url's
3. Convert all text into lower case

In addition to the cleaned_text column consisting of the tweet preprocessed with above steps, we create the following additional columns in the training dataset

1. *word_count* : Contains the number of words contained in the text
2. *sen_count* : Contains the number of sentences in the tweet
3. *url_list* : List of urls in tweet. List is [ ] for tweets with no url's
4. *char_count* : Number of characters in the tweet
5. *selected_char_count* : Number of characters in *selected_text*
6. *jaccard* : Lists the jaccard score of the original text and the selected text

Figure 1 Distribution of the number of tweets by sentiment in the training and test data

Exploratory Data Analysis

Once the dataset is cleaned, we look at the trends in the data. First, lets look at the distribution of the number of tweets with each sentiment in training and the text data. As can be seen from the figure, we do notice that the classes are imbalanced. The number of tweets with neutral sentiment seem to be higher for both test and train dataset

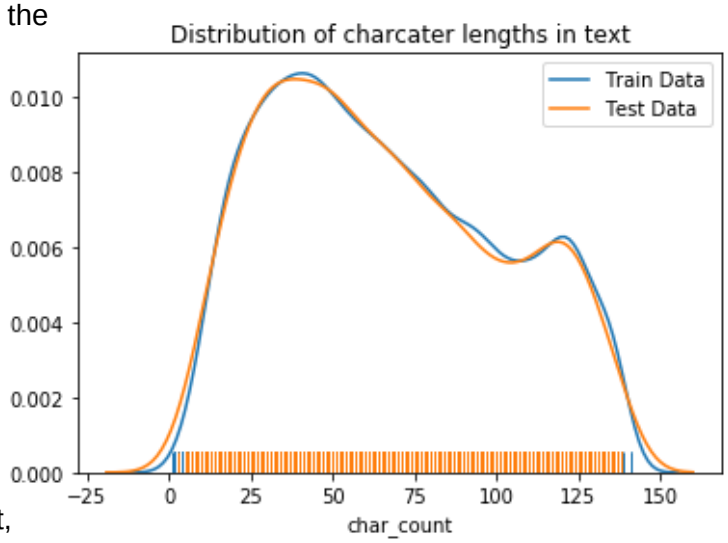


Figure 2 Distribution of character counts in training data and the test data

Similarly, we see that the distribution of character lengths in tweets is also similar between the test and training set, as shown in the figure. Furthermore, the character count distribution does not seem to be impacted by the sentiment of the tweet. However, the distribution of character counts in selected text shows a dependence on the sentiment. It appears that for neutral tweets, most of the actual tweet is selected while for the positive and the negative tweets, only a small part of the tweet is selected for most cases.

This is also shown in another way in the scatterplot of the character counts of selected text plotted against the main tweet. As we expect, we see that most of the data points for positive and negative tweets lie toward the right-bottom corner of the plot. This shows that in most cases, the selected point is smaller than the main tweet for positive and negative sentiments.

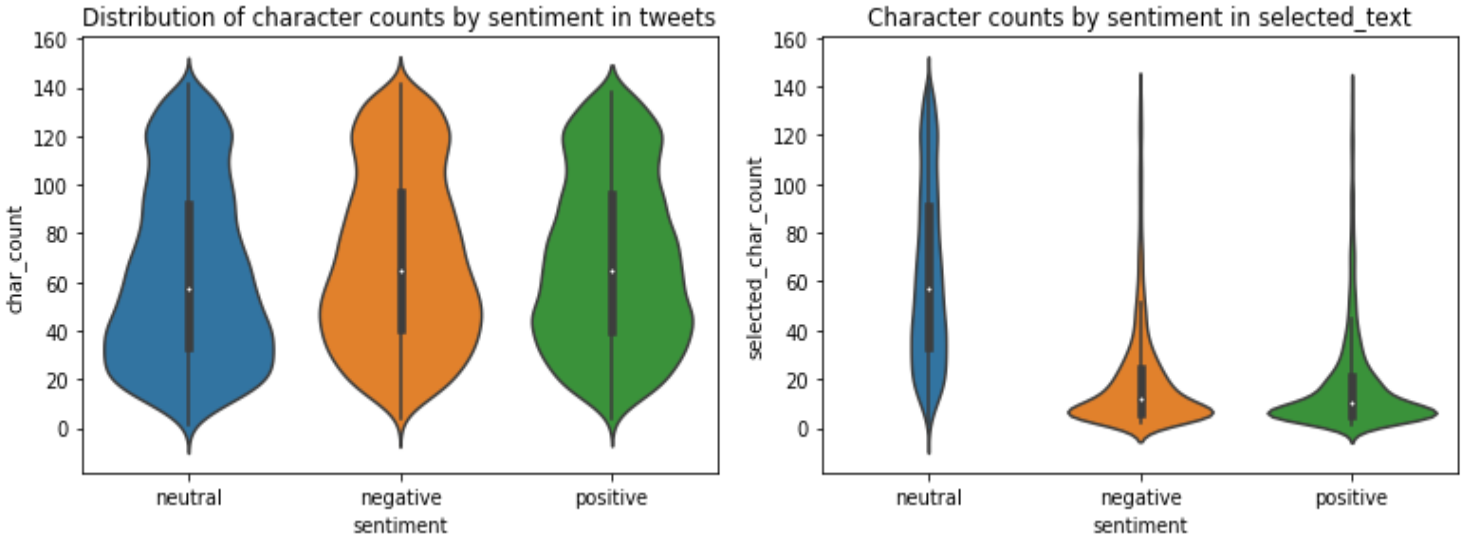


Figure 3 Distribution of character lengths by sentiment in the full tweet and in the selected text

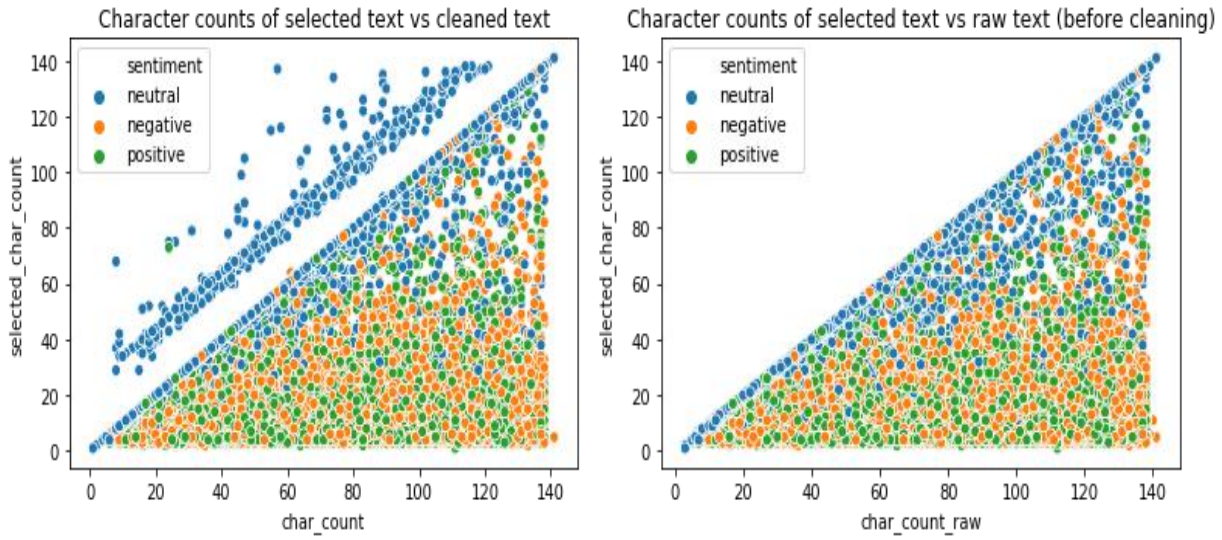


Figure 4 Scatter plot of character counts in selected text vs main text, with character counts in tweet extarcted after removing URL's and before

However, we see that for neutral sentiment, we see lot of datapoints on the 45-degree curve, suggesting that in those cases, the entire text of the tweet is selected to capture the sentiment. In addition, we see a few datapoints where the selected data has more characters than the main tweet. These correspond to the tweets with URL's embedded in the text body. For these tweets, the selected text seems to have included the URL"s. Because we removed the URL's from the main text but not from the selected text, we see this artifact of selected tweet having more characters than the main tweet. This is no longer the case, as we can see, when the charcater count of main tweets is extracted without removing the url.

Finally, we expect the most common words associated with each of the sentiment to be different. To visualize this, we plot word clouds for each of the sentiments in the talks. We find that the words in tweets associated with positive sentiment have 'happy' words more often and tweets with negative tweets have more occurrences of words with negative connotations. For example, the most common words associated with positive sentiment are ***love, thank, good, great, happy, fun, haha, hope, etc***. On the other hand, common words in the negative tweets are ***word, miss, sad, sorry, hate, etc***. Word *work* also commonly appears in neutral tweets. Other common words in neutral tweets are ***lol, time, going, etc.***

The word clouds for the sentiments are shown in the following page

WordCloud, Sentiment : Neutral

WordCloud, Sentiment : Positive

WordCloud, Sentiment : Negative