Twitter for Sentiment Analysis

Anitta Ann Raju, Ayush Kumar Priyadarshan

1GD18CS008(Gopalan College of Eng. & Mgmt.), 1VI18CS015(Vemana institute of technology)
Bangalore, Karnataka
anittaannraju@gmail.com, aayushkmr0109@gmail.com

Abstract

The use of microblogging platforms such as Twitter, Reddit has seen a enormous growth in the recent years.
This has indefinitely led the companies and media organizations to seek ways to mine this data from these sites and use it to understand the sentiment (information about what people think and feel) the public associate with their product and services. Twitter is a huge online platform that offers users to post short messages on their platform (called tweets) with a maximum length of 140 characters, with 200 million registered users - of which 50 million of them log on twitter on a daily basis - generating nearly 250 million tweets per day. There are serval companies out there that use these tweets from Twitter as the data to analyse the sentiment of the public (like Twitratr - twitrratr.com, tweetfeel - www.tweetfeel.com, and Social Mention - www.socialmention.com). Our aim is to build a model that uses tweets from twitter as raw data to analyse the sentiment the public associates with a certain keyword. The sentiment expressed can be: positive, negative or neutral. Due to the large amount of usage of twitter we hope to achieve a reflection of public sentiment by analysing the sentiments expressed in the tweets. Analysing the public sentiment is important for many applications such as firms trying to find out the response of their products in the market, predicting political elections and predicting socioeconomic phenomena like stock exchange.

1. Introduction

Microblogging, today, has changed the way people communicate, with various microblogging services people tend to use them instead of the traditional services. The online platform has opened a door for people to express their opinion on different topics from the comfort of their own space. In social media there is an abundance of opinion information available this can be used as data to analyse the sentiment. Sentiment analysis refers to the broad area of natural language processing which deals with the computational study of opinions, sentiments and emotions expressed in text.

Sentiment Analysis (SA) or Opinion Mining (OM) aims at learning people's opinions, attitudes and emotions towards an entity. The entity can represent individuals, events or topics. An immense amount of research has been performed in the area of sentiment analysis. But most of them focused on classifying formal and larger pieces of text data like reviews, recently though the domain has shifted to micro-blogging sites like twitter.

Using sentiment analysis techniques, the polarity of the data can be found; such as positive, negative, or neutral by analysing the text. this has been useful for companies or organisations to get their customer's opinions on their products, or predicting outcomes of elections, and getting movie rating by reviews posted. This information helps the companies in making future decisions.

Many of the traditional sentiment analysis models uses the bag of words method. The bag of words technique does not consider language morphology, which leads to incorrectly classifying two phrases to have the same meaning simply because they share the same bag of words. Therefore, it is important to consider the relationship between individual words instead of between collection of words. The overall sentiment is determined by combining the sentiment determined for each word using a function. Bag of words also ignores word order, which leads to phrases with negation in them to be incorrectly classified. Other techniques in sentiment analysis include Naive Bayes, Maximum Entropy, and Support Vector Machines. We have implemented the Naive Bayes technique in our project.

2. Literature Survey

2.1 Limitation to Prior Systems

Sentiment analysis in the context of microblogging is a relatively new study field, thus there is plenty of space for more investigation. Prior to this, a substantial amount of work on sentiment analysis of user reviews, documents, web blogs/articles, and general phrase level sentiment analysis had been done. They are distinct from Twitter since tweets are limited to 140 characters each this forces the user their express opinion in very short compressed text. Although supervised learning approaches such as Naive Bayes and Support Vector Machines produce the greatest results in sentiment classification, the manual labelling necessary for the supervised approach is quite expensive. Unsupervised and semi-supervised techniques have received some attention, but there is still potential for improvement. Various researchers experimenting with new features and classification techniques frequently compare their results to baseline performance. In order to select the best features and most efficient classification techniques for specific applications, proper and formal comparisons between these results obtained through different features and classification techniques are required

2.2 Proposed System

Many language processing jobs are classification tasks, but our classes are thankfully lot easier to describe than Borges'. The naive Bayes algorithms classification is demonstrated on an important classification problem in this classification: The task of classifying an entire text by assigning it a text categorization label from a list of categories is known as text categorization. We focus on one typical text categorization problem, sentiment analysis, the ex-sentiment analysis Suction of sentiment, or a writer's positive or negative attitude toward a certain item. A web review of a movie, book, or product reflects the author's feelings about the product, whereas an editorial or political piece reflects feelings about a candidate or political action. Automatically extracting customer sentiment is critical for every type of product marketing, while gauging public opinion is critical for politics and market forecasting. A binary classification job is the most basic form of sentiment analysis, and the words in the review provide great cues. Consider the phrases below, which were collected from favourable and bad movie and restaurant reviews. Words like great, richly, awesome, and pathetic, and awful and ridiculously are very informative cues:

+ ...zany characters and richly applied satire, and some great plot twists
− It was pathetic. The worst part about it was the boxing scenes...
+ ...awesome caramel sauce and sweet toasty almonds. I love this place!
− ...awful pizza and ridiculously overpriced...

Naive Bayes is a probabilistic classifier, which means that for a document d, the classifier gives the class ˆ c with the highest posterior probability given the document, out of all classes c ∈ C. The hat notation is used in Eq. 1 to denote "our best guess of the correct class."

$c = \text{argmax } P(c|d) \text{ where } c \in C$

The concept of Bayesian inference has been there since Bayes' work in 1763, and Mosteller and Wallace were the first to apply it to text classification (1964). The idea behind Bayesian classification is to apply Bayes' rule to convert Eq.6.1in to various probabilities with useful features. The Bayes rule is shown in Eq. 2; it gives us a way to break down any conditional probability P(x|y) into three other probabilities:

$$P(x|y) = P(y|x)P(x) / P(y)$$

The algorithm of sentiment analysis using Naive Bayes Classification:

```
function BOOTSTRAP(x,b) returns p-value(x)
        Calculate δ(x)
        for i = 1to b do
                for j = 1to n do
                # Draw a bootstrap sample x∗(i) of size n
                # Select a member of x at random and
        add
                # it to x∗(i)
                Calculate δ(x∗(i))
        for each x∗(i)
                s←s + 1ifδ(x∗(i)) > 2δ(x)
p-value(x)≈ s b
return p-value(x)
```

• A lot of language processing jobs can be classified as classification tasks. Understand how to model the class based on the observation.
• Text categorization includes tasks like sentiment analysis, spam detection, email classification, and authorship attribution, in which an entire text is allocated a class from a finite collection.
• Sentiment analysis classifies a text as reflecting a writer's favourable or negative attitude (sentiment) toward a certain object.
• The generative model Naive Bayes makes the bag of words assumption (position doesn't matter) and the condition all in dependency assumption (words are conditionally independent of each other given the class)
• Naive Bayes with binarized features seems to work better for many text classification tasks.

The TextBlob package for Python is a convenient way to do a lot of Natural Language Processing (NLP) tasks. For example:

```
From textblob import TextBlob
TextBlob("not a very great calculation").sentiment
```

This tells us that the English phrase "not a very great calculation" has a polarity of about -0.3, meaning it is slightly negative, and a subjectivity of about 0.6, meaning it is fairly subjective. There are helpful comments like this one, which gives us more information about the numbers we're interested in:

```
# Each word in the lexicon has scores for:
# 1) polarity: negative vs. positive (-1.0 => +1.0)
# 2) subjectivity: objective vs. subjective (+0.0 => +1.0)
# 3) intensity: modifies next word? (x0.5 => x2.0)
```

The lexicon it refers to is in en-sentiment.xml, an XML document that includes the following four entries for the word “great”.

<word Form=”great” cornetto svnset id=”n_a-525317” wordnet id=”a-01123879” pos=”JJ” sense=”very good” polarity=”1.0” subjectivity=”1.0” intensity=”1.0” confidence=”0.9” />

<word Form=”great” wordnet id=”a-011238818” pos=”JJ” sense=”of major significance or importance” polarity y=”1.0” subjectivity=”1.0” intensity=”1.0” confidence=”0.9” />

<word Form=”great” wordnet id=”a-01123883” pos=”JJ” sense=”relativity large in size or number or extent” polarity =”0.4” subjectivity=”0.2” intensity=”1.0” confidence=”0.9” />

<word Form=”great” wordnet id=”a-01677433” pos=”JJ” sense=”remarkable or out of the ordinary in degree or magnitude or effect” polarity =”0.8” subjectivity=”0.8” intensity=”1.0” confidence=”0.9” />

There's also "confidence" in addition to the polarity, subjectivity, and intensity indicated in the previous paragraph, but I don't see it utilised elsewhere. It's all the same part of speech (JJ, adjective) in the case of "great," and the senses are natural language and aren't employed. To make it easier to read:

Word	Polarity	Subjectivity	Intensity
Great	1.0	1.0	10
Great	1.0	1.0	1.0
Great	0.4	0.2	1.0
Great	0.8	0.8	1.0

When calculating sentiment for a single word, TextBlob uses a sophisticated technique known to Mathematicians as “averaging”.

TextBlob(“great").sentiment

Sentiment(polarity=0.8, subjectivity=0.75)

2.3 Related Works

Opinion mining and sentiment analysis have become popular study topics as the number of blogs and social networks has grown. In this paper, a fairly wide summary of existing work was presented (Pang and Lee, 2008). The authors describe existing methodologies and approaches for opinion-oriented information retrieval in their survey. However, few opinion mining studies have taken into account blogs, and even fewer have addressed microblogging. The authors of (Yang et al., 2007) construct a corpus for sentiment analysis using web-blogs and employ emotion icons given to blog entries as indications of users' mood. The authors used SVM and CRF learners to identify feelings at the sentence level, then looked into a variety of ways to assess the document's overall sentiment. As a result, the winning strategy is defined by considering the sentiment of the document's last sentence as the document's emotion.

J. Read (Read, 2005) employed emoticons like “:-)” and “:-(” to create a sentiment classification training set. The author gathered emoticon-containing texts from Usenet newsgroups for this project. The data was split into two categories: "positive" (texts with cheerful emoticons) and "negative" (texts with sad or angry emoticons). SVM and Naive Bayes, both trained with emoticons, were able to achieve up to 70% accuracy on the test set.

The authors of (Go et al., 2009) used Twitter to acquire training data before doing a sentiment search. The strategy is similar to that of (Read, 2005). The authors create corpora by utilising emoticons to generate "positive" and "negative" samples, which they then classify using various classifiers. The Naive Bayes classifier with a mutual information measure for feature selection produced the best results. On their test set, the authors were able to get up to 81 percent accuracy. With three classes ("negative," "positive," and "neutral"), however, the technique performed poorly.

3. Functionality and Design

The flow of the model is as follows:

I. Data Acquisition
II. Human Labelling
III. Tweet Classification
IV. Sentiment in text and graphical format

3.1 Data Acquisition:

The Python module “tweepy,” which provides a package for simple twitter streaming API, is used to obtain data in the form of raw tweets. For this, we first created a Twitter developer account and then used the API keys and token keys to connect our project to the twitter platform, enabling us to pull tweets for our analysis.

There are two ways to obtain tweets with this API: SampleStream and FilterStream. SampleStream merely provides a short, random sample of all tweets that are being streamed in real time. FilterStream sends tweets that meet a set of criteria.

It has the ability to filter tweets based on three criteria:

• Tracking/searching for a certain term in tweets
• Tracking/searching for a specific Twitter user based on their name
• Tweets coming from specific location(s) (only for geotagged tweets).

Since we don’t have any restrictions in our collection, we will stick to SampleStream mode.

3.2 Human Labelling

Following the retrieval of the relevant tweets, we must analyse the sentiment polarity of each tweet using the

TextBlob module's analysis.sentiment.polarity. This polarity is used to categorise tweets as "positive," "negative," or "neutral." It's "positive" if it's greater than zero, "neutral" if it's equal to zero, and "negative" if it's less than zero.

For the purpose of human labelling, we made three copies of the tweets so that they can be labelled by three individual sources. This is done so that we can take average opinion of people on the sentiment of the tweet and in this way the noise and inaccuracies in labelling can be minimized. Generally speaking, the more copies of labels we can get the better it is, but we have to keep the cost of labelling in our mind,
Hence, we reached at the reasonable figure of three.
We labelled the tweets in four classes according to sentiments
expressed/observed in the tweets: positive, negative, neutral/objective (including ambiguous).

We gave the following guidelines to our labellers to help them in the labelling process:

• Positive: If the entire tweet has a positive / happy / excited / joyful attitude or if something is mentioned with positive connotations. Also, if more than one sentiment is expressed in the tweet but the positive sentiment is more dominant. Example: "4 more years of being in shithole Australia then I move to the USA! :D"

• Negative: If the entire tweet has a negative/sad/displeased attitude or if something is mentioned with negative connotations. Also ,if more than one sentiment is expressed in the tweet but the negative sentiment is more dominant. Example: "I want an android now this iPhone is boring :S".

• Neutral/Objective: If the creator of tweet expresses no personal sentiment/opinion in the tweet and merely transmits information. Advertisements of different products would be labelled under this category. Example: "US House Speaker vows to stop Obama contraceptive rule... http://t.co/cyEWqKlE".

3.3. Tweet Classification
Pattern classification is the process through which data is divided into different classes according to some common patterns which are found in one class which differ to some degree with the patterns found in the other classes. The ultimate aim of our project is to design a classifier which accurately classifies tweets in the following three sentiment classes: positive, negative, neutral.

Contextual sentiment analysis and generic sentiment analysis are two types of sentiment classifications that can be used in this area. Contextual sentiment analysis involves classifying specific parts of a tweet based on the context provided. For example, in the tweet "4 more years in shithole Australia before I move to the USA," a contextual sentiment classifier would classify Australia as negative and the United States as positive.

General sentiment analysis, on the other hand, considers the overall sentiment of the text (in this case, a tweet). Thus, for the tweet mentioned earlier since there is an overall positive attitude, an accurate general sentiment classifier would identify it as positive. For our particular project we will only be dealing with the latter case, i.e., of general (overall) sentiment analysis of the tweet as a whole.

We use the Naïve Bayes classification in our project. There are, of course, serval other classification techniques out there just as good or even better, like:
• K-Means Clustering
• Support Vector Machine
• Logistic Regression
• K Nearest Neighbours
• Rule Based Classifiers

3.4. Sentiment in text and graphical format
Once, the classification part is done, its relatively easy from there. Next, we calculate the percent of tweets in each cluster as oppose to the total tweets retrieved. After that, we determine which sentiment is dominate overall and take that as the overall sentiment associated with the keyword. The graphical representation is also given to get an in-depth understanding of the sentiment associated with that word.

4. Conclusion and Future Enhancements

Sentiment analysis, particularly in the context of microblogging, is still in its early stages and far from complete. As a result, we present a handful of ideas that we believe are worth pursuing in the future and could lead to even better performance. We've just dealt with the most basic unigram models so far; we may improve them by adding more information, such as the proximity of the word to a negation word. We may define a window before the word under consideration (for example, two or three words), and the effect of negation could be incorporated into the model if it falls within that window. If the negative is close to the word, for example, it may simply flip the polarity of that word; nevertheless, the further the negation is from the word, the less the ifs effect should be.

Apart from that, we are now concentrating on unigrams, while the impact of bigrams and trigrams may be investigated in the future. When bigrams are employed in conjunction with unigrams, as mentioned in the literature review section, performance is frequently improved.

We are concentrating on general sentiment analysis in this study. There is potential for work in the field of sentiment analysis with only a partial understanding of the context. For example, we discovered that users generally use our website for specific types of keywords, which can be classified into two groups: politics/politicians, celebrities, products/brands, sports/sportsmen, and media/movies/music. So we can try to perform separate sentiment analysis on tweets that only belong to one of these classes (i.e. the training data is not general but specific to one of these categories) and compare the results to what we get if we use general sentiment analysis on it instead.

Last but not least, we can try to simulate human confidence in our system. For example, if we have 5 human labellers labelling each tweet, we can plot the tweet in the 2-dimensional objectivity / subjectivity and positivity / negativity planes while distinguishing between tweets where all 5 labels agree, tweets where only 4 labels agree, tweets where only 3 agree, and tweets where no majority vote is reached. We could create our own cost function to generate optimised class boundaries, such that the highest weightage is given to tweets in which all five labels agree, and as the number of agreements decreases, so do the weights assigned. The effects of human confidence can thus be visualised in sentiment analysis.

5. References

[1] Albert Biffet and Eibe Frank. Sentiment Knowledge Discovery in Twitter Streaming Data. Discovery Science, Lecture Notes in Computer Science, 2010, Volume 6332/2010, 1-15, DOI: 10.1007/978-3-642-16184-1_1

[2] Alec Go, Richa Bhayani and Lei Huang. Twitter Sentiment Classification using Distant Supervision. Project Technical Report, Stanford University, 2009.

[3] Alexander Pak and Patrick Paroubek. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In Proceedings of international conference on Language Resources and Evaluation (LREC), 2010.

[4] Andranik Tumasjan, Timm O. Sprenger, Philipp G. Sandner and Isabell M. Welpe. Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. In Proceedings of AAAI Conference on Weblogs and social media (ICWSM), 2010.

[5] Bo Pang, Lillian Lee and Shivakumar Vaithyanathan. Thumbs up? Sentiment Classification using Machine Learning Techniques. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2002.

[6] Chenhao Tan, Lilian Lee, Jie Tang, Long Jiang, Ming Zhou and Ping Li. User Level Sentiment Analysis Incorporating Social Networks. In Proceedings of ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD), 2011.

[7] Efthymios Kouloumpis, Theresa Wilson and Johanna Moore. Twitter Sentiment Analysis: The Good the Bad and the OMG! In Proceedings of AAAI Conference on Weblogs and Social Media (ICWSM), 2011.

[8] Hatzivassiloglou, V., & McKeown, K.R.. Predicting the semantic orientation of adjectives. In Proceedings of the 35th Annual Meeting of the ACL and the 8th Conference of the European Chapter of the ACL, 2009.

[9] Johann Bollen, Alberto Pepe and Huina Mao. Modelling Public Mood and Emotion: Twitter Sentiment and socio-economic phenomena. In Proceedings of AAAI Conference on Weblogs and Social Media (ICWSM), 2011.

[10] Luciano Barbosa and Junlan Feng. Robust Sentiment Detection on Twitter from Biased and Noisy Data. In Proceedings of the international conference on Computational Linguistics (COLING), 2010.

[11] Peter D. Turney. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL), 2002.

[12] Rudy Prabowo and Mike Thelwall. Sentiment Analysis: A Combined Approach. Journal of Infometrics, Volume 3, Issue 2, April 2009, Pages 143-157, 2009.

[13] Samuel Brody and Nicholas Diakopoulus Using Word Lengthening to Detect52 Sentiment in Microblogs. In Proceedings of Empirical Methods on Natural Language Processing (EMNLP), 2011.

[14] Soo-Min Kim and Eduard Hovy. Determining the Sentiment of Opinions. In Proceedings of International Conference on Computational Linguistics (ICCL), 2004.

[15] Stefano Baccianella, Andrea Esuli, Fabrizio Sebastiani. SENTIWORDNET 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In Proceedings of international conference on Language Resources and Evaluation (LREC), 2010.

[16] Theresa Wilson, Janyce Wiebe and Paul Hoffmann. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In the Annual Meeting of Association of Computational Linguistics: Human Language Technologies (ACL-HLT), 2005.

[17] Ian H. Witten, Eibe Frank & Mark A. Hall. Data Mining – Practical Machine
Learning Tools and Techniques.

[19] Ricgard O. Duda, Peter E. Hart & David G. Stork: Pattern Classification.

[19] Steven Bird, Even Klein & Edward Loper. Natural Language Processing with
Python.