Summary of “Twitter as a corpus for Sentiment Analysis and Opinion Mining”

By Max Xie

1. Introduction

Sentiment analysis on microblogging on platforms such as Facebook and Twitter can answer many questions för companies, such as:

- What do people think about our product (service, company, etc.)
- How positive (or negative) are people about our product?
- What would people prefer our product to be like?

Why are microblogs a particular good source for general sentiment analysis of the public?

- Microblogging platforms are used by different people to express their opinion about different topics, thus it is valuable source of people’s opinions.
- Twitter contains an enormous number of text posts and it grows every day. The collected corpus can be arbitrarily large.
- Twitter’s audience varies from regular users to celebrities, company representatives, politicians, and even country presidents. Therefore, it is possible to collect text posts of users from different social and interests groups.
- Twitter’s audience is represented by users from many countries. Although users from U.S. are prevailing, it is possible to collect data in different languages.

After having collected 300000 text posts from Twitter, three categories could be made:

1. Texts containing positive emotions, such as happiness, amusement and joy.
2. Texts containing negative emotions, such as sadness, anger or disapplointment.
3. Objective tests that only state a fact or do not express any emotions.

1.1 Contributions

The report presents the whole process in four steps:

1. Gather all the data, positive, negative and neutral
2. Perform statistical linguistic analysis on the collected corpus
3. Build a sentiment classification system for microblogging
4. Conduct experimental evaluations and tests on a set of real microblogging posts.

2. Related work

Related reports have shown that SVM and Naïve Bayes are typically good methods for this type of classification (positive, negative, neutral) two different reports have achieved 70% and 81% accuracy on their test sets respectively.

3. Corpus collection

To collect negative and positive sentiments, they queried Twitter for two types of emoticons:

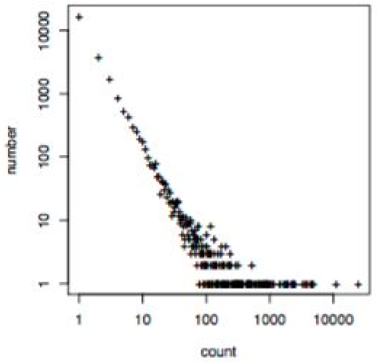
1. Happy emoticons: “:-)”, “:)”, “=)”, “:D” etc.
2. Sad emoticons: “:-(“, “:(“, “=(“, “;(“ etc.

Since microblogging generally are very short, we make the assumption that the emoticon represent the emotion of the entire message.

In order to get the neutral/objective tweets they extracted tweets from big newspapers. *I personally think that this is a weird way to get neutral data.*

4. Corpus analysis

First, they checked the distribution of words frequencies in the corpus. When it is plotted it is apparent that the distribution follows Zipf's law. The distributions only counted the lemmas of the different words that occurred. So that "is" and "be" was considered the same word.

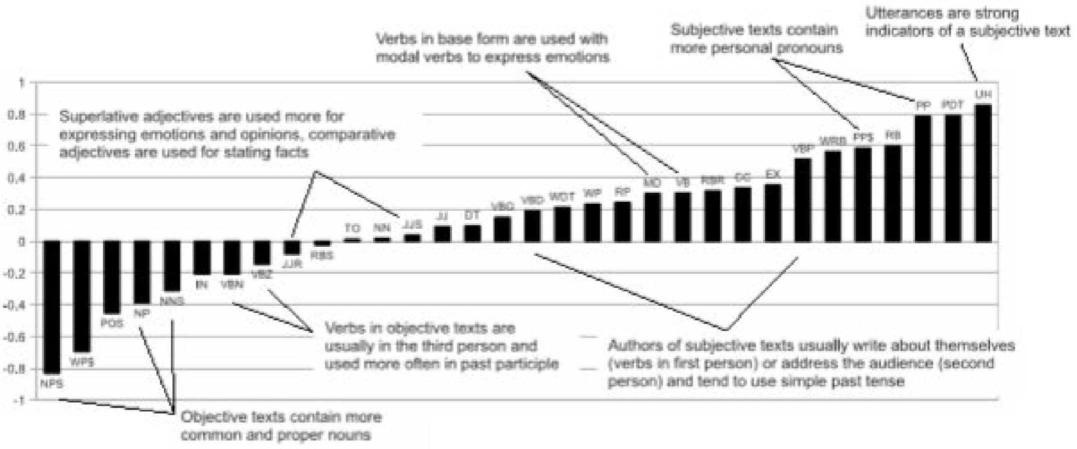


4.1 Subjective vs. objective

We want to examine if there are any differences in the usage of tags/classes in positive/negative texts v.s. neutral texts.

$$P^T_{1,2} = \frac{N^T_1 - N^T_2}{N^T_1 + N^T_2}$$

P is measures the difference in tag/classes usages between subjective and objective texts. Luckily the statistics is given by the report, and we can observe that proper nouns (NPS, NP, NNS) are more common in objective texts. Authors of subjective texts use more often personal pronouns (PP, PP$) they also usually describe themselves in first person and address the audience as second person, while verbs in objective texts usually are in third person. You can see the complete statistics below.



Verbs in base form are used with modal verbs to express emotions

Subjective texts contain more personal pronouns

Utterances are strong indicators of a subjective text

Superlative adjectives are used more for expressing emotions and opinions, comparative adjectives are used for stating facts

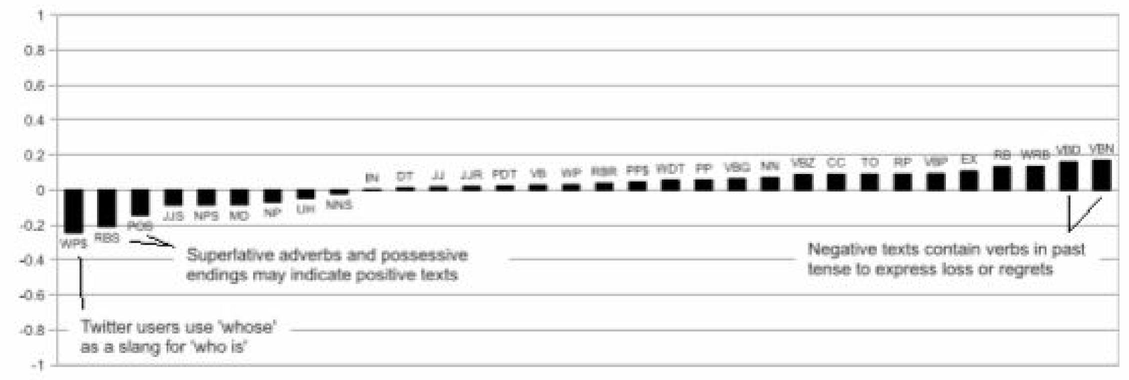
Verbs in objective texts are usually in the third person and used more often in past participle

Authors of subjective texts usually write about themselves (verbs in first person) or address the audience (second person) and tend to use simple past tense

Objective texts contain more common and proper nouns

Figure 2: P^T values for objective vs. subjective

Negative values mean that these types of tags are common in objective texts. Positive values indicate high frequency in subjective texts.



Negative values in this graph means that those tags often are associated with positive texts. While positive values indicate tags commonly seen in negative texts. We can furthermore see that superlative adverbs such as “most” and “best” are more common in positive texts, and the negative set more often contains verbs in the past tense (BVN, VBD) because many authors express their negative sentiments about loss or disappointment. Frequent verbs are: “missed”, “bored”, “gone”, “lost”, “stuck”, “taken”.

In order to guarantee homogeneity of the corpus we randomly divide the data into two parts and compare their distributions. If the distributions are somewhat similar the data also have a high probability of being homogenous.

5. Training the classifier

5.1 Feature extraction

There are reports where unigrams have outperformed bigrams when it comes to sentiment analysis of movie reviews. While bigrams and trigrams have worked better for product-review classifications. The report has a hypotheses that the frequency of different keywords are a good feature, although there are reports that it is better to measure the presence of different keywords rather than its frequency. The conclusion is that it still is not very clear what features make a good classifier.

The report however concluded that there are four main steps for making a sentiment analysis classifier. They are the following:

1. Filtering: Remove all URL links, Twitter user names, twitter special words and emoticons.
2. Tokenization: Segment the text by splitting it by spaces and punctuation marks, and form a bag of words. However, they let words like “don’t”, “I’ll”, “she’d” to remain being just one word.
3. Removing stopwords: Remove articles such as “a”, “an”, “the” from the bag of words.
4. Constructing n-grams: A negation such as “no” or “not” is attached to the preceding word and they are together treated as if it were one word. e.g. the sentence “I do not like fish” will form two bigrams “I do+not”, “do+not like”, “not+like fish”. Negations play a special role in an opinion and sentiment expression.

5.2 Classifier

The report builds the sentiment classifier using a multinomial Naïve Bayes classifier. They also tried using SVM and CRF, but the Naïve Bayes seemed to yield the best results. Naïve Bayes is based on Bayes' theorem:

$$P(s|M) = \frac{P(s) \cdot P(M|s)}{P(M)}$$

Where *s* is a sentiment and *M* is a Twitter message. Because they have equal sets of positive, negative and neutral messages, they can simplify the equation:

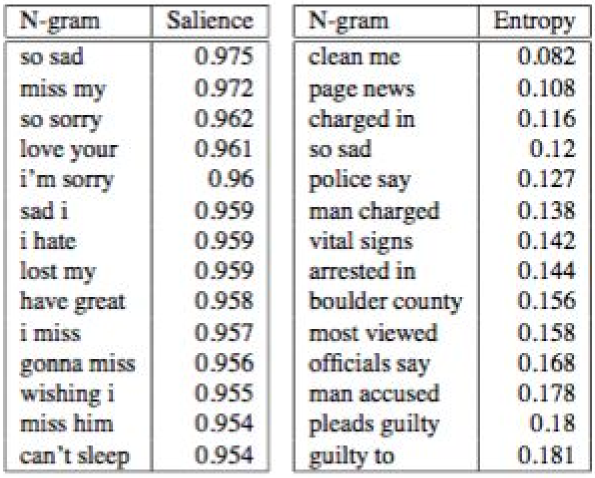
$$P(s|M) = \frac{P(M|s)}{P(M)}$$

$$P(s|M) \sim P(M|s)$$

Since the probability of a specific category becomes equal.

Further the report train two separate Bayes classifiers, which use different features:

1. Presence of n-grams. (Binary feature)



N-gram	Salience	N-gram	Entropy
so sad	0.975	clean me	0.082
miss my	0.972	page news	0.108
so sorry	0.962	charged in	0.116
love your	0.961	so sad	0.12
i'm sorry	0.96	police say	0.127
sad i	0.959	man charged	0.138
i hate	0.959	vital signs	0.142
lost my	0.959	arrested in	0.144
have great	0.958	boulder county	0.156
i miss	0.957	most viewed	0.158
gonna miss	0.956	officials say	0.168
wishing i	0.955	man accused	0.178
miss him	0.954	pleads guilty	0.18
can't sleep	0.954	guilty to	0.181

2. Part-of-speech distribution information.

This estimates the probability of POS-tags presence within different sets of texts and uses it to calculate posterior probability. Although POS-tags and presence of n-grams are correlated, we ignore this fact out of simplicity.

$$P(s|M) \sim P(G|s) \cdot P(T|S)$$

$$P(G|s) = \prod_{g \in G} P(g|s)$$

$$P(T|s) = \prod_{t \in G} P(t|s)$$

$$P(s|M) \sim \prod_{g \in G} P(g|s) \cdot \prod_{t \in G} P(t|s)$$

G is a set of n-grams representing the message, *T* is a set of POS-tags of the message. We also assume that POS-tags and n-grams are conditionally independent.

$$L(s|M) = \sum_{g \in G} log(P(g|s)) + \sum_{t \in G} log(P(t|s))$$

Finally, we calculate the log-likelihood of each sentiment.

5.3 Increasing accuracy

One good method is to remove n-grams that occur evenly in all texts/datasets regardless of its sentiment. To fix this there are mainly two strategies:

1. Calculate entropy

$$entropy(g) = H(p(S|g)) = -\sum_{i=1}^{N} p(S_i|g) \log p(S_i|g)$$

Where *N* is the number of sentiments, in our case 3. A high entropy value indicates that the distributions of the n-grams is uniform regardless of sentiment. Therefore, n-grams with high entropy should not contribute much to the classification. A low value indicates the opposite. We could control this by using a threshold, where we ignore all n-grams with an entropy higher than the threshold. This will affect our recall, but keep our accuracy higher, which is more relevant since we have a large dataset.

2. Calculate salience

$$salience(g) = \frac{1}{N} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} 1 - \frac{\min(P(g|s_i), P(g|s_j))}{\max(P(g|s_i), P(g|s_j))}$$

N = 3, in our case. Low value indicate a low salience for the n-gram, and such n-grams should be ignored.

Using either one of these two strategies we get a new log-likelihood formula:

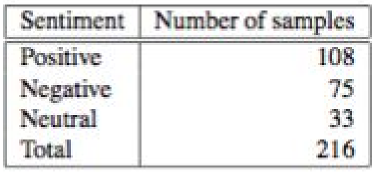
$$L(s|M) = \sum_{g \in G} log(P(g|s)) \cdot if(f(g) > \theta, 1, 0) + \sum_{t \in G} log(P(t|s))$$

Where *f(g)* is the entropy or the salience of an n-gram, and theta is the threshold value.

We could later see in the result that the best results were reached by calculating salience.

5.4 Data and methodology

This part of the report is left for testing the model.



Sentiment	Number of samples
Positive	108
Negative	75
Neutral	33
Total	216

$$accuracy = \frac{N(\text{correct classifications})}{N(\text{all classifications})}$$

$$decision = \frac{N(\text{retrieved documents})}{N(\text{all documents})}$$

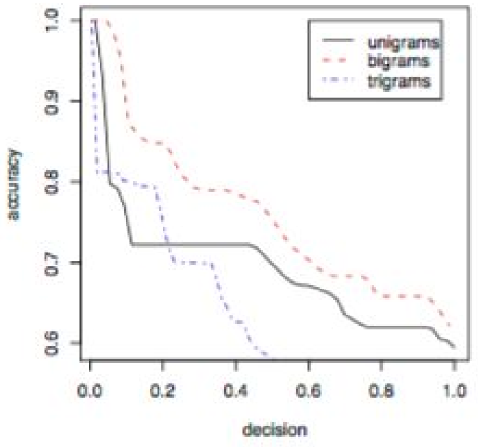


Figure 4: The comparison of the classification accuracy when using unigrams, bigrams, and trigrams

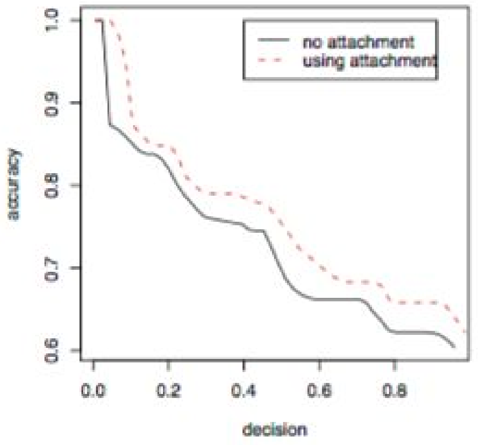


Figure 5: The impact of using the attachment of negation words

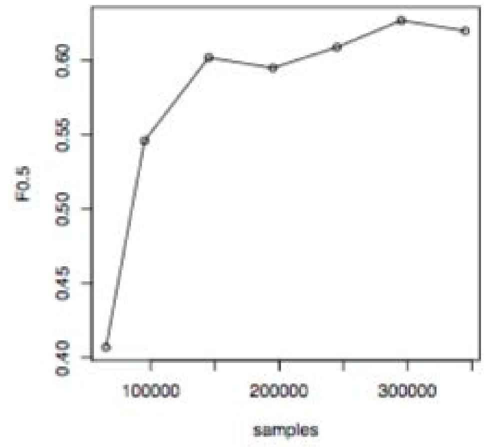


Figure 6: The impact of increasing the dataset size on the $F_{0.5}$-measure

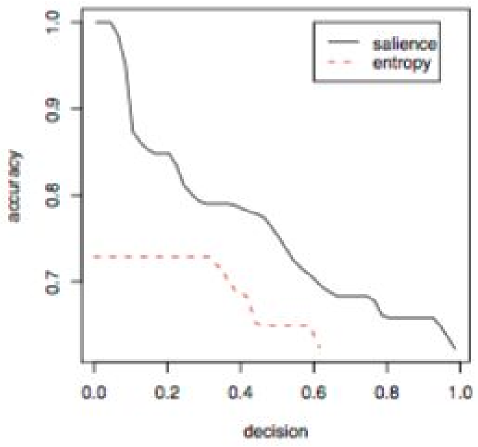


Figure 7: Salience vs. entropy for discriminating common n-grams

5.5 Results

The best results are achieved by using bigrams. They provide a good balance between a coverage (unigrams) and an ability to capture the sentiment expression patterns (trigrams).

In the valuation of the F-score, precision is replaced with accuracy and recall with decision.

$$F = (1+\beta^2)\frac{accuracy \cdot decision}{\beta^2 \cdot accuracy + decision}$$

Where *beta* is 0.5

We can also see that performance is increased as we increase the size of our dataset, but it reaches a plateau at around 150000 data points. We can also read that the salience strategy beats the entropy strategy.

6. Conclusion

They used TreeTagger for POS-tagging and from there observed a difference in distributions of tags/classes in positive, negative and neutral texts. Authors therefore use syntactic structures to describe emotions, and some POS-tags may be strong indicators of emotional text. The classifier used N-gram and POS-tags as features and could determine positive, negative and neutral sentiments.