SOCIAL MEDIA SENTIMENTAL ANALYSIS: TWITTER USING PYTHON MACHINE LANGUAGE

AMITRAJ M amithrajmahalings872000@gmail.com

JOYJIT BHANDARI joyjitbhandari91210@gmail.com

BHAVANI V bhavanisvreddy3242@gmail.com

(Department of Computer Science & Engineering,

Karavali Institute of Technology

Mangalore-575001.)

Abstract

Twitter comments and posts express opinion of news entities (people, places, things) while reporting on recent events. We present a system that assigns scores indicating positive or negative opinion to each distinct entity in the text corpus. Our system consists of a sentiment identification phase, which associates expressed opinions with each relevant entity, and a sentiment aggregation and scoring phase, which scores each entity relative to others in the same class. Finally, we evaluate the significance of our scoring techniques over large corpus of comments and posts.

1.Introduction

Natural Language Processing (NLP) is a hotbed of research in data science these days and one of the most common applications of NLP is sentiment analysis. From opinion polls to creating entire marketing strategies, this domain has completely reshaped the way businesses work, which is why this is an area every data scientist must be familiar with.

Thousands of text documents can be processed for sentiment (and other features including named entities, topics, themes, etc.) in seconds, compared to the hours it would take a team of people to manually complete the same task.

We will do so by following a sequence of steps needed to solve a general sentiment analysis problem. We will start with pre-processing and cleaning of the raw text of the tweets. Then we will explore the cleaned text and try to get some intuition about the context of the tweets. After that, we will extract numerical features from the data and finally use these feature sets to train models and identify the sentiments of the tweets.

2.Problem Statement Analysis

Let's go through the problem statement once as it is very crucial to understand the objective before working on the dataset. The problem statement is as follows: The objective of this task is to detect hate speech in tweets. For the sake of simplicity, we say a tweet contains hate speech if it has a racist or sexist sentiment associated with it. So, the task is to classify racist or sexist tweets from other tweets.

Formally, given a training sample of tweets and labels, where labels denotes the tweet is racist/sexist and, our objective is to predict the labels on the given test dataset.

3.Processing And Cleaning

You are searching for a document in this office space. In which scenario are you more likely to find the document easily? Of course, in the less cluttered one because each item is kept in its proper place. The data cleaning exercise is quite similar. If the data is arranged in a structured format then it becomes easier to find the right information.

The pre-processing of the text data is an essential step as it makes the raw text ready for mining, i.e., it becomes easier to extract information from the text and apply machine learning algorithms to it. If we skip this step then there is a higher chance that you are working with noisy and inconsistent data. The objective of this step is to clean noise those are less relevant to find the sentiment of tweets such as punctuation, special characters, numbers, and terms which don't carry much weightage in context to the text.

In one of the later stages, we will be extracting numeric features from our Twitter text data. This feature space is created using all the unique words present in the entire data. So, if we pre-process our data well, then we would be able to get a better quality feature space.

4. Story Generation and Visualization from Tweets

In this section,we will explore the cleaned tweets text. Exploring and visualizing data, no matter whether its text or any other data, is an essential step in gaining insights. Do not limit yourself to only these methods told in this tutorial, feel free to explore the data as much as possible.

Before we begin exploration, we must think and ask questions related to the data in hand. A few probable questions are as follows:

- What are the most common words in the entire dataset?
- What are the most common words in the dataset for negative and positive tweets, respectively?

- how many hashtags are there in a tweet?
- Which trends are associated with my dataset?
- Which trends are associated with either of the sentiments? Are they compatible with the sentiments?

5. Interpretation And Scoring of Sentiment Data

We use our sentiment lexicons to mark up all sentiment words and associated entities in our corpus. We reverse the polarity of a sentiment word whenever it is preceded by a negation. We increase/decrease the polarity strength when a word is preceded by a modifier. Thus not good = -1; good = +1; very good = +2. Our sentiment analyser ignores articles which are detected as being a duplicate of another . This prevents news syndicate articles from having a larger impact on the sentiment than other articles. Since our system processes vast quantities of text on a daily basis, speed considerations prevent us from doing careful parsing. Instead, we use co-occurrence of an entity and a sentiment word in the same sentence to mean that the sentiment is associated with that entity. This is not always accurate, particularly in complex sentences. Still the volume of text we process enables us to generate accurate sentiment scores. We take several steps to aggregate entity references under different names. By employing techniques for pronoun resolution, we can identify more entity/sentiment co-occurrences than occur in the original news text. Further, Lydia's system for identifying co-reference sets associates alternate references such as George W. Bush and George Bush under the single synonym set header George W. Bush. This consolidates sentiment pertaining to a single entity.

6. Polarity Scores

We use the raw sentiment scores to track two trends over time:

• Polarity: Is the sentiment associated with the entity positive or negative?

• Subjectivity: How much sentiment (of any polarity) does the entity garner? Subjectivity indicates proportion of sentiment to frequency of occurrence, while polarity indicates percentage of positive sentiment references among total sentiment references. We focus first on polarity. We evaluate world polarity using sentiment data for all entities for the entire time period: world polarity = positive sentiment references total sentiment references We evaluate entity polarity using sentiment data for that day (Dayi) only: entity polarity = positive sentiment references total sentiment references shows the correlation coefficient between the various sentiment indices. In general, pairs of indices are positively correlated but not very strongly. This is good, as it shows each subindex measures different things. The General index is the union of all the indices and hence is positively correlated with each individual index.

7.Sentiment Lexicon Generation

Sentiment analysis depends on our ability to identify the sentimental terms in a corpus and their orientation. We defined separate lexicons for each of seven sentiment dimensions (general, health, crime, sports, business, politics, media). We selected these dimensions based on our identification of distinct news spheres with distinct standards of opinion and sentiment. Enlarging the number of sentiment lexicons permits greater focus in analyzing topic-specific phenomena, but potentially at a substantial cost in human curation. To avoid this, we developed an algorithm for expanding small dimension sets of seed sentiment words into full lexicons. 3.1 Lexicon expansion through path analysis Previous systems detailed in Section 2 have expanded seed lists into lexicons by recursively querying for synonyms using the computer dictionary WordNet . The pitfall of such methods that that synonym set coherence weakens with distance. shows four separate ways to get from good to bad using chains of WordNet synonyms. To counteract such problems, our sentiment word generation algorithm expands a set of seed words using synonym and antonym queries as follows:

• We associate a polarity (positive or negative) to each word and query both the synonyms and antonyms, akin to [15, 16] Synonyms inherit the polarity from the parent, whereas antonyms get the opposite polarity.

• The significance of a path decreases as a function of its length or depth from a seed word, akin to . The significance of a word W at depth d decreases exponentially as score(W) = 1/cd for some constant c > 1. The final score of each word is the summation of the scores received over all paths.

.

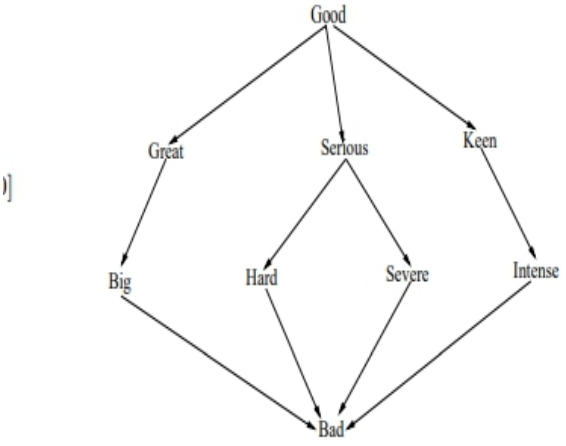

Fig. 1: *Four ways to get from bad to good in three hops*

8.Future Scopes

Sentiment Analysis has been more than just a social analytic tool. It's been an interesting field of study. But it is a field that is still being studied, although not at great lengths due to the intricacy of this analysis. That is this field has functions that are too complicated for machines to understand. The ability to understand sarcasm, hyperbole, positive feelings, or negative

feelings has been difficult, for machines that lack feelings. Algorithms have not been able to predict with more than 99.90% accuracy the feelings portrayed by people. Yet with so many limitations this is one field which is growing at great pace within many industries. Companies want to accommodate the sentiment analysis tools into areas of customer feedback, marketing, CRM, and ecommerce.

9.Conclusion

There are many interesting directions that can be explored. We are interested in how sentiment can vary by demographic group, news source or geographic location. By expanding our spatial analysis of news entities [1] to sentiment maps, we can identify geographical regions of favorable or adverse opinions for given entities. We are also studying in analyzing the degree to which our sentiment indices predict future changes in popularity or market behaviour.
In this article, we learned how to approach a sentiment analysis problem. We started with pre-processing and exploration of data. Then we extracted features from the cleaned text using Bag-of-Words and TF-IDF. Finally, we were able to build a couple of models using both the feature sets to classify the tweets.

10.Reference

- https://github.com/joyjitbhandari/sentiment_analysis_project.git
- https://youtu.be/1gQ6uG5Ujiw
- https://youtu.be/lMQzEk5vht4
- https://github.com/Komal7209/BackUp-Twitter-Sentiment-Analysis-
- https://github.com/sharmaroshan/Twitter-Sentiment-Analysis/find/master