# Twitter Topic Modeling and Sentiment Analysis[1]

**Cristina Cano**
University of Southern California
Los Angeles, CA 90089
cristinc@usc.edu

**Sayat Satybaldiyev**
University of Southern California
Los Angeles, CA 90089
satybald@usc.edu

## Abstract

In this paper we discuss a two-part system for sentiment analysis and topic modeling of tweets. In our system, tweets are classified as either "positive", "negative", or "neutral" sentiment-bearing. The tweets are then sorted into document groups based on the search term used to retrieve the tweet and sent to an LDA topic modeling subsystem to extract words/topics related to the respective search terms.

## 1 Introduction

With the advent of real-time social networks, user-generated data in the form of millions of short blurbs and status updates are generated each day about hundreds of different topics. The task of extracting data from these small texts has become immensely useful for sorting and ranking popularity of topics mentioned within the updates. Furthermore, it is important to not only extract information regarding which topics are most popular (in Twitter terminology, "trending") and how they correlate, but also what types of reactions users have towards these topics. This data is especially useful in the context of marketing for analyzing consumer reactions (via social media) to brands and products. The largest challenge when processing social media updates is their inherently short nature, which makes it difficult to extract very much information about sentiment and topics related to the updates.

We approach the problem with respect to Twitter data, or "tweets', with a two-fold process of sentiment analysis and LDA topic modeling.

## 2 Related Work

Here we discuss existing work done related to our system.

### 2.1 Sentiment Analysis

In previous work, the method of choice for the sentiment analysis task is often supervised learning, requiring an annotated training dataset to teach the system. In the system of Go et al, annotation was done naively based on the presence of either positive smiley emoticons (such as ":)") or negative frowning emoticons (such as ":("). Training on only these associated "positive" and "negative" sentiment tweets ignores the important factor of neutral sentiment tweets. Go's work also strips emoticons from the training data, eliminating a rich source of sentiment information. Go's work also considers the usefulness of different feature sets, including unigram, bigram, unigram+bigram, and parts of speech tags. Go's results show that the unigram features work very well, with small improvement when bigrams are also used.

Hu and Liu's system for customer review opinion mining uses a sentiment analysis component based on word lists generated for both positive and negative sentiment orientation. We, too, incorporate these sentiment word lists, but do not rely solely on these lists to distinguish sentiment. As discussed in Liu's 2010 work, word lists of sentiment orientation are useful but insufficient on their own to determine overall text sentiment, partially as this approach also ignores the influence of "neutral" sentiment.

In our work we also use supervised learning algorithms to conduct our sentiment classification task using a fully human-annotated training dataset. With the annotated dataset, we aim to have a finer-grained judgment of sentiment including information distinguishing neutral tweets. We have also opted for unigram features,

[1] This paper has been submitted as a final project for the course CSCI-544 Applied Natural Language Processing at University of Southern California. This course was taught by Dr. Zornitsa Kozareva during the Spring 2013 semester.

feeling that the addition of bigrams added only very sparse and noisy data for our relatively small corpus of very short and irregular texts (tweets).

### 2.2 Topic Modeling

In Ramage et al's 2010 paper, they present a work on extracting and labeling topics from Twitter using a Labeled LDA algorithm. They use several latent dimensions for categorizing tweets as related to style, sociability, status and substance. Based on the content of the user's tweets they build a system that suggests which users to follow and personalize the feed.

Wayne et. al., 2011 introduced a new unsupervised Twitter-LDA model to discover topics from a representative sample of the entire Twitter dataset. They used the New York Times topic dataset for categorizing Twitter messages. As a result, a new Twitter-LDA was made for labeling short tweets.

As a part of the topic modeling task we try to use an alternative approach of representing Twitter content. Based on this approach, we are using LDA (Blei, et al., 2003) for the unsupervised topic modeling task. It discovers latent structure of topics over the documents, calculating probabilistic prior of words in a three-level hierarchical Bayesian model (Blei, et al., 2003). Each topic is modeled as infinite mixture over an underlying set of topic probabilities (Blei, et al., 2003). In the end, a set of topics represents the context of the corpus.

## 3 Data Collection & Annotation

We used an existing human-annotated sentiment dataset of tweets from Sanders Analytics (http://www.sananalytics.com/lab/twitter-sentiment/). This dataset includes tweets extracted by one of the following four topic keywords: @apple, #google, #microsoft, #twitter. Because the existing dataset also includes many non-English tweets, we first removed those tweets to obtain our final dataset for training. Additionally, all of the tweet text was sanitized by removing any URL links and replacing them with the string "URL". Note that all remaining entities in the tweet were kept intact, including all usernames indicated by "@" and all tagging text denoted by "#". Table 1 shows the distribution of tweets in the resultant dataset.

| | **Pos** | **Neg** | **Neu** | **Irrel.** | **Total** |
|---|---|---|---|---|---|
| **Apple** | 148 | 307 | 470 | 33 | 958 |
| **Google** | 194 | 50 | 536 | 114 | 894 |
| **Microsoft** | 82 | 124 | 539 | 34 | 779 |
| **Twitter** | 54 | 59 | 513 | 39 | 665 |
| **Total** | 478 | 540 | 2058 | 220 | 3296 |

Table 1: Tweet distribution

As shown in the table, our initial training dataset is annotated according to one of four classes: positive, negative, neutral, irrelevant. The tweets are annotated according to the following table. In our system, we group the "irrelevant" classification with the "neutral" classification into the class "other" to focus more on the distinctions between positive, negative and neutral/other sentiments.

| | |
|---|---|
| **Positive** | • positive indicator on topic |
| **Neutral** | • neither positive nor negative indicators<br>• mixed positive and negative indicators<br>• on topic, but indicator undeterminable<br>• simple factual statements<br>• questions with no strong emotions indicated |
| **Negative** | • negative indicator on topic |
| **Irrelevant** | • not English language<br>• not on-topic (e.g. spam) |

Table 2: Annotation guideline

For testing, we extracted a small dataset using the Twitter Streaming API. The dataset comprised of 100 unique (not re-tweeted) English tweets for each of 4 stream filtering terms ("apple", "microsoft", "google", "amazon"). As in the training dataset, tweets were sanitized by replacing URL links with the text "URL".

## 4 Method Description

In the sentiment analysis task, we used the following features to train and test our system.

- numPosEmots – the number of positive emoticons in the tweet
- numNegEmots – the number of negative emoticons in the tweet
- numExclam – the number of exclamation marks ("!") in the tweet

- numQuest – the number of questions marks (“?”) in the tweet
- numPosGaz – the number of positive gazetteer words in the tweet
- numNegGaz – the number of negative gazetteer words in the tweet
- unigrams – the TF-IDF values for unigrams not in the stopwords list (extended from the Python NLTK stopwords list) and which appear in the total corpus more than once; a total of 3228 such unigrams were extracted

The positive and negative word gazetteers used were the ones generated in the work by Hu and Liu, and the positive and negative emoticon gazetteers were manually created with some guidance from Wikipedia’s Western-style emoticon list. Note that our use of emoticons does not include Unicode smiley-characters. Also note that the unigrams used in training and testing the system are from the initial training corpus, offering only a limited collection of possibly relevant unigrams.

With the above feature set, we trained our system using 10-fold cross validation with several supervised learning algorithms in Weka. The results of this training will be discussed in the next section.

In the topic modeling task, we especially consider the challenges of the short tweets. Because tweets are short and contain net-lingo abbreviations, the data is inherently noisy. We apply LDA analysis for the group of tweets for each particular search term. For example, our test corpus contains four search terms, so we treat this as four individual documents separated by search term.

For the topic modeling task we use the MALLET software package, which has a stable API and a solid research community.

For LDA we used our first dataset to run an initial experiment. We set up 4 topics in MALLET. To run an experiment in MALLET we convert our corpus to an internal serialization file. With --output-topic-keys MALLET generated our topics. We used the --word- topic-counts-file argument to generate how many words in the document related to the topics. Also, to generate distribution topics over particular words in each document --xml-topic-phrase-report key is used.

For the topic modeling task, the following configurations/features were used:

- stopwords to eliminate unnecessary words
- define the number of topics (how many topics we are going to extract from the corpus)
- number of iterations - how many iterations of Gibs sampling the algorithm will do
- alpha parameter which we use for smoothing over topic distribution
- beta which we use for smoothing over unigram distribution

## 5 Results

The tables below summarize the results of 10-fold cross validation training, given the Precision (P), Recall (R), and F-Measure (F) values.

| | **P** | **R** | **F** |
|---|---|---|---|
| **Positive** | 0.679 | 0.381 | 0.488 |
| **Negative** | 0.739 | 0.52 | 0.611 |
| **Other** | 0.805 | 0.935 | 0.865 |
| **Average** | 0.776 | 0.787 | 0.769 |

Table 3: DMNBtext (5 iterations) results

| | **P** | **R** | **F** |
|---|---|---|---|
| **Positive** | 0.543 | 0.475 | 0.507 |
| **Negative** | 0.614 | 0.537 | 0.573 |
| **Other** | 0.82 | 0.866 | 0.842 |
| **Average** | 0.746 | 0.755 | 0.75 |

Table 4: SMO results

| | **P** | **R** | **F** |
|---|---|---|---|
| **Positive** | 0.267 | 0.598 | 0.369 |
| **Negative** | 0.352 | 0.648 | 0.456 |
| **Other** | 0.831 | 0.449 | 0.583 |
| **Average** | 0.671 | 0.503 | 0.531 |

Table 5: NaiveBayes results

| | **P** | **R** | **F** |
|---|---|---|---|
| **Positive** | 0.38 | 0.295 | 0.332 |
| **Negative** | 0.348 | 0.639 | 0.45 |
| **Other** | 0.81 | 0.687 | 0.743 |
| **Average** | 0.672 | 0.622 | 0.636 |

Table 6: BayesNet results

| | **P** | **R** | **F** |
|---|---|---|---|
| **Positive** | 0.416 | 0.372 | 0.393 |
| **Negative** | 0.46 | 0.359 | 0.403 |
| **Other** | 0.771 | 0.827 | 0.798 |
| **Average** | 0.668 | 0.685 | 0.654 |

Table 7: IB1 (1-Nearest Neighbor) results

| | P | R | F |
|---|---|---|---|
| **Positive** | 0.415 | 0.308 | 0.353 |
| **Negative** | 0.544 | 0.289 | 0.377 |
| **Other** | 0.755 | 0.88 | 0.812 |
| **Average** | 0.671 | 0.7 | 0.675 |

Table 8:HyperPipes results

| | P | R | F |
|---|---|---|---|
| **Positive** | 0.597 | 0.186 | 0.284 |
| **Negative** | 0.58 | 0.281 | 0.379 |
| **Other** | 0.743 | 0.941 | 0.831 |
| **Average** | 0.695 | 0.724 | 0.677 |

Table 9: RandomForest results

Our baseline average F-Measure for the training system is 0.565, and we can see that all but the NaiveBayes algorithm manages to surpass the baseline. We also note here the variety of results given by the different algorithms. In general, there seems to be fairly low precision and recall scores for the Positive and Negative classes. In systems where precision is relatively high for these two sentiment classes, we see correspondingly low recall scores (see results for DMNBtext and RandomForest). Overall, the two most balanced and well-performing algorithms are DMNBtext and SMO, with overall F-scores of 0.769 and 0.75 respectively.

Although the results are not shown here, we also ran the system without the "numPosGaz" and "numNegGaz" features. Surprisingly, there was very little difference in overall scoring without these features. This could be due to the very specific domain of tweets used in our training data, as the training tweets are all related to technology brands. As such, we maintain that these features and the sentiment gazetteers are still valuable so that the system will be applicable to a wider variety of tweets. Alternatively, we could generate technology-specific sentiment gazetteers to gain greater improvement with this particular training dataset and other tweets in the consumer technology domain.

Another possible source of noise is the grouping of tweets annotated as "neutral" and "irrelevant". In particular, "irrelevant" tweets may actually be positive or negative sentiment-bearing but deemed "off-topic" by the human annotator. As it is not the task of our system to determine topicality of tweets, the "irrelevant" tweets introduce some noise into the system.

It is important to note here the article by Ogneva, which states that human annotators (from the Amazon Mechanical Turk service) only agree on sentiment classifications about 79% of the time. As such, a raw accuracy score of about 70% is considered a statistically well-performing system for the sentiment analysis task.

For the topic modeling subsystem, we can see from the tables below that LDA yields a good representation of our documents. It does give some meaningless words, such as "url", "lol", "rt", because these words have not been eliminated from the content and occur frequently.

| Topic | Weight | Count | Words |
|---|---|---|---|
| 1 | 0.18 | 772 | twitter |
| 1 | 0.04 | 160 | rt |
| 1 | 0.02 | 77 | facebook |
| 1 | 0.01 | 64 | url |
| 1 | 0.01 | 55 | lol |
| 1 | 0.01 | 41 | follow |
| 1 | 0.01 | 38 | followers |
| 2 | 0.08 | 559 | url |
| 2 | 0.02 | 141 | phone |
| 2 | 0.01 | 78 | great |
| 2 | 0.01 | 65 | time |
| 2 | 0.01 | 63 | love |
| 2 | 0.01 | 51 | day |
| 3 | 0.19 | 1113 | apple |
| 3 | 0.05 | 291 | iphone |
| 3 | 0.03 | 150 | ios |
| 3 | 0.02 | 122 | siri |
| 3 | 0.02 | 96 | rt |
| 4 | 0.16 | 907 | microsoft |
| 4 | 0.08 | 430 | url |
| 4 | 0.02 | 125 | windows |
| 4 | 0.02 | 112 | rt |
| 4 | 0.02 | 100 | ballmer |

Table 10: Distribution of top 4 topics with their weight

| Term | apple | google | microsoft | twitter |
|---|---|---|---|---|
| **Document** | 1 | 2 | 3 | 4 |
| **Topic 1** | 3 | 0 | 4 | 1 |
| **Weight 1** | 0.62 | 0.73 | 0.69 | 0.78 |
| **Topic 2** | 2 | 2 | 2 | 2 |
| **Weight 2** | 0.29 | 0.18 | 0.23 | 0.17 |
| **Topic 3** | 1 | 1 | 0 | 0 |
| **Weight 3** | 0.05 | 0.03 | 0.04 | 0.02 |
| **Topic 4** | 0 | 4 | 1.00 | 4 |

| | | | | |
|---|---|---|---|---|
| **Weight 4** | 0.02 | 0.02 | 0.01 | 0.01 |

Table 11: Distribution of topics in the documents

## 6 Conclusions

Overall, we found that LDA is an informative tool for representing Twitter. Combining the topic modeling with sentiment analysis yields an impressive overall picture of the content. However, given the short and noisy nature of the Twitter text, extensive preprocessing techniques are required for acceptable data, and grouping content by particular search term gives a better overall representation of the data at hand.

## References


Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. "Sentiment Analysis of Twitter Data". http://www1.ccls.columbia.edu/~beck/pubs/lsm2011_full.pdf.

Alec Go, Richa Bhayani, and Lei Huang. "Twitter Sentiment Classification using Distant Supervision". http://cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf.

DM Blei, AY Ng, MI Jordan. 2003. "Latent dirichlet allocation." *Journal of Machine Learning Research*.

Minqing Hu and Bing Liu. 2004. "Mining and summarizing customer reviews". *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Seattle, Washington.

"List of emoticons." Wikipedia. http://en.wikipedia.org/wiki/List_of_emoticons. Retrieved 2013-04-15.

Bing Liu. 2010. "Sentiment Analysis and Subjectivity." *Handbook of Natural Language Processing*.

Maria Ogneva. 2010. "How Companies Can Use Sentiment Analysis to Improve Their Business". http://mashable.com/2010/04/19/sentiment-analysis/. Retrieved 2013-04-22.

Bo Pang and Lillian Lee. 2008. "Opinion mining and sentiment analysis." *Foundations and trends in information retrieval*, 2(1-2):1-135.

Daniel Ramage, Susan Dumais and Dan Liebling. 2010. "Characterizing Microblogs with Topic Models". *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media.*

Wayne Zhao, et al. 2011. "Comparing Twitter and Traditional Media Using Topic Models." *Advances in Information Retrieval.*