Twitter as a Corpus for Sentiment Analysis and Opinion Mining

Alexander Pak, Patrick Paroubek

Université de Paris-Sud, Laboratoire LIMSI-CNRS, Bâtiment 508, F-91405 Orsay Cedex, France
alexpak@limsi.fr, pap@limsi.fr

Abstract

Microblogging today has become a very popular communication tool among Internet users. Millions of users share opinions on different aspects of life everyday. Therefore microblogging web-sites are rich sources of data for opinion mining and sentiment analysis. Because microblogging has appeared relatively recently, there are a few research works that were devoted to this topic. In our paper, we focus on using Twitter, the most popular microblogging platform, for the task of sentiment analysis. We show how to automatically collect a corpus for sentiment analysis and opinion mining purposes. We perform linguistic analysis of the collected corpus and explain discovered phenomena. Using the corpus, we build a sentiment classifier, that is able to determine positive, negative and neutral sentiments for a document. Experimental evaluations show that our proposed techniques are efficient and performs better than previously proposed methods. In our research, we worked with English, however, the proposed technique can be used with any other language.

1. Introduction

Microblogging today has become a very popular communication tool among Internet users. Millions of messages are appearing daily in popular web-sites that provide services for microblogging such as Twitter1, Tumblr2, Facebook3. Authors of those messages write about their life, share opinions on variety of topics and discuss current issues. Because of a free format of messages and an easy accessibility of microblogging platforms, Internet users tend to shift from traditional communication tools (such as traditional blogs or mailing lists) to microblogging services. As more and more users post about products and services they use, or express their political and religious views, microblogging web-sites become valuable sources of people’s opinions and sentiments. Such data can be efficiently used for marketing or social studies.

We use a dataset formed of collected messages from Twitter. Twitter contains a very large number of very short messages created by the users of this microblogging platform. The contents of the messages vary from personal thoughts to public statements. Table 1 shows examples of typical posts from Twitter.

As the audience of microblogging platforms and services grows everyday, data from these sources can be used in opinion mining and sentiment analysis tasks. For example, manufacturing companies may be interested in the following questions:

- 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?

Political parties may be interested to know if people support their program or not. Social organizations may ask people’s opinion on current debates. All this information can be obtained from microblogging services, as their users post everyday what they like/dislike, and their opinions on many aspects of their life.

In our paper, we study how microblogging can be used for sentiment analysis purposes. We show how to use Twitter as a corpus for sentiment analysis and opinion mining. We use microblogging and more particularly Twitter for the following reasons:

- Microblogging platforms are used by different people to express their opinion about different topics, thus it is a 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, politicians4, 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 countries5. Although users from U.S. are prevailing, it is possible to collect data in different languages.

We collected a corpus of 300000 text posts from Twitter evenly split automatically between three sets of texts:

1. texts containing positive emotions, such as happiness, amusement or joy
2. texts containing negative emotions, such as sadness, anger or disappointment
3. objective texts that only state a fact or do not express any emotions

We perform a linguistic analysis of our corpus and we show how to build a sentiment classifier that uses the collected corpus as training data.

---

1http://twitter.com
2http://tumblr.com
3http://facebook.com
4http://www.sysomos.com/insidetwitter/politics
5http://www.sysomos.com/insidetwitter/#countries
The contributions of our paper are as follows:

1. Contributions

We present a method to collect a corpus with positive and negative sentiments, and a corpus of objective texts. Our method allows to collect negative and positive sentiments such that no human effort is needed for classifying the documents. Objective texts are also collected automatically. The size of the collected corpora can be arbitrarily large.

2. We perform statistical linguistic analysis of the collected corpus.

3. We use the collected corpora to build a sentiment classification system for microblogging.

4. We conduct experimental evaluations on a set of real microblogging posts to prove that our presented technique is efficient and performs better than previously proposed methods.

1.1. Contributions

The contributions of our paper are as follows:

1. We present a method to collect a corpus with positive and negative sentiments, and a corpus of objective texts. Our method allows to collect negative and positive sentiments such that no human effort is needed for classifying the documents. Objective texts are also collected automatically. The size of the collected corpora can be arbitrarily large.

2. We perform statistical linguistic analysis of the collected corpus.

3. We use the collected corpora to build a sentiment classification system for microblogging.

4. We conduct experimental evaluations on a set of real microblogging posts to prove that our presented technique is efficient and performs better than previously proposed methods.

1.2. Organizations

The rest of the paper is organized as follows. In Section 2, we discuss prior works on opinion mining and sentiment analysis and their application for blogging and microblogging. In Section 3, we describe the process of collecting the corpora. We describe the linguistic analysis of the obtained corpus in Section 4 and show how to train a sentiment classifier and our experimental evaluations in Section 5. Finally, we conclude about our work in Section 6.

2. Related work

With the population of blogs and social networks, opinion mining and sentiment analysis became a field of interest for many researches. A very broad overview of the existing work was presented in (Pang and Lee, 2008). In their survey, the authors describe existing techniques and approaches for an opinion-oriented information retrieval. However, not many researches in opinion mining considered blogs and even much less addressed microblogging. In (Yang et al., 2007), the authors use web-blogs to construct a corpora for sentiment analysis and use emotion icons assigned to blog posts as indicators of users’ mood. The authors applied SVM and CRF learners to classify sentiments at the sentence level and then investigated several strategies to determine the overall sentiment of the document. As the result, the winning strategy is defined by considering the sentiment of the last sentence of the document as the sentiment at the document level.

J. Read in (Read, 2005) used emoticons such as “:-)” and “:-(" to form a training set for the sentiment classification. For this purpose, the author collected texts containing emoticons from Usenet newsgroups. The dataset was divided into “positive” (texts with happy emoticons) and “negative” (texts with sad or angry emoticons) samples. Emoticon-trained classifiers: SVM and Naïve Bayes, were able to obtain up to 70% of an accuracy on the test set.

In (Go et al., 2009), authors used Twitter to collect training data and then to perform a sentiment search. The approach is similar to (Read, 2005). The authors construct a corpora by using emoticons to obtain “positive” and “negative” samples, and then use various classifiers. The best result was obtained by the Naïve Bayes classifier with a mutual information measure for feature selection. The authors were able to obtain up to 81% of accuracy on their test set. However, the method showed a bad performance with three classes (“negative”, “positive” and “neutral”).

3. Corpus collection

Using Twitter API we collected a corpus of text posts and formed a dataset of three classes: positive sentiments, negative sentiments, and a set of objective texts (no sentiments). To collect negative and positive sentiments, we followed the same procedure as in (Read, 2005; Go et al., 2009). We queried Twitter for two types of emoticons:

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

The two types of collected corpora will be used to train a classifier to recognize positive and negative sentiments.

In order to collect a corpus of objective posts, we retrieved text messages from Twitter accounts of popular newspapers and magazines, such as “New York Times”, “Washington Posts” etc. We queried accounts of 44 newspapers to collect a training set of objective texts. Because each message cannot exceed 140 characters by the rules of the microblogging platform, it is usually composed of a single sentence. Therefore, we assume that an emotion within a message represents an emotion for the whole message and all the words of the message are related to this emotion. In our research, we use English language. However, our method can be adapted easily to other languages since Twitter API allows to specify the language of the retrieved posts.

| funkeybrewster: @redeyechicago I think Obama’s visit might’ve sealed the victory for Chicago. Hopefully the games mean good things for the city. |
| vcurve: I like how Google celebrates little things like this: Google.co.jp honors Confucius Birthday — Japan Probe |
| mattfellows: Hai world. I hate faulty hardware on remote systems where politics prevents you from moving software to less faulty systems. |
| brrroooklyn: I love the sound my iPod makes when I shake to shuffle it. Boo bee boo |

Table 1: Examples of Twitter posts with expressed users’ opinions
In the graph, we see that superlative adjectives (JJS) are used more often for expressing emotions and opinions, and comparative adjectives (JJR) are used for stating facts and providing information. Adverbs (RB) are mostly used in subjective texts to give an emotional color to a verb. Figure 3 shows values of $P_T$ for negative and positive sets. As we see from the graph, a positive set has a prevailing number of possessive wh-pronoun 'whose' (WH$), which is unexpected. However, if we look in the corpus, we discover that Twitter users tend to use 'whose' as a slang version of 'who is'. For example:

    dinner & jack o’lantern spectacular tonight! :)  whose ready for some pumpkins??

Another indicator of a positive text is superlative adverbs (RBS), such as “most” and “best”. Positive texts are also characterized by the use of possessive ending (POS).

As opposite to the positive set, the negative set contains more often verbs in the past tense (VBN, VBD), because many authors express their negative sentiments about their loss or disappointment. Here is an example of the most frequent verbs: “missed”, “bored”, “gone”, “lost”, “stuck”, “taken”.

We have compared distributions of POS-tags in two parts of the same sets (e.g. a half of the positive set with another half of the positive set). The proximity of the obtained distributions allows us to conclude on the homogeneity of the corpus.

5. Training the classifier

5.1. Feature extraction

The collected dataset is used to extract features that will be used to train our sentiment classifier. We used the presence of an n-gram as a binary feature, while for general information retrieval purposes, the frequency of a keyword’s occurrence is a more suitable feature, since the overall sentiment may not necessarily be indicated through the repeated use of keywords. Pang et al. have obtained better results by using a term presence rather than its frequency (Pang et al., 2002).

We have experimented with unigrams, bigrams, and trigrams. Pang et al. (Pang et al., 2002) reported that unigrams outperform bigrams when performing the sentiment classification of movie reviews, and Dave et al. (Dave et al., 2003) have obtained contrary results: bigrams and trigrams worked better for the product-review polarity classification. We tried to determine the best settings for the microblogging data. On one hand high-order n-grams, such as trigrams, should better capture patterns of sentiments expressions. On the other hand, unigrams should provide a good coverage of the data. The process of obtaining n-grams from a Twitter post is as follows:

1. Filtering – we remove URL links (e.g. http://example.com), Twitter user names (e.g. @alex – with symbol @ indicating a user name), Twitter special words (such as “RT”\textsuperscript{6}), and emoticons.

- An abbreviation for retweet, which means citation or reposting of a message.
2. Tokenization – we segment text by splitting it by spaces and punctuation marks, and form a bag of words. However, we make sure that short forms such as “don’t”, “I’ll”, “she’d” will remain as one word.

3. Removing stopwords – we remove articles (“a”, “an”, “the”) from the bag of words.

4. Constructing n-grams – we make a set of n-grams out of consecutive words. A negation (such as “no” and “not”) is attached to a word which precedes it or follows it. For example, a sentence “I do not like fish” will form two bigrams: “I do+not”, “do+not like”, “not+like fish”. Such a procedure allows to improve the accuracy of the classification since the negation plays a special role in an opinion and sentiment expression (Wilson et al., 2005).

5.2. Classifier

We build a sentiment classifier using the multinomial Naïve Bayes classifier. We also tried SVM (Alpaydin, 2004) and CRF (Lafferty et al., 2001), however the Naïve Bayes classifier yielded the best results. Naïve Bayes classifier is based on Bayes’ theorem (Anthony J, 2007).

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

where \( s \) is a sentiment, \( M \) is a Twitter message. Because, we have equal sets of positive, negative and neutral messages, we simplify the equation:

\[
P(s|M) = \frac{P(M|s)}{P(M)}
\]

\[
P(s|M) \sim P(M|s)
\]

We train two Bayes classifiers, which use different features: presence of n-grams and part-of-speech distribution information. N-gram based classifier uses the presence of an n-gram in the post as a binary feature. The classifier based
on POS distribution estimates probability of POS-tags presence within different sets of texts and uses it to calculate posterior probability. Although, POS is dependent on the n-grams, we make an assumption of conditional independence of n-gram features and POS information for the calculation simplicity:

$$P(s|M) \sim P(G|s) \cdot P(T|S)$$  \hfill (5)

where $G$ is a set of n-grams representing the message, $T$ is a set of POS-tags of the message. We assume that n-grams are conditionally independent:

$$P(G|s) = \prod_{g \in G} P(g|s)$$  \hfill (6)

Similarly, we assume that POS-tags are conditionally independent:

$$P(T|s) = \prod_{t \in G} P(t|s)$$  \hfill (7)

$$P(s|M) \sim \prod_{g \in G} P(g|s) \cdot \prod_{t \in G} P(t|s)$$  \hfill (8)

Finally, we calculate log-likelihood of each sentiment:

$$L(s|M) = \sum_{g \in G} \log(P(g|s)) + \sum_{t \in G} \log(P(t|s))$$  \hfill (9)

### 5.3. Increasing accuracy

To increase the accuracy of the classification, we should discard common n-grams, i.e. n-grams that do not strongly indicate any sentiment nor indicate objectivity of a sentence. Such n-grams appear evenly in all datasets. To discriminate common n-grams, we introduced two strategies. The first strategy is based on computing the entropy of a probability distribution of the appearance of an n-gram in different datasets (different sentiments). According to the formula of Shannon entropy(Shannon and Weaver, 1963):

$$entropy(g) = H(p(S|g)) = -\sum_{i=1}^{N} p(S_i|g) \log p(S_i|g)$$  \hfill (10)

where $N$ is the number of sentiments (in our research, $N = 3$). The high value of the entropy indicates that a distribution of the appearance of an n-gram in different sentiment datasets is close to uniform. Therefore, such an n-gram does not contribute much in the classification. A low value of the entropy on the contrary indicates that an n-gram appears in some of sentiment datasets more often than in others and therefore can highlight a sentiment (or objectivity). Thus, to increase the accuracy of the sentiment classification, we would like to use only n-grams with low entropy values. We can control the accuracy by putting a threshold value $\theta$, filtering out n-grams with entropy above $\theta$. This would lower the recall, since we reduce the number of used features. However our concern is focused on high accuracy, because the size of the microblogging data is very large.

For the second strategy, we introduced a term “salience” which is calculated for each n-gram:

$$salience(g) = \frac{1}{N} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} 1 - \frac{\min(P(g|s_i), P(g|s_j))}{\max(P(g|s_i), P(g|s_j))}$$  \hfill (11)

### Table 2: N-grams with high values of salience (left) and low values of entropy (right)

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

The introduced measure takes a value between 0 and 1. The low value indicates a low salience of the n-gram, and such an n-gram should be discriminated. Same as with the entropy, we can control the performance of the system by tuning the threshold value $\theta$.

In Table 5.3, examples of n-grams with low entropy values and high salience values are presented. Using the entropy and salience, we obtain the final equation of a sentiment’s log-likelihood:

$$L(s|M) = \sum_{g \in G} \log(P(g|s)) \cdot \mathbb{I}(f(g) > \theta, 1, 0)$$  \hfill (12)

where $f(g)$ is the entropy or the salience of an n-gram, and $\theta$ is a threshold value.

### 5.4. Data and methodology

We have tested our classifier on a set of real Twitter posts hand-annotated. We used the same evaluation set as in (Go et al., 2009). The characteristics of the dataset are presented in Table 5.4.

<table>
<thead>
<tr>
<th>Sentiment</th>
<th>Number of samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>108</td>
</tr>
<tr>
<td>Negative</td>
<td>75</td>
</tr>
<tr>
<td>Neutral</td>
<td>33</td>
</tr>
<tr>
<td>Total</td>
<td>216</td>
</tr>
</tbody>
</table>

Table 3: The characteristics of the evaluation dataset

We compute accuracy (Manning and Schütze, 1999) of the classifier on the whole evaluation dataset, i.e.:

$$accuracy = \frac{N(correct\ classifications)}{N(all\ classifications)}$$  \hfill (13)

We measure the accuracy across the classifier’s decision (Adda et al., 1998):

$$decision = \frac{N(retrieved\ documents)}{N(all\ documents)}$$  \hfill (14)
The value of the decision shows what part of data was classified by the system.

5.5. Results
First, we have tested the impact of an n-gram order on the classifier’s performance. The results of this comparison are presented in Figure 4. As we see from the graph, the best performance is achieved when using bigrams. We explain it as bigrams provide a good balance between a coverage (unigrams) and an ability to capture the sentiment expression patterns (trigrams).

Next, we examine the impact of attaching negation words when forming n-grams. The results are presented in Figure 5.

From the both figures, we see that we can obtain a very high accuracy, although with a low decision value (14). Thus, if we use our classifier for the sentiment search engine, the outputted results will be very accurate.

We have also examined the impact of the dataset size on the performance of the system. To measure the performance, we use $F$-measure (Manning and Schütze, 1999):

$$F = \left(1 + \beta^2\right) \frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{recall} + \text{recall}}$$  \hspace{1cm} (15)

In our evaluations, we replace precision with accuracy (13) and recall with decision (14), because we deal with multiple
classes rather than binary classification:

\[ F = \frac{1 + \beta^2 \cdot \text{accuracy} \cdot \text{decision}}{\beta^2 \cdot \text{accuracy} + \text{decision}} \]  

(16)

where \( \beta = 0.5 \) We do not use any filtering of n-grams in this experiment. The result is presented on Figure 6. As we see from the graph, by increasing the sample size, we improve the performance of the system. However, at a certain point when the dataset is large enough, the improvement may not be achieved by only increasing the size of the training data.

We examined two strategies of filtering out the common n-grams: salience (11) and entropy (10). Figure 7 shows that using the salience provides a better accuracy, therefore the salience discriminates common n-grams better then the entropy.

6. Conclusion

Microblogging nowadays became one of the major types of the communication. A recent research has identified it as online word-of-mouth branding (Jansen et al., 2009). The large amount of information contained in microblogging web-sites makes them an attractive source of data for opinion mining and sentiment analysis.

In our research, we have presented a method for an automatic collection of a corpus that can be used to train a sentiment classifier. We used TreeTagger for POS-tagging and observed the difference in distributions among positive, negative and neutral sets. From the observations we conclude that authors use syntactic structures to describe emotions or state facts. Some POS-tags may be strong indicators of emotional text.

We used the collected corpus to train a sentiment classifier. Our classifier is able to determine positive, negative and neutral sentiments of documents. The classifier is based on the multinomial Naı¨ ve Bayes classifier that uses N-gram and POS-tags as features.

As the future work, we plan to collect a multilingual corpus of Twitter data and compare the characteristics of the corpus across different languages. We plan to use the obtained data to build a multilingual sentiment classifier.

7. References


