Sentiment Analysis of Occupy Wall Street Tweets

Robert Chang, Sam Pimentel, Alexandr Svistunov

Acknowledgements

Richard Socher, Andrew Maas, and Maren Pearson.

I. Introduction

THE rise of social media has changed political discourse around the world by extending the potential reach of otherwise marginal voices and creating opportunities for massive group coordination and action. The "Arab Spring" protest movements of the past year demonstrate the political power of these tools. A survey by a news organization in the region reported that the "vast majority of . .. people surveyed over three weeks in March said they were getting their information from social media sites (88 per cent in Egypt and 94 per cent in Tunisia)" [5].

As the Twitter stream becomes richer in information and more significant in political impact, the value of monitoring and understanding the stream increases. The United States' Department of Homeland Security, for example, recently announced its intention to develop a system for collecting intelligence information from Twitter and Facebook [2]. The immense quantity of the data available makes identifying key trends non-trivial, however, as Department of Homeland Security Undersecretary Carolyn Wagner comments: "We're still trying to figure out how you use things like Twitter as a source . . .How do you establish trends and how do you then capture that in an intelligence product?" Human observers, inundated by thousands of tweets, need computational tools to aid them in analysis of social media data. While application programming interfaces allow easy automated collection of large social media datasets, analysis of these datasets remains difficult. Simple keyword searches can help identify topics discussed in social media posts, but much of the value of these data lies in the posters' opinions about these topics, encoded in their messages but difficult to extract computationally.

Fortunately, machine learning offers a potential solution through the field of sentiment analysis. A sentiment analysis algorithm "seeks to identify the viewpoint(s) underlying a text span" [8] by extracting descriptive features from text fragments and using them as inputs to a learned hypothesis function. Such algorithms have already been used to classify opinions on current events as expressed in news sources [4]. Asur and Huberman applied sentiment analysis to Twitter data to forecast box-office revenue for movies with competitive accuracy [1]. By training such an algorithm to recognize specific political sentiments of interest rather than opinions about movies, an observer could "predict the future" of relevant political movements much as Asur and Huberman predicted market behavior [1].

In this investigation we apply machine learning methods to analysis of political sentiment in social media. We concentrate on a specific political phenomenon, the Occupy Wall Street movement, and a particular social media platform, Twitter.com. Furthermore, we restrict ourselves to classifying postings into three categories: pro-Occupy-Wall-Street, anti-Occupy-Wall-Street, and neutral or unrelated. We apply a series of machine learning techniques to these data.

II. Methodology

A. Data Collection

To gain access to tweets associated with the OWS movement, we leveraged the Python scripting language and its existing *Python-Twitter* API library. From early November to early December, we collected the daily most recent 1500 tweets (which is roughly the rate limit per call) on *search.twitter.com*. We stored information such as screen names, geo-locations, and texts were stored on the *Redis* server. Before any pre-cleaning, our sample size was a little over $20,000$ tweets.

B. Assigning Responses to Observations

Given the difficulty of assigning sentiment values to text computationally, we relied on a consensus vote by several human judges to determine response values for the observations. We used the Amazon Mechanical Turk platform to collect these votes from users over the Internet, who completed questionnaires requiring them to classify our observed tweets as pro-Occupy-Wall-Street, anti-Occupy-Wall-street, or neutral/irrelevant in exchange for small monetary rewards. To help control for random guessing, we randomized the order of the response buttons on the online forms. Each tweet was offered to five independent judges, and the mode of the resulting votes was taken as the consensus value. When a set of votes had multiple modes (i.e. when a 2-2-1 split occurred among the votes) we concluded that the tweet was likely to have ambiguous sentiment and classified it as neutral.

Unfortunately we obtained low-quality results from Amazon Mechanical Turk. Even with a five-vote consensus system, we discovered many egregious misclassifications while manually checking a small sample of our results. In addition, we discovered by examining the vote breakdown that few of the tweets had unanimous or even strong majority voting results. Figure 1 shows a histogram of the percent of voters agreeing with the assigned consensus label for each tweet (e.g. a 60% figure means 3 of the 5 voters agreed on the label) for a batch of 1162 tweets. Clearly only a tiny fraction of the labels received a unanimous vote, and less than 15% received four or more agreeing votes out of five. In contrast, 11% of the tweets received only one vote for the top label, a case that occurred only when the vote distribution had multiple modes. Although some of the

tweets in our data set were genuinely ambiguous, this excessive level of disagreement suggested to us that an influential subset of the Amazon Mechanical Turk workers who had voted had done so either negligently or with insufficient basic background knowledge of Occupy Wall Street and/or of English to distinguish sentiment on the issue.

In response to these results, we chose to discard the Amazon Mechanical Turk labels for all our tweets and assign new, accurate ones ourselves. Although this approach limited the size of our dataset to the number of tweets we could assign labels to, we already faced similar limits when using Amazon Mechanical Turk due to constraints on the financial resources needed to pay workers. In addition, our self-labeled dataset contained much more trustworthy labels. The resulting set of labels for 1980 tweets was 23% positive, 24% negative, and 53% neutral or irrelevant.

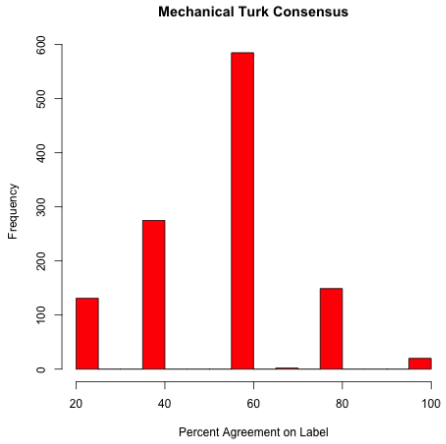

Fig. 1. Histogram showing percent agreement with consensus label for each tweet among Amazon Mechanical Turk voters.

C. Data Cleaning

As we scrutinized the raw data more closely, we realized how diverse and noisy the data were, which meant additional data filtering was required. This was particularly true because we wished to focus on tweets expressing a strong opinion about OWS and not just containing the search query "Occupy Wall Street". Even longer tweets expressing opinions that mentioned OWS were not all relevant. For example, during final exam week in December, we came across many tweets in which students complained about having difficulties in finishing papers on OWS, rather than having direct opinions on the movement itself.

In addition to removing non-English and redundant retweets, we made the major decision to remove OWS related news tweets. Arguably, news organizations are sometimes biased when reporting (e.g. Fox news?), but many news tweets simply update current events rather than reflecting opinions of its own or the general public. With this rather aggresive filtering, we are left with only 10%($\approx$ 2000 tweets) of the original tweet, but they tend to be relevant and carry high sentiments.

D. Feature Creation

In the next step, we used regular expression to remove non-expressive characters but kept emoticons such as !, ?, :), :(and twitter special characters such as hashtag # and direct messaging @. Furthermore, we removed all URLs in the tweets, although the presence of URLs could be later encoded as an indicator variable. We then tokenized the string data, removed the stop words, and applied Porter Stemming using Python's $NLTK$ library.

In the next step, we performed frequency analysis on single tokens, bi-grams, and $n-$grams to identify popular words and remove rare or short tokens (< 2 appearances and < 2 characters long). Beyond simple freqency analysis, we also computed common subsets of tokens across sentiment category groups. This information was useful since if a high frequency token only appeared in one sentiment category, it indicated that this token was important for identifying sentiments. On the other hand, if a frequent token appeared across all different sentiment group, then its widespread appearances were less helpful and could be discounted. This intuition is related to the rationale behind δ TF-IDF, which tries to identify important tokens based on information on the response labels.

E. Term Document Matrix

Finally, we created the Term Document Matrix and applied the TF-IDF transformation. In particular, we created two versions of the document matrix: one that took frequency into account while the other kept track of presence only. Pang *et al.* [8] found out that, unlike topic categorization, repeated use of certain keywords usually does not indicate or help to identify strong sentiments. Finally, some tweets contained no features at all after parsing (e.g. tweets such as "Occupy Wall Street!" that only contained search query), and these were also removed.

III. "Static" Classifier

The static classifier was implemented to serve as a benchmark for other classifiers, since it was expected that this type of algorithm would perform badly for the type of data each tweet contains.

The static classifier was implemented using the AFINN-111 dictionary of valence-rated English words developed specifically for microblogs ([7]). The dictionary contains 2477 words (some of the words are grammatically different versions of the same word, for instance "favorite" and "favorites" are two different words). The valence is an integer value from -5 to 5 (with 5 being the most positive connotation, -5 the most negative one).

Each tweet was then assigned a score as follows. For each tweet in the sample, its score was the sum of the scores of the words in it. If a word did not appear in the dictionary, its score was taken to be 0.

The next part was to classify tweets according to the score they obtained. While classification into two groups would be straightforward (for instance, all non-zero scores would go into the appropriate category, with zero-rated

tweets being assigned randomly to one of the categories), expansion to three categories requires a choice of bounds (u, l) such that a tweet is assigned to a positive category if $score \geq u$, to the neutral if $l \leq score < u$, and to the negative category if $score < l$. The bounds were chosen to maximize total accuracy on the largest balanced sample. The bounds came out to be $u = 8, l = -4$. The overall accuracy of the algorithm for the largest sample size came out to be 0.48875. The sensitivity for the groups was 0.05 for positive tweets, 0.10 for negative ones, and 0.98 for neutral ones. As we can see, not only was the overall accuracy small, but it was also achieved by very good performance on one particular group of tweets (neutral) at the expense of the others.

There are several possible reasons for poor performance of the static classifier. First, the dictionary is fixed and limited in size; any word, no matter how emotionally charged, will go undetected if it does not appear in the dictionary. A related issue is that the dictionary does not pick up Twitter-specific features like hash tags, which are picked up by other algorithms.

Second, the classifier is unable to pick up subtle linguistic constructs such as sarcasm, euphemisms, double entendre, etc.

IV. Naïve Bayes Algorithm

The model we implemented after the data-gathering step is a slightly modified version of the Naïve Bayes model with Laplace smoothing. The modification is due to the fact that we are using three categories instead of two in the response variables.

Let $y_i = 1$ for a positive tweet, $y_i = 0$ for a neutral tweet, and $y_i = -1$ for a negative sentiment, and let $\phi_{j|y=k}$ be the probability of a jth token appearing in a tweet given that the sentiment is k, $k \in \{1, 0, -1\}$. Let n_i be the number of words in a given tweet, and x_i^j the number of times token j appears in tweet i. Then, the Maximum Likelihood estimates of the probabilities become:

$$\phi_{j|y=k} = \frac{\sum_{i=1}^{m} x_i^j \mathbf{1}(x_i^j > 0, y_i = k)}{\sum_{i=1}^{m} n_i \mathbf{1}(y_i = k)}$$

$$\phi_{y=k} = \frac{\sum_{i=1}^{m} \mathbf{1}(y_i = k)}{m}$$

Whenever we want to make a prediction on a new tweet, the probability of the new tweet having sentiment k is as follows:

$$p(y = k|x) = \frac{\prod_{j=1}^{n} p(x^j|y = k)p(y = k)}{\sum_{k=-1}^{1}(\prod_{j=1}^{n} p(x^j|y = k)p(y = k))}$$

The overall leave-one-out cross-validation accuracy of this algorithm when applied to the full data was 0.488.

V. Support Vector Machine

In addition to Naïve Bayes, we fit a support vector machine with a linear kernel to our data using the Liblinear package [3]. In order to handle 3-way classification, Liblinear fits 3 separate support vector machines, one to classify tweets as being in or out of each of our possible label categories. The leave-one-out cross validation accuracy achieved by this model on the full data set was 0.323, slightly worse than random guessing.

VI. Stanford Classifier

The Stanford Classifier is a particular implementation of a Maximum-Entropy classifier for text classification. A maximum-entropy classifier is equivalent to a multiclass logistic regression model [6]. This implementation (written in Java) takes as inputs the response variables, as well as the tweets themselves, in the trainng set, creates a set of features according to the input parameters, and then tests the performance of the model on the test set.

The Stanford Classifier model which used N-grams consisting of letters showed very good performance. The overall accuracy for the largest possible training size was 65%, with the sensitivities for the three groups being 0.508 for positive tweets, 0.493 for negative tweets, and 0.57 for neutral tweets.

The Stanford Classifier achieves its high level of accuracy by using N-grams of letters along with individual words. This expansion of the feature space has drawbacks: while it improves performance, the high-influence features it selects for a large role in prediction are no longer emotionally charged *words* but mostly meaningless text fragments. Nevertheless, for classification purposes, the Stanford classifier is the best algorithm among those we tried, judging by overall accuracy and the sensitivities for different groups.

VII. Results

The two plots show the results of the four classification methods tested. Figure 2 shows test error plotted against training sample size for each of the four algorithms. Note that leave-one-out cross-validation error was used for the static, Naïve Bayes, and support vector machine models but that the Stanford Classifier, which does not have a built-in cross-validation method or an interface conducive to performing cross-validation, was tested against a held-out test, composed in large part of tweets discarded while balancing the data. Figure 3 contains four subplots, each of which shows conditional accuracy in each label category for one of our algorithms, plotted as a function of training sample size. Conditional accuracy is defined as the percentage of the tweets sharing a certain true label assigned to that label by the classification algorithm. As such, it is an adaptation of the concept of sensitivity that is applicable to three-way-classification.

Contrary to expectation, few of the plots show a strong decrease in test error with increasing training sample size. Support vector machine and Stanford Classifier performance do appear to improve slightly with the largest training samples, however, and we conclude that our small sample size prevents us from seeing the larger trend. We note that although support vector machines can often give excellent performance on classification problems, our small

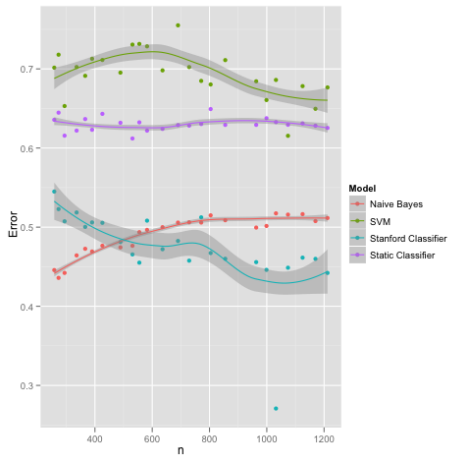

Fig. 2. Test error for the algorithms for different training set sizes

sample size greatly reduces the accuracy of the linear kernel SVM, which has the highest cross-validation error of any algorithm we tested, including the static classifier.

The sensitivity plots demonstrate some of the subtleties of the different models. In particular, the static classifier achieves excellent performance on neutral tweets by classifying almost every tweet as neutral; however, this destroys its conditional accuracy in predicting positive and negative tweets. Assuming we are interested in identifying and distinguishing positive and negative tweets, the static classifier is a terrible choice, worse in fact than the SVM although its cross-validation error is lower. Thus all our machine learning algorithms outperform a static classifier in areas of interest.

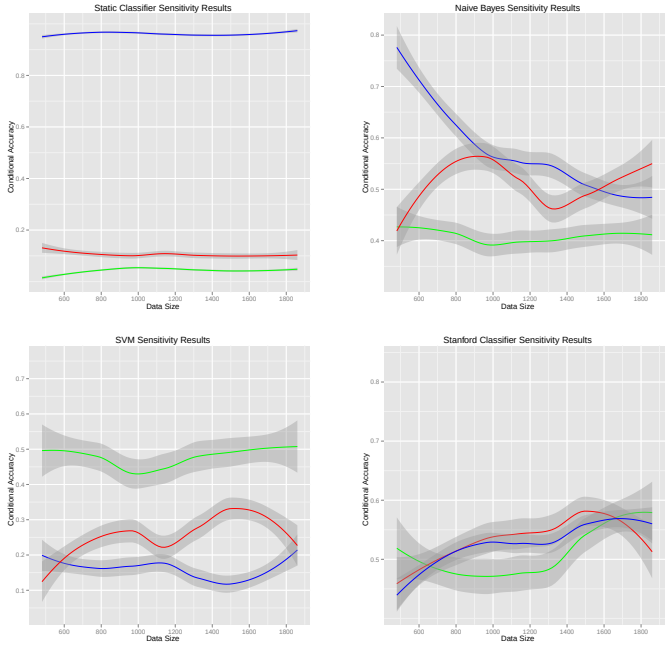

Fig. 3. Test error curves for each of the four algorithms tested, plotted against training sample size. All test errors were computed using leave-one-out cross-validation except for the Stanford Classifier's.

In addition to performance measurements, we obtained lists of "high influence" tokens from our Naïve Bayes and support vector machine models. For Naïve Bayes, we selected the tokens with the highest conditional probabilities of appearing given a tweet was positive and given a tweet was negative, respectively, and for the SVM we extracted the tokens with the highest positive weights for each of the two categories. The results are shown in the table. In the column headings, NB indicates Naïve Bayes, "+" indicates these tweets are associated with pro-OWS tweets and "-" indicates that they are associated with anti-OWS tweets.

NB +	NB -	SVM +	SVM -
movement	don't	banker	nba
#occupywallstreet	movement	demand	strike
video	like	corrupt	union
music	get	greed	riot
tweet	f***	show	move
7xfgk2k	Would	cool	clue
support	I'm	&gt	miley
don't	street?	interview	opinion
get	think	might	camp

Looking at the tabular results, we notice that some of the tokens the Naïve Bayes finds likely to occur in positive tweets are also likely to occur in negative tweets ("don't" and "movement," for example) - this suggests these tokens are simply very frequent and are not necessarily indicators of positive or negative sentiment. In contrast, the tokens with high positive weights for a given category in the SVM do tend to be good discriminators. While the lists contain some tokens that are likely mere artifacts of small sample size (it's unlikely that the token "miley," taken from Twitter references to Miley Cyrus, would be a strong indicator of anti-OWS sentiment in general), others show us interesting things about sentiment expression in Twitter. For example, one of the tokens with the highest pro-OWS association in the Naïve Bayes was the hashtag "#occupywallstreet," suggesting, as we might expect, that hashtags can have strong sentiment value separate from that of the words themselves. In addition, the negative association of the token "street?" points to the importance of punctuation marks in sentiment analysis. While the token "street" occurred in every tweet in our data set as part of the phrase "Occupy Wall Street," the addition of a question mark gives it a strong negative slant.

VIII. Discussion

In this section, we will briefly discuss ideas that we think are worth pursuing to improve model performance.

A. Data Collection

Leveraging URLs of News Tweets: As mentioned earlier, we filtered out news tweets (which are estimated to be about 40% of the tweets we collected) aggresively. One idea to leverage these news tweets is to crawl the actual news article when their URLs are available. This would give us textual information beyond the 140 characters restriction on twitter. Although we did not implement this idea, we did use similar techniques to scrape the titles of

news articles when URLs are available. But we found out that URLs tend to expire quickly, and many titles were highly correlated with the tweet contents themselves.

Target Specific Accounts: An alternative approach to obtain more data set without labeling the sentiments ourselves is to target specific Twitter accounts that are mostly likely to have pro or anti-sentiments. Accounts such as #OccupyTogether, #OWS contains tweets that mostly supports OWS, while others such as #Anti Occupier, #AntiOccupyWs, #AntiOccupier are certainly to be against OWS movement. The caveat, of course, is to be careful when selecting these accounts, since voices of these individuals might not be generalizable.

B. Feature Creation & Selection

Feature Creation: Based on the "bag of words" technique, our features were generated directly from the training set. While this allowed us to build a more domain specific dictionary, the feature set was often sensitive to current events. For example, during the brief time surrounding Miley Cyrus' appearance at the OWS concert, the token "Miley" came as one of the top tokens, but it is unlikely to be a significant sentiment indicator in the long term. One way to produce a more robust dictionary would be to keep track a frequency table for existing tokens, and update the ranking as new tweets arrives.

Feature Selection: With the benefit of a largre data set, we could enlarge the feature space using different strategies and perform regularizations on the model (this is in fact what the Stanford Classifer did: max-entropy classification with regularization on a large feature space). Techniques such as **negation (**attaching "NOT" to tokens after negation terms), part of speech (considering only adjectives and adverbs), and **co-occurence/constrastive distance analysis** (measuring how likely two tokens are to co-appear) could be helpful to replace independence assumptions like those made by the Naïve Bayes algorithm.

C. Re-evaluation of Sentiment Definition

Even among directly relevant tweets expressing strong sentiment, classifying that sentiments can be difficult for humans. For example, at the time when NBA announced the quarter of a billion salary for star players, many commented that Occupy protesters were focusing on the wrong target, and that they should be occupying NBA instead of Wall Street. Are such tweets pro, neutral, or anit OWS? Perhaps a re-evaluation and clarification of the sentiment definition is needed to provide a more consistent approach in labeling such tweets.

IX. Conclusion

In this current project, we have shown that even a basic model such as Naïve Bayes trained on a small data set outperform a non-machine learning model. Although our model performance did not achieve the kind of performance that would be desirable in an industrial application, we have shown that with increasing sample size, the accuracy would reach much better results. We also propose methods for improving data size, quality, and introduce strategies for constructing better feature selection. In addition, machine learning concepts such as similarity measures, clustering, and cross validations were applied in helpful ways throughout the project.

The explosion of social networks allows researchers to gain great insights to social interactions through text based conversation and discourse. In our project we learned to deal with the noisy nature of social data and recognized that natural language processing is an iterative process that requires careful fine tuning. Achieving high predictive accuracy ultimately depends on the design decisions in data collection, data parsing, and feature selection.

References

[1] Sitaram Asur, Bernardo A. Huberman, *Predicting the Future With Social Media* IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2010.

[2] P. Solomon Banda, *Homeland Security reviews social media guidelines*, Associated Press, Oct. 31, 2011.

[3] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin. *LIBLINEAR: A Library for Large Linear Classification.* Journal of Machine Learning Research 9(2008), 1871-1874. Software available at http://www.csie.ntu.edu.tw/ cjlin/liblinear

[4] Namrata Godbole, Manjunath Srinivasaiah, Steven Skiena, *Large-Scale Sentiment Analysis for News and Blogs* International Conference on Weblogs and Social Media, Boulder, CO, 2007.

[5] Carol Huang, *Facebook and Twitter key to Arab Spring uprisings: report*, The National, Abu Dhabi, June 6, 2011. Retrieved November 11, 2011 from http://www.thenational.ae

[6] Christopher Manning, Daniel Klein, Kristina Toutanova, Jenny Finkel, Galen Andrew, Joseph Smarr, Chris Cox, Roger Levy, Rajat Raina, Pi-Chuan Chang, Marie-Catherine de Marneffe, Eric Yeh, Anna Rafferty, and John Bauer, *Stanford Classifier*, The Board of Trustees of Leland Stanford Junior University, 2003-2011. Distributed under GNU General Public License. Retrieved December 10, 2011 from http://nlp.stanford.edu/software/classifier.shtml

[7] F. Å Nielsen, *AFINN-111 (list of valence-rated English words)*, Informatics and Mathematical Modelling, Technical University of Denmark. Retrieved December 10, 2011 from http://www2.imm.dtu.dk/pubdb/p.php?6010

[8] Bo Pang, Lillian Lee, *A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts*, ACL '04 Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, 2004.