NTNU: Domain Semi-Independent Short Message Sentiment Classification

Øyvind Selmer **Mikael Brevik** **Björn Gambäck** **Lars Bungum**
Department of Computer and Information Science
Norwegian University of Science and Technology (NTNU)
Sem Sælands vei 7–9, NO–7491 Trondheim, Norway
{oyvinsel,mikaelbr}@stud.ntnu.no {gamback,larsbun}@idi.ntnu.no

Abstract

The paper describes experiments using grid searches over various combinations of machine learning algorithms, features and preprocessing strategies in order to produce the optimal systems for sentiment classification of microblog messages. The approach is fairly domain independent, as demonstrated by the systems achieving quite competitive results when applied to short text message data, i.e., input they were not originally trained on.

1 Introduction

The informal texts in microblogs such as Twitter and on other social media represent challenges for traditional language processing systems. The posts ("tweets") are limited to 140 characters and often contain misspellings, slang and abbreviations. On the other hand, the posts are often opinionated in nature as a very result of their informal character, which has led Twitter to being a gold mine for sentiment analysis (SA). SA for longer texts, such as movie reviews, has been explored since the 1990s;[1] however, the limited amount of attributes in tweets makes the feature vectors shorter than in documents and the task of analysing them closely related to phrase- and sentence-level SA (Wilson et al., 2005; Yu and Hatzivassiloglou, 2003). Hence there are no guarantees that algorithms that perform well on document-level SA will do as well on tweets. On the other hand, it is possible to exploit some of the special features of the web language, e.g., emoticons and emotionally loaded abbreviations. Thus the data will normally go through some preprocessing before any classification is attempted, e.g., by filtering out Twitter specific symbols and functions, in particular retweets (reposting another user's tweet), mentions ('@', tags used to mention another user), hashtags ('#', used to tag a tweet to a certain topic), emoticons, and URLs (linking to an external resource, e.g., a news article or a photo). The first system to really use Twitter as a corpus was created as a student course project at Stanford (Go et al., 2009). Pak and Paroubek (2010) experimented with sentiment classification of tweets using Support Vector Machines and Conditional Random Fields, benchmarked with a Naïve Bayes Classifier baseline, but were unable to beat the baseline. Later, and as Twitter has grown in popularity, many other systems for Twitter Sentiment Analysis (TSA) have been developed (see, e.g., Maynard and Funk, 2011; Mukherjee et al., 2012; Saif et al., 2012; Chamlertwat et al., 2012).

Clearly, it is possible to classify the sentiment of tweets in a single step; however, the approach to TSA most used so far is a two-step strategy where the first step is subjectivity classification and the second step is polarity classification. The goal of *subjectivity classification* is to separate subjective and objective statements. Pak and Paroubek (2010) counted word frequencies in a subjective vs an objective set of tweets; the results showed that interjections and personal pronouns are the strongest indicators of subjectivity. In general, these word classes, adverbs and (in particular) adjectives (Hatzivassiloglou and Wiebe, 2000) have shown to be good subjectivity indicators, which has made part-

[1] See Pang and Lee (2008); Feldman (2013) for overviews.

*Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 430–437, Atlanta, Georgia, June 14-15, 2013.

of-speech (POS) tagging a reasonable technique for filtering out objective tweets. Early research on TSA showed that the challenging vocabulary made it harder to accurately tag tweets; however, Gimpel et al. (2011) report on using a POS tagger for marking tweets, performing with almost 90% accuracy.

Polarity classification is the task of separating the subjective statements into positives and negatives. Kouloumpis et al. (2011) tried different solutions for tweet polarity classification, and found that the best performance came from using n-grams together with lexicon and microblog features. Interestingly, performance dropped when a POS tagger was included. They speculate that this can be due to the accuracy of the POS tagger itself, or that POS tagging just is less effective for analysing tweet polarity.

In this paper we will explore the application of a set of machine learning algorithms to the task of Twitter sentiment classification, comparing one-step and two-step approaches, and investigate a range of different preprocessing methods. What we explicitly will not do, is to utilise a sentiment lexicon, even though many methods in TSA rely on lexica with a sentiment score for each word. Nielsen (2011) manually built a sentiment lexicon specialized for Twitter, while others have tried to induce such lexica automatically with good results (Velikovich et al., 2010; Mohammad et al., 2013). However, sentiment lexica — and in particular specialized Twitter sentiment lexica — make the classification more domain dependent. Here we will instead aim to exploit domain independent approaches as far as possible, and thus abstain from using sentiment lexica. The rest of the paper is laid out as follows: Section 2 introduces the twitter data sets used in the study. Then Section 3 describes the system built for carrying out the twitter sentiment classification experiments, which in turn are reported and discussed in Sections 4 and 5.

2 Data

Manually collecting information from Twitter would be a tedious task, but Twitter offers a well documented Representational State Transfer Application Programming Interface (REST API) which allows users to collect a corpus from the microblogosphere. Most of the data used in TSA research is collected through the Twitter API, either by searching for a certain topic/keyword or by streaming realtime data. Four different data sets were used in the experiments described below. three were supplied by the organisers of the SemEval'13 shared task on Twitter sentiment analysis (Wilson et al., 2013), in the form of a training set, a smaller initial development set, and a larger development set. All sets consist of manually annotated tweets on a range of topics, including different products and events.

	Training		Dev 1		Dev 2		NTNU	
Class	Num	%	Num	%	Num	%	Num	%
Negative	1288	15	176	21	340	26	86	19
Neutral	4151	48	144	45	739	21	232	50
Positive	3270	37	368	35	575	54	142	31
Total	8709		688		1654		461	

Table 1: The data sets used in the experiments

Tweet-level classification (Task 2B) was split into two subtasks in SemEval'13, one allowing training only on the data sets supplied by the organisers (*constrained*) and one allowing training also on external data (*unconstrained*). To this end, a web application[2] for manual annotation of tweets was built and used to annotate a small fourth data set ('NTNU'). Each of the 461 tweets in the 'NTNU' data set was annotated by one person only.

The distribution of target classes in the data sets is shown in Table 1. The data was neither preprocessed nor filtered, and thus contain hashtags, URLs, emoticons, etc. However, all the data sets provided by SemEval'13 had more than three target classes (e.g., 'objective', 'objective-OR-neutral'), so tweets that were not annotated as 'positive' or 'negative' were merged into the 'neutral' target class.

Due to Twitter's privacy policy, the given data sets do not contain the tweet text, but only the tweet ID which in turn can be used to download the text. The Twitter API has a limit on the number of downloads per hour, so SemEval'13 provided a Python script to scrape texts from `https://twitter.com`. This script was slow and did not download the texts for all tweet IDs in the data sets, so a faster and more precise download script[3] for node.js was implemented and submitted to the shared task organisers.

[2] `http://tinyurl.com/tweetannotator`
[3] `http://tinyurl.com/twitscraper`

3 Experimental Setup

In order to run sentiment classification experiments, a general system was built. It has a Sentiment Analysis API Layer which works as a thin extension of the Twitter API, sending all tweets received in parallel to a Sentiment Analysis Classifier server. After classification, the SA API returns the same JSON structure as the Twitter API sends out, only with an additional attribute denoting the tweet's sentiment. The Sentiment Analysis Classifier system consists of preprocessing and classification, described below.

3.1 Preprocessing

As mentioned in the introduction, most approaches to Twitter sentiment analysis start with a preprocessing step, filtering out some Twitter specific symbols and functions. Go et al. (2009) used ':)' and ':(' as labels for the polarity, so did not remove these emoticons, but replaced URLs and user names with placeholders. Kouloumpis et al. (2011) used both an emoticon set and a hashtagged set. The latter is a subset of the Edinburgh Twitter corpus which consists of 97 million tweets (Petrović et al., 2010). Some approaches have also experimented with normalizing the tweets, and removing redundant letters, e.g., "loooove" and "crazyyy", that are used to express a stronger sentiment in tweets. Redundant letters are therefore often not deleted, but words rather trimmed down to one additional redundant letter, so that the stronger sentiment can be taken into consideration by a score/weight adjustment for that feature.

To find the best features to use, a set of eight different combinations of preprocessing methods was designed, as detailed in Table 2. These include no preprocessing ($P0$, not shown in the table), where all characters are included as features; full remove ($P4$), where all special Twitter features like user names, URLs, hashtags, retweet (RT) tags, and emoticons are stripped; and replacing Twitter features with placeholder texts to reduce vocabulary. The "hashtag as word" method transforms a hashtag to a regular word and uses the hashtag as a feature. "Reduce letter duplicate" removes redundant characters more than three ("happyyyyyyyy!!!!!!" → "happyyy!!!"). Some methods, like $P1$, $P2$, $P4$, $P5$ and $P7$ remove user names from the text, as they most likely are just noise for the sentiment. Still, the fact that there are references to URLs and user names might be relevant for the sentiment. To make these features more informative for the machine learning algorithms, a preprocessing method ($P3$) was implemented for replacing them with placeholders. In addition, a very rudimentary treatment of negation was added, in which the negation is attached to the preceding and following words, so that they will also reflect the change in sentence polarity.

Method	$P1$	$P2$	$P3$	$P4$	$P5$	$P6$	$P7$
Remove Usernames	✓	✓		✓	✓		✓
Username placeholder			✓				
Remove URLs		✓		✓	✓		✓
URL placeholder			✓				
Remove hashtags				✓	✓		
Hashtag as word		✓					
Hashtag placeholder			✓				
Remove RT-tags		✓		✓	✓		
Remove emoticons				✓	✓		
Reduce letter duplicate		✓		✓		✓	✓
Negation attachment				✓		✓	✓

Table 2: Overview of the preprocessing methods

Even though this preprocessing obviously is Twitter-specific, the results after it will still be domain semi-independent, in as far as the strings produced after the removal of URLs, user names, etc., will be general, and can be used for system training.

3.2 Classification

The classification step currently supports three machine learning algorithms from the Python `scikit-learn`[4] package: Naïve Bayes (NB), Maximum Entropy (MaxEnt), and Support Vector Machines (SVM). These are all among the supervised learners that previously have been shown to perform well on TSA, e.g., by Bermingham and Smeaton (2010) who compared SVM and NB for microblogs. Interestingly, while the SVM technique normally beats NB and MaxEnt on longer texts, that comparison indicated that it has some trouble with outperforming NB when feature vectors are shorter. Three different models were implemented:

1. *One-step model*: a single algorithm classifies tweets as negative, neutral or positive.
2. *Two-step model*: the tweets are first classified as either subjective or neutral. Those that are

[4] `http://scikit-learn.org`

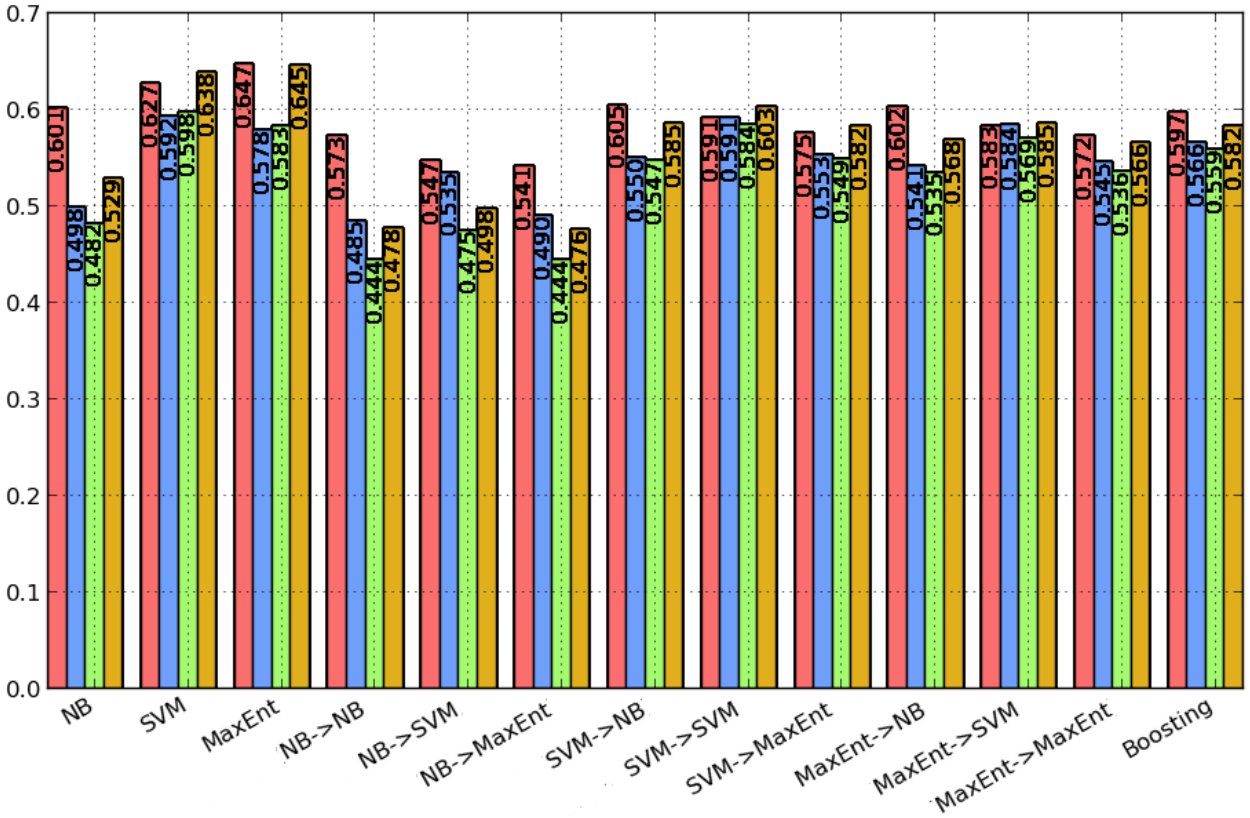


Figure 1: Performance across all models (red=precision, blue=recall, green=F_1-score, brown=accuracy)

classified as subjective are then sent to polarity classification (i.e., negative or positive).

3. *Boosting* (Freund and Schapire, 1997): a way to combine classifiers by generating a set of sub-models, each of which predicts a sentiment on its own and then sends it to a voting process that selects the sentiment with highest score.

In all cases, the final classification is returned to the API Layer sentiment provider.

4 Experimental Results

Experiments were carried out using the platform introduced in the previous section, with models built on the training set of Table 1. The testing system generates and trains different models based on a set of parameters, such as classification algorithm, preprocessing methods, whether or not to use inverse document frequency (IDF) or stop words. A grid search option can be activated, so that a model is generated with the best possible parameter set for the given algorithm, using 10-fold cross validation.

4.1 Selection of Learners and Features

An extensive grid search was conducted. This search cycled through different algorithms, parameters and preprocessing techniques. The following parameters were included in the search. Three binary (`Yes/No`) parameters: `Use IDF`, `Use Smooth IDF`, and `Use Sublinear IDF`, together with `ngram` (`unigram/bigram/trigram`). SVM and MaxEnt models in addition included `C` and NB models `alpha` parameters, all with the value ranges [`0.1/0.3/0.5/0.7/0.8/1.0`]. SVM and MaxEnt models also had `penalty` (`L1/L2`).

Figure 1 displays the precision, recall, F_1-score, and accuracy for each of the thirteen classifiers with the Dev 2 data set (see Table 1) used for evaluation. Note that most classifiers involving the NB algorithm perform badly, both in terms of accuracy and F-score. This was observed for the other data sets as well. Further, we can see that one-step classifiers did better than two-step models, with MaxEnt obtaining the best accuracy, but SVM a slightly better F-score.

Data set	Dev 2		Dev 1	
Learner	SVM	MaxEnt	SVM	MaxEnt
Precision	0.627	**0.647**	**0.700**	0.561
Recall	**0.592**	0.578	**0.726**	0.589
F_1-score	**0.598**	0.583	**0.707**	0.556
Accuracy	0.638	**0.645**	**0.728**	0.581

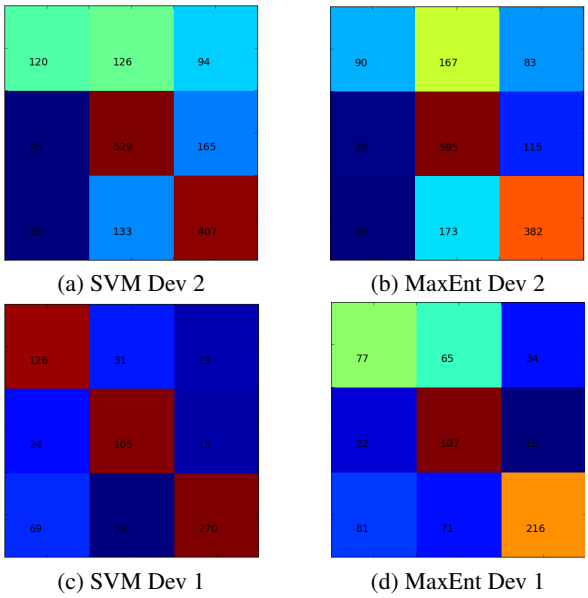
Table 3: Best classifier performance (bold=best score; all classifiers were trained on the training set of Table 1)

A second grid search with the two best classifiers from the first search was performed instead using the smaller Dev 1 data set for evaluation. The results for both the SVM and MaxEnt classifiers are shown in Table 3. With the Dev 1 data set, SVM performs much better than MaxEnt. The larger Dev 2 development set contains more neutral tweets than the Dev 1 set, which gives us reasons to believe that evaluating on the Dev 2 set favours the MaxEnt classifier.

A detailed error analysis was conducted by inspecting the confusion matrices of all classifiers. In general, classifiers involving SVM tend to give better confusion matrices than the others. Using SVM only in a one-step model works well for positive and neutral tweets, but a bit poorer for negative. Two-step models with SVM-based subjectivity classification exhibit the same basic behaviour. The one-step MaxEnt model classifies more tweets as neutral than the other classifiers. Using MaxEnt for subjectivity classification and either MaxEnt or SVM for polarity classification performs well, but is too heavy on the positive class. Boosting does not improve and behaves in a fashion similar to two-step MaxEnt models. All combinations involving NB tend to heavily favour positive predictions; only the two-step models involving another algorithm for polarity classification gave some improvement for negative tweets.

The confusion matrices of the two best learners are shown in Figures 2a-2d, where a learner is shown to perform better if it has redish colours on the main diagonal and blueish in the other fields, as is the case for SVM on the Dev 1 data set (Figure 2c).

As a part of the grid search, all preprocessing methods were tested for each classifier. Figure 3 shows that $P2$ (removing user names, URLs, hashtags prefixes, retweet tokens, and redundant letters) is the preprocessing method which performs best



(a) SVM Dev 2 (b) MaxEnt Dev 2

(c) SVM Dev 1 (d) MaxEnt Dev 1

Figure 2: SVM and MaxEnt confusion matrices (output is shown from left-to-right: negative-neutral-positive; the correct classes are in the same order, top-to-bottom. "Hotter" colours (red) indicate that more instances were assigned; "colder" colours (blue) mean fewer instances.)

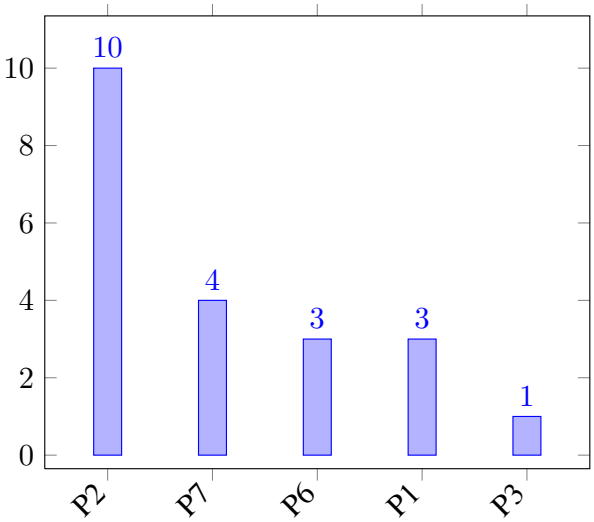


Figure 3: Statistics of preprocessing usage

(gives the best accuracy) and thus used most often (10 times). Figure 3 also indicates that URLs are noisy and do not contain much sentiment, while hashtags and emoticons tend to be more valuable features ($P2$ and $P7$ — removing URLs — perform best, while $P4$ and $P5$ — removing hashtags and emoticons in addition to URLs — perform badly).

Data set	Twitter		SMS	
System	NTNUC	NTNUU	NTNUC	NTNUU
Precision	**0.652**	0.633	**0.659**	0.623
Recall	**0.579**	0.564	**0.646**	0.623
F_1-score	**0.590**	0.572	**0.652**	0.623
$F_1 +/-$	**0.532**	0.507	**0.580**	0.546

Table 4: The NTNU systems in SemEval'13

4.2 SemEval'13 NTNU Systems and Results

Based on the information from the grid search, two systems were built for SemEval'13. Since one-step SVM-based classification showed the best performance on the training data, it was chosen for the system participating in the *constrained* subtask, **NTNUC**. The preprocessing also was the one with the best performance on the provided data, $P2$ which involves lower-casing all letters; reducing letter duplicates; using hashtags as words (removing #); and removing all URLs, user names and RT-tags.

Given the small size of the in-house ('NTNU') data set, no major improvement was expected from adding it in the *unconstrained* task. Instead, a radically different set-up was chosen to create a new system, and train it on both the in-house and provided data. **NTNUU** utilizes a two-step approach, with SVM for subjectivity and MaxEnt for polarity classification, a combination intended to capture the strengths of both algorithms. No preprocessing was used for the subjectivity step, but user names were removed before attempting polarity classification.

As further described by Wilson et al. (2013), the SemEval'13 shared task involved testing on a set of 3813 tweets (1572 positive, 601 negative, and 1640 neutral). In order to evaluate classification performance on data of roughly the same length and type, but from a different domain, the evaluation data also included 2094 Short Message Service texts (SMS; 492 positive, 394 negative, and 1208 neutral).

Table 4 shows the results obtained by the NTNU systems on the SemEval'13 evaluation data, in terms of average precision, recall and F-score for all three classes, as well as average F-score for positive and negative tweets only ($F_1 +/-$; i.e., the measure used to rank the systems participating in the shared task).

5 Discussion and Conclusion

As can be seen in Table 4, the extra data available to train the NTNUU system did not really help it: it gets outperformed by NTNUC on all measures. Notably, both systems perform well on the out-of-domain data represented by the SMS messages, which is encouraging and indicates that the approach taken really is domain semi-independent. This was also reflected in the rankings of the two systems in the shared task: both were on the lower half among the participating systems on Twitter data ($24^{\text{th}}/36$ resp. $10^{\text{th}}/15$), but near the top on SMS data, with NTNUC being ranked 5^{th} of 28 constrained systems and NTNUU 6^{th} of 15 unconstrained systems.

Comparing the results to those shown in Table 3 and Figure 1, NTNUC's (SVM) performance is in line with that on Dev 2, but substantially worse than on Dev 1; NTNUU (SVM→MaxEnt) performs slightly worse too. Looking at system output with and without the 'NTNU' data, both one-step SVM and MaxEnt models and SVM→MaxEnt classified more tweets as negative when trained on the extra data; however, while NTNUC benefited slightly from this, NTNUU even performed better without it.

An obvious extension to the present work would be to try other classification algorithms (e.g., Conditional Random Fields or more elaborate ensembles) or other features (e.g., character n-grams). Rather than the very simple treatment of negation used here, an approach to automatic induction of scope through a negation detector (Councill et al., 2010) could be used. It would also be possible to relax the domain-independence further, in particular to utilize sentiment lexica (including twitter specific), e.g., by automatic phrase-polarity lexicon extraction (Velikovich et al., 2010). Since many users tweet from their smartphones, and a large number of them use iPhones, several tweets contain iPhone-specific smilies ("Emoji"). Emoji are implemented as their own character set (rather than consisting of characters such as ':)' and ':(', etc.), so a potentially major improvement could be to convert them to character-based smilies or to emotion-specific placeholders.

Acknowledgements

Thanks to Amitava Das for initial discussions and to the human annotators of the 'NTNU' data set.

References

Bermingham, A. and Smeaton, A. F. (2010). Classifying sentiment in microblogs: Is brevity an advantage? In *Proceedings of the 19th International Conference on Information and Knowledge Management*, pages 1833–1836, Toronto, Canada. ACM.

Chamlertwat, W., Bhattarakosol, P., Rungkasiri, T., and Haruechaiyasak, C. (2012). Discovering consumer insight from Twitter via sentiment analysis. *Journal of Universal Computer Science*, 18(8):973–992.

Councill, I. G., McDonald, R., and Velikovich, L. (2010). What's great and what's not: learning to classify the scope of negation for improved sentiment analysis. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 51–59, Uppsala, Sweden. ACL. Workshop on Negation and Speculation in Natural Language Processing.

Feldman, R. (2013). Techniques and applications for sentiment analysis. *Communications of the ACM*, 56(4):82–89.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., and Smith, N. A. (2011). Part-of-speech tagging for Twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 2: short papers, pages 42–47, Portland, Oregon. ACL.

Go, A., Huang, L., and Bhayani, R. (2009). Twitter sentiment analysis. Technical Report CS224N Project Report, Department of Computer Science, Stanford University, Stanford, California.

Hatzivassiloglou, V. and Wiebe, J. M. (2000). Effects of adjective orientation and gradability on sentence subjectivity. In *Proceedings of the 18th International Conference on Computational Linguistics*, pages 299–305, Saarbrücken, Germany. ACL.

HLT10 (2010). *Proceedings of the 2010 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, Los Angeles, California. ACL.

Kouloumpis, E., Wilson, T., and Moore, J. (2011). Twitter sentiment analysis: The good the bad and the OMG! In *Proceedings of the 5th International Conference on Weblogs and Social Media*, pages 538–541, Barcelona, Spain. AAAI.

Maynard, D. and Funk, A. (2011). Automatic detection of political opinions in tweets. In #MSM2011 (2011), pages 81–92.

Mohammad, S., Kiritchenko, S., and Zhu, X. (2013). NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. In SemEval'13 (2013).

#MSM2011 (2011). *Proceedings of the 1st Workshop on Making Sense of Microposts (#MSM2011)*, Heraklion, Greece.

Mukherjee, S., Malu, A., Balamurali, A., and Bhattacharyya, P. (2012). TwiSent: A multistage system for analyzing sentiment in Twitter. In *Proceedings of the 21st International Conference on Information and Knowledge Management*, pages 2531–2534, Maui, Hawaii. ACM.

Nielsen, F. Å. (2011). A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. In #MSM2011 (2011), pages 93–98.

Pak, A. and Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, Valetta, Malta. ELRA.

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.

Petrović, S., Osborne, M., and Lavrenko, V. (2010). The Edinburgh Twitter corpus. In HLT10 (2010), pages 25–26. Workshop on Computational Linguistics in a World of Social Media.

Saif, H., He, Y., and Alani, H. (2012). Semantic sentiment analysis of Twitter. In *Proceedings of the 11th International Semantic Web Conference*, pages 508–524, Boston, Massachusetts. Springer.

SemEval'13 (2013). *Proceedings of the International Workshop on Semantic Evaluation*, SemEval '13, Atlanta, Georgia. ACL.

Velikovich, L., Blair-Goldensohn, S., Hannan, K., and McDonald, R. (2010). The viability of web-derived polarity lexicons. In HLT10 (2010), pages 777–785.

Wilson, T., Kozareva, Z., Nakov, P., Ritter, A., Rosenthal, S., and Stoyanov, V. (2013). SemEval-2013 Task 2: Sentiment analysis in Twitter. In SemEval'13 (2013).

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the 2005 Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 347–354, Vancouver, British Columbia, Canada. ACL.

Yu, H. and Hatzivassiloglou, V. (2003). Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In Collins, M. and Steedman, M., editors, *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 129–136, Sapporo, Japan. ACL.