**Identifying Sentiment On Social Media Platform using Stanford NLP Techniques**

**Abstract:**

Many users share their comments on Twitter, making it a valuable platform for tracking and analyzing public feelings. Such tracking and analysis will provide important information for higher efficacy processes across multiple domains During this work, we tend to move one step toward the interpretation of sensory variations of tweets. We tend to discover that the topic has varied sentiments within a unit period of time. The variance of confidence relates to the real reason behind the variations. To support this observation, we are likely to propose a model to refine topics and highlight tweets. We tend to select the most important tweets for the topic, and develop another mechanical model, called the hybrid model to rank them by reference aspect wise using heuristic rules within. To increase readability of the reason for digging, we chose the most representative tweets for the foreground topic, and developed another model called hybrid model using triple relation extraction to collect and to rank the topics with respect to their "popularity" during the transition. The results show that our method can find the foreground and the logical order effectively. The proposed format can be used for other tasks, such as finding the difference between different twitter datasets.

**Introduction**

The explosive growth of user generated messages, Twitter has become a social site where millions of users can exchange their opinion. Sentiment analysis on Twitter data has provided an economical and effective way to expose public opinion timely, which is critical for decision making in various domains. For instance, a company can study the public sentiment in tweets to obtain users’ feedback towards its products; while a politician can adjust his/her position with respect to the sentiment change of the public.

There have been a large number of research studies and industrial applications in the area of public sentiment tracking and modeling. Previous research like O’Connor et al. focused on tracking public sentiment on Twitter and studying its correlation with consumer confidence and presidential job approval polls. Similar studies have been done for investigating the reflection of public sentiment on stock markets and oil price indices. They reported that events in real life indeed have a significant and immediate effect on the public sentiment on Twitter. However, none of these studies performed further analysis to mine useful insights behind significant sentiment variation, called public sentiment variation. One valuable analysis is to find possible reasons behind sentiment variation, which can provide important decision-making information.

For example, if negative sentiment towards Barack Obama increases significantly, the White House Administration Office may be eager to know why people have changed their opinion and then react accordingly to reverse this trend. Another example is, if public sentiment changes greatly on some products, the related companies may want to know why their products receive such feedback. It is generally difficult to find the exact causes of sentiment variations since they may involve complicated internal and external factors. We observed that the emerging topics discussed in the variation period could be highly related to the genuine reasons behind the variations. When people express their opinions, they often mention reasons (e.g., some specific events or topics) that support their current view.

In this work, we consider these emerging events/topics as possible reasons. Mining emerging events/topics is challenging:

(1) The tweets collection in the variation period could be very noisy, covering irrelevant “background” topics which had been discussed for a long time and did not contribute to the changes of the public’s opinion. How to filter out such background topics is an issue we need to solve. Text clustering and summarization techniques are not appropriate for this task since they will discover all topics in a text collection.

(2) The events and topics related to opinion variation are hard to represent.

**Our Objective:**

* We aim to implement an algorithm for automatic classification of text into positive, negative or neutral sets.
* To extract the meaning of an input text or tweet using natural language processing.
* To determine the attitude of the mass into various objective sets towards the subject of interest.
* To improve the accuracy of the analysis using our algorithm.

Keywords produced by topic modeling can describe the underlying events to some extent. But they are not as intuitive as natural language sentences.

Reasons could be complicated and involve a number of events. These events might not be equally important. Therefore, the mined events should be ranked with respect to their contributions. In this paper, we analyze public sentiment variations on Twitter and mine possible reasons behind such variations. To track public sentiment, we combine two state-of-the-art sentiment analysis tools to obtain sentiment information towards interested targets (e.g., “Obama”) in each tweet. Based on the sentiment label obtained for each tweet, we can track the public sentiment regarding the corresponding target using some descriptive statistics (e.g., Sentiment Percentage).

Then it will associate each remaining tweet in the variation period with one reason candidate and rank the reason candidates by the number of tweets associated with them. Experimental results on real Twitter data show that our method can outperform baseline methods and effectively mine desired information behind public sentiment variations. In summary, the main contributions of this paper are two-folds:

(1) To the best of our knowledge, our study is the first work that tries to analyze and interpret the public sentiment variations in microblogging services.

(2) Two novel generative models are developed to solve the reason mining problem. The two proposed models are general: they can be applied to other tasks such as finding topic differences between two sets of documents.

**MODELS FOR SENTIMENT VARIATION ANALYSIS**

To analyze public sentiment variations and mine possible reasons behind these variations, we propose Stanford NLP based model:

It is hard to find the exact causes for sentiment variation. However, it is possible to find clues via analyzing the relevant tweets within the variation period, since people often justify their opinion with supporting reasons. For example, if we want to know why positive sentiment on “Obama” increases, we can analyze the tweets with positive sentiment in the changing period and dig out the underlying events/topics co-occurring with these positive opinions. We consider the emerging events or topics that are strongly correlated with sentiment variations, as possible reasons. Mining such events/topics is not trivial. Topics discussed before the variation period may continue receiving attention for a long time. Therefore, we need to make use of the tweets generated just before the variation period to help “eliminate” these background topics. We can formulate this special topic mining problem as follows: given two document sets, a background set B and a foreground set T, we want to mine the special topics inside T but outside B. In our reason mining task, the foreground set T contains tweets appearing within the variation period and the background set B contains tweets appearing before the variation period. Note that this problem setting is general: it has applications beyond sentiment analysis.

To solve this topic mining problem, we develop a generative model called Hybrid Rules Based Extraction (SENTIMENT ANALYSIS Model). Benefiting from the reference role of the background tweets set, SENTIMENT ANALYSIS can distinguish the foreground topics out of the background or noise topics. Such foreground topics can help reveal possible reasons of the sentiment variations, in the form of word distributions.

**Proposed Architecture**

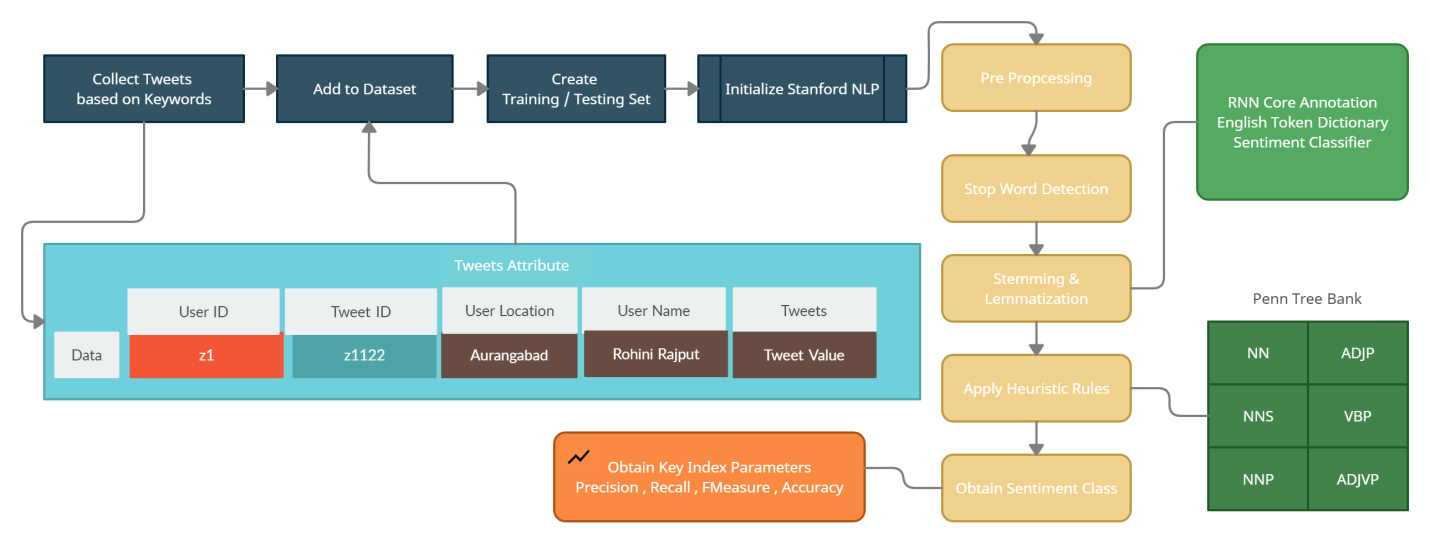


Figure 1.0 System Architecture

**Tweets Extraction Process**

Extracting Tweets from twitter based on the query keywords. In this module based on the twitter access key and consumer key, we are going to extract tweets based on the query keyword. Based on the Hash tag Retrieve data. Twitter messages are taken as input, as twitter messages are very informal , these messages are filtered out using techniques like, url filtering, Slang words translation, Non-English tweets filtering, stopwords removal. Preprocessing of Tweet is done by removing unnecessary words, removing hyperlink, removing special characters and filtering data for identifying sentiment value.

In this module we have keep first initial positive, negative and neutral words ,based on this initial expansion we are going to give positive, negative and neutral count for the words in the tweet and finally we will detect the sentiment of the tweet.

To extract tweets related to the target, we can go through the whole dataset and extract all the tweets which contain the keywords of the target. Compared with usual dataset containing tweets are generally less formal and repeatedly written in an adhoc manner. Sentiment analysis tools applied on raw tweets repeatedly achieve very poor performance in most cases. Therefore, pre-processing techniques on tweets are needed for obtaining satisfactory outcomes on sentiment analysis. The Stanford NLP is used for this purpose and the messages are labelled as positive or negative or neutral sentiment.

**Tracking Sentiment Variation**

Once obtaining the sentiment labels of all extracted tweets regarding a target, The Proposed system can track the sentiment variation using various descriptive statistics. Here the percentage of positive or negative tweets among all the extracted tweets is adopted as an indicator for tracking sentiment variation over time [1]. Based on these descriptive statistics, sentiment variations can be found using various heuristics (Example- the percentage of positive/ negative tweets increases for more than 50%) [1]. We have used Stanford NLP Framework that works with PennTree bank and utilizes PTB Tokenizer.

Initially Penn Treebank style tokenization of English text was extended to handle both other languages and noisy web-style text then it was implemented as deterministic finite automaton. Fast and efficient - supposedly 1,000,000 tokens per second are processed by the framework. Sentence splitting is a deterministic consequence of tokenization: sentence ends when a sentence-ending character (., !, ?) is found which is not grouped with other tokens.

Annotator pipeline is used, once the text has to processed further, a pipeline offers more control and functionality.

Properties props = new Properties();

props.setProperty("annotators", "tokenize");

StanfordCoreNLP pipeline = new StanfordCoreNLP(props);

Annotation annotation = new Annotation("This is a sentence to be tokenized"); pipeline.annotate(annotation);

The Properties class = persistent set of properties that can be saved to or loaded from a stream

The property list stores key and values as Strings

The StanfordCoreNLP class is designed to apply multiple Annotators (or functions) to an Annotation via the annotate() method

Annotation = a representation of text, either raw or processed

CoreLabel = Map from keys to values. It provides convenient methods to access tags, lemmas, etc.

TokensAnnotation = CoreMap key for getting the tokens contained by the Annotation.

**Foreground and Background Aspects**

Foreground and Background LDA (SENTIMENT ANALYSIS ), can filter out background topics and extract foreground topics from tweets in the variation period, with the help of supplementary set of background tweets generated just before the variation. The SENTIMENT ANALYSIS algorithm is used for this purpose.

SENTIMENT ANALYSIS first extracts representative tweets for the foreground topics (obtained from SENTIMENT ANALYSIS ) as reason candidates. Then it will associate each remaining tweet in the variation period with one reason candidate and rank the reason candidates by the number of tweets associated with them [1]. The SENTIMENT ANALYSIS algorithm is used for this purpose.

**Proposed Model Psuedocode**

1. Go through each tweet, and randomly assign each word in the tweet to one of the K topics.

2. Notice that this random assignment already gives you both topic representations of all the tweets and word distributions of all the topics.

3. So to improve on them, for each tweet d...

4. foreach word w in d... And for each topic t, compute two things:

* p(topic t | tweet d) = the proportion of words in tweet d that are currently assigned to topic t,
* p(word w | topic t) = the proportion of assignments to topic t over all tweets that come from this word w.

5. Reassign w a new topic, where you choose topic t with probability p(topic t | tweet d) \* p(word w | topic t.

6. In other words, in this step, we're assuming that all topic assignments except for the current word in question are correct, and then updating the assignment of the current word using our model of how tweets are generated.

7. After repeating the previous step a large number of times, we will eventually reach a roughly steady state where your assignments are pretty good. So use these assignments to estimate the topic mixtures of each tweet (by counting the proportion of words assigned to each topic within that tweet) and the words associated to each topic (by counting the proportion of words assigned to each topic overall).

**Psuedo Steps for generative process**

We automatically select reason candidates by finding the most relevant tweets for each foreground topic learnt from SENTIMENT ANALYSIS , using the following measure:

1. Get Foreground topics, and find relevance of it tweets using below formula:

where φ kf f is the word distribution for the foreground topic kf and i is the index of each non-repetitive word in tweet t.

2. For each tweet find word distribution and Find word relevance.

3. Extract tweets which have more relevance.

4. Display its count and tweet.

The experiment has been carried on a I3 processor using Java platform on 4 GB of RAM as primary memory. The tweets were collected using auth and consumer key with Twitter API using a Twitter account.

#Hashtag method was used to collect tweets pertaining to a specific topic and these tweets were then pre processed to clean up noise and other degenerative terms so as to obtain accurate sentiments of the sentences.

To obtain aspects we have used Penn tree bank to identify nouns:

Table 1.0 Penn Tree bank notations

|  |  |
| --- | --- |
| POS Tag | Expression |
| CD | CARDINAL |
| FW | FOREIGNWORD |
| UH | INTERJECTION |
| LS | LISTMARKER |
| NN, NNS, NNP, NNPS | NOUN |
| PRP | INTERJECTION |
| MD | MODAL |
| PB, RBR, RBS | ADVERBS |

Tweets are collected along with other attributes such as USERNAME, LOCATION, TWEETID, DATE-TIME and TWEET CONTENTS. These records are then parsed using a Line Separator Parser and then processed for identifying variations. Implementing heuristic grammar rules has helped to improve total gain of aspects along with implementation of finding topic words.