

# AVAYA: Sentiment Analysis on Twitter with Self-Training and Polarity Lexicon Expansion

Lee Becker, George Erhart, David Skiba and Valentine Matula

Avaya Labs Research

1300 West 120th Avenue

Westminster, CO 80234, USA

{beckerl, gerhart, dskiba, matula}@avaya.com

## Abstract

This paper describes the systems submitted by Avaya Labs (AVAYA) to SemEval-2013 Task 2 - Sentiment Analysis in Twitter. For the constrained conditions of both the message polarity classification and contextual polarity disambiguation subtasks, our approach centers on training high-dimensional, linear classifiers with a combination of lexical and syntactic features. The constrained message polarity model is then used to tag nearly half a million unlabeled tweets. These automatically labeled data are used for two purposes: 1) to discover prior polarities of words and 2) to provide additional training examples for self-training. Our systems performed competitively, placing in the top five for all subtasks and data conditions. More importantly, these results show that expanding the polarity lexicon and augmenting the training data with unlabeled tweets can yield improvements in precision and recall in classifying the polarity of non-neutral messages and contexts.

## 1 Introduction

The past decade has witnessed a massive expansion in communication from long-form delivery such as e-mail to short-form mechanisms such as microblogging and short messaging service (SMS) text messages. Simultaneously businesses, media outlets, and investors are increasingly relying on these messages as sources of real-time information and are increasingly turning to sentiment analysis to discover product trends, identify customer preferences, and categorize users. While a variety of corpora ex-

ist for developing and evaluating sentiment classifiers for long-form texts such as product reviews, there are few such resources for evaluating sentiment algorithms on microblogs and SMS texts.

The organizers of SemEval-2013 task 2, have begun to address this resource deficiency by coordinating a shared evaluation task for Twitter sentiment analysis. In doing so they have assembled corpora in support of the following two subtasks:

### Task A - Contextual Polarity Disambiguation

*“Given a message containing a marked instance of a word or phrase, determine whether that instance is positive, negative or neutral in that context.”*

**Task B - Message Polarity Classification** *“Given a message, classify whether the message is of positive, negative, or neutral sentiment. For messages conveying both a positive and negative sentiment, whichever is the stronger sentiment should be chosen.”*

This paper describes the systems submitted by Avaya Labs for participation in subtasks A and B. Our goal for this evaluation was to investigate the usefulness of dependency parses, polarity lexicons, and unlabeled tweets for sentiment classification on short messages. In total we built four systems for SemEval-2013 task 2. For task B we developed a constrained model using supervised learning, and an unconstrained model that used semi-supervised learning in the form of self-training and polarity lexicon expansion. For task A the constrained system utilized supervised learning, while the unconstrained model made use of the expanded lexicon

from task B. Output from these systems were submitted to all eight evaluation conditions. For a complete description of the data, tasks, and conditions, please refer to Wilson et al. (2013). The remainder of this paper details the approaches, experiments and results associated with each of these models.

## 2 Related Work

Over the past few years sentiment analysis has grown from a nascent topic in natural language processing to a broad research area targeting a wide range of text genres and applications. There is now a significant body of work that spans topics as diverse as document level sentiment classification (Pang and Lee, 2008), induction of word polarity lexicons (Hatzivassiloglou and McKeown, 1997; Turney, 2002; Esuli and Sebastiani, 2006; Mohammad and Turney, 2011) and even election prediction (Tumasjan et al., 2010).

Efforts to train sentiment classifiers for Twitter messages have largely relied on using emoticons and hashtags as proxies of the true polarity (Barbosa and Feng, 2010; Davidov et al., 2010b; Pak and Paroubek, 2010; Agarwal et al., 2011; Kouloumpis et al., 2011; Mohammad, 2012). Classification of word and phrase sentiment with respect to surrounding context (Wilson et al., 2005) has yet to be explored for the less formal language often found in microblog and SMS text. Semi-supervised learning has been applied to polarity lexicon induction (Rao and Ravichandran, 2009), and sentiment classification at the sentence level (Täckström and McDonald, 2011) and document level (Sindhwani and Melville, 2008; He and Zhou, 2011); however to the best of our knowledge self-training and other semi-supervised learning has seen only minimal use in classifying Twitter texts (Davidov et al., 2010a; Zhang et al., 2012).

## 3 System Overview

Given our overarching goal of combining polarity lexicons, syntactic information and unlabeled data, our approach centered on first building strong constrained models and then improving performance by adding additional data and resources. For both tasks, our data-constrained approach combined standard features for document classification

$conj \rightarrow conj\_ \langle conjunction \rangle$
$pobj \rightarrow prep\_ \langle preposition \rangle$
$pcomp \rightarrow prepc\_ \langle preposition \rangle$
$prep punct cc \rightarrow \emptyset$

Table 1: Collapsed Dependency Transformation Rules

with dependency parse and word polarity features into a weighted linear classifier. For our data-unconstrained models we used pointwise mutual information for lexicon expansion in conjunction with self-training to increase the size of the feature space.

## 4 Preprocessing and Text Normalization

Our systems were built with ClearTK (Ogren et al., 2008) a framework for developing NLP components built on top of Apache UIMA. Our preprocessing pipeline utilized ClearTK’s wrappers for ClearNLP’s (Choi and McCallum, 2013) tokenizer, lemmatizer, part-of-speech (POS) tagger, and dependency parser. ClearNLP’s ability to retain emoticons and emoji as individual tokens made it especially attractive for sentiment analysis. POS tags were mapped from Penn Treebank-style tags to the simplified, Twitter-oriented tags introduced by Gimpel et al. (2011). Dependency graphs output by ClearNLP were also transformed to the Stanford Collapsed dependencies representation (de Marneffe and Manning, 2012) using our own transformation rules (table 1). Input normalization consisted solely of replacing all usernames and URLs with common placeholders.

## 5 Sentiment Resources

A variety of our classifier features rely on manually tagged sentiment lexicons and word lists. In particular we make use of the MPQA Subjectivity Lexicon (Wiebe et al., 2005) as well as manually-created negation and emoticon dictionaries<sup>1</sup>. The negation word list consisting of negation words such as *no* and *not*. Because tokenization splits contractions, the list includes the sub-word token *n’t* as well as the apostrophe-less version of 12 contractions (e.g. *cant*, *wont*, etc ...). To support emoticon-specific features we created a dictionary, which paired 183 emoticons with either a positive or negative polarity.

<sup>1</sup><http://leebecker.com/resources/semEval-2013>

## 6 Message Polarity Classification

### 6.1 Features

**Polarized Bag-of-Words Features:** Instead of extracting raw bag-of-words (BOW), we opted to integrate negation directly into the word representations following the approaches used by Das and Chen (2001) and Pang et al. (2002). All words between a negation word and the first punctuation mark after the negation word were suffixed with a **\_NOT** tag – essentially doubling the number of BOW features. We extended this polarized BOW paradigm to include not only the raw word forms but all of the following combinations: raw word, raw word+PTB POS tag, raw word+simplified POS tag, lemma+simplified POS tag.

**Word Polarity Features:** Using a subjectivity lexicon, we extracted features for the number of positive, negative, and neutral words as well as the net polarity based on these counts. Individual word polarities were inverted if the word had a child dependency relation with a negation (*neg*) label. Constrained models use the MPQA lexicon, while unconstrained models use an expanded lexicon that is described in section 6.2.

**Emoticon Features:** Similar to the word polarity features, we computed features for the number of positive, negative, and neutral emoticons, and the net emoticon polarity score.

**Microblogging Features:** As noted by Kouloumpis et al. (2011), the emotional intensity of words in social media messages is often emphasized by changes to the word form such as capitalization, character repetition, and emphasis characters (asterisks, dashes). To capture this intuition we compute features for the number of fully-capitalized words, words with characters repeated more than 3 times (e.g. *booooo*), and words surrounded by asterisks or dashes (e.g. *\*yay\**). We also created a binary feature to indicate the presence of a winning score or winning record within the target span (e.g. *Oh yeah #Nuggets 15-0*).

**Part-of-Speech Tag Features:** Counts of the Penn Treebank POS tags provide a rough measure of the content of the message.

**Syntactic Dependency Features:** We extracted dependency pair features using both standard and collapsed dependency parse graphs. Extracted

head/child relations include: raw word/raw word, lemma/lemma, lemma/simplified POS tag, simplified POS tag/lemma. If the head node of the relation has a child negation dependency, the pair’s relation label is prefixed with a **NEG\_** tag.

### 6.2 Expanding the Polarity Lexicon

Unseen words pose a recurring challenge for both machine learning and dictionary-based approaches to sentiment analysis. This problem is even more prevalent in social media and SMS messages where text lengths are often limited to 140 characters or less. To expand our word polarity lexicon we adopt a framework similar to the one introduced by Turney (2002). Turney’s unsupervised approach centered on computing pointwise mutual information (PMI) between highly polar seed words and bigram phrases extracted from a corpus of product reviews.

Instead of relying solely on seed words for polarity, we use the constrained version of the message polarity classifier to tag a corpus of approximately 475,000 unlabeled, English language tweets. These tweets were collected over the period from November 2012 to February 2013. To reduce the number of noisy instances and to obtain a more balanced distribution of sentiment labels, we eliminated all tweets with classifier confidence scores below 0.9, 0.7, and 0.8 for *positive*, *negative* and *neutral* instances respectively. Applying the threshold, reduced the tweet count to 180,419 tweets (50,789 positive, 59,029 negative, 70,601 neutral). This filtered set of automatically labeled tweets was used to accumulate co-occurrence statistics between the words in the tweets and their corresponding sentiment labels. These statistics are then used to compute word-sentiment PMI (equation 1), which is the joint probability of a word and sentiment co-occurring divided by the probability of each of the events occurring independently. A word’s net polarity is computed as the signum (*sgn*) of the difference between its positive and negative PMI values (equation 2). It should be noted that polarities were deliberately limited to values of  $\{-1, 0, +1\}$  to ensure consistency with the existing MPQA lexicon, and to dampen the bias of any single word.

$$PMI(word, sentiment) = \log_2 \frac{p(word, sentiment)}{p(word)p(sentiment)} \quad (1)$$

$$polarity(word) = \text{sgn}(PMI(word, positive) - PMI(word, negative)) \quad (2)$$

Words with fewer than 10 occurrences, words with neutral polarities, numbers, single characters, and punctuation were then removed from this PMI-derived polarity dictionary. Lastly, this dictionary was merged with the dictionary created from the MPQA lexicon yielding a final polarity dictionary with 11,740 entries. In cases where an entry existed in both dictionaries, the MPQA polarity value was retained. This final polarity dictionary was used by the unconstrained models for task A and B.

### 6.3 Model Parameters and Training

**Constrained Model:** Models were trained using the LIBLINEAR classification library (Fan et al., 2008). L2 regularized logistic regression was chosen over other LIBLINEAR loss functions because it not only gave improved performance on the development set but also produced calibrated outcomes for confidence thresholding. Training data for the constrained model consisted of all 9829 examples from the training (8175 examples) and development (1654 examples) set released for SemEval 2013. Cost and label-specific cost weight parameters were selected via experimentation on the development set to maximize the average positive and negative  $F_1$  values. The  $c$  values ranged over  $\{0.1, 0.5, 1, 2, 5, 10, 20, 100\}$  and the label weights  $w_{polarity}$  ranged over  $\{0.1, 1, 2, 5, 10, 20, 25, 50, 100\}$ . Final parameters for the constrained model were cost  $c = 1$  and weights  $w_{positive} = 1$ ,  $w_{negative} = 25$ , and  $w_{neutral} = 1$ .

**Unconstrained Model:** In addition to using the expanded polarity dictionary described in 6.2 for feature extraction, the unconstrained model also makes use of automatically labeled tweets for self-training (Scudder, 1965). In contrast to preparation of the expanded polarity dictionary, the self-training placed no threshold on the examples. Combining the self-labeled tweets, with the official training and development set yielded a new training set consisting

of 485,112 examples. Because the self-labeled instances were predominantly tagged *neutral*, the LIBLINEAR cost parameters were adjusted to heavily discount *neutral* while emphasizing *positive* and *neutral* instances. The size and cost of training this model prevented extensive parameter tuning and instead were chosen based on experience with the constrained model and to maximize recall on positive and negative items. Final parameters for the unconstrained model were cost  $c = 1$  and category weights  $w_{positive} = 2$ ,  $w_{negative} = 5$ , and  $w_{neutral} = 0.1$ .

## 7 Contextual Polarity Disambiguation

### 7.1 Features

The same base set of features used for message polarity classification were used for the contextual polarity classification, with the exception of the syntactic dependency features. To better express the in-context and out-of-context relation these additional feature classes were added:

**Scoped Dependency Features:** Because this task focuses on a smaller context within the message, collapsed dependencies are less useful as the compression may cross over context boundaries. Instead the standard syntactic dependency features described above were modified to account for their relation to the context. All governing relations for the words contained within the context were extracted. Relations wholly contained within the boundaries of the context were prefixed with an **IN** tag, whereas those that crossed outside of the context were prefixed with an **OUT** tag. Additionally counts of IN and OUT relations were included as features.

**Dependency Path Features:** Like the single dependency arcs, a dependency path can provide additional information about the syntactic and semantic role of the context in the sentence. Our path features consisted of two varieties: 1) POS-path and 2) Sentiment-POS-path. The POS-path consisted of the PTB POS tags and dependency relation labels for all nodes between the head of the context and the root node of the parent sentence. The Sentiment-POS-path follows the same path but omits the dependency relation labels, uses the simplified POS tags and appends word polarities (POS/NEG/NTR) to the POS tags along the path.

	System	Positive			Negative			Neutral			$F_{avg}$ +/-	Rank
		P	R	F	P	R	F	P	R	F		
Tweet	NRC-Canada (top)	0.814	0.667	0.733	0.697	0.604	0.647	0.677	0.826	0.744	0.690	1
	AVAYA-Unconstrained	0.751	0.655	0.700	0.608	0.557	0.582	0.665	0.768	0.713	0.641	5
	AVAYA-Constrained	0.791	0.580	0.669	0.593	0.509	0.548	0.636	0.832	0.721	0.608	12
	<i>Mean of submissions</i>	0.687	0.591	0.626	0.491	0.456	0.450	0.612	0.663	0.615	0.538	-
SMS	NRC-Canada (top)	0.731	0.730	0.730	0.554	0.754	0.639	0.852	0.753	0.799	0.685	1
	AVAYA-Constrained	0.630	0.667	0.648	0.526	0.581	0.553	0.802	0.756	0.778	0.600	4
	AVAYA-Unconstrained	0.609	0.659	0.633	0.494	0.637	0.557	0.814	0.710	0.759	0.595	5
	<i>Mean of submissions</i>	0.512	0.620	0.546	0.462	0.518	0.456	0.754	0.578	0.627	0.501	-

Table 2: Message Polarity Classification (Task B) Results

	System	Positive			Negative			Neutral			$F_{avg}$ +/-	Rank
		P	R	F	P	R	F	P	R	F		
Tweet	NRC-Canada (top)	0.889	0.932	0.910	0.866	0.871	0.869	0.455	0.063	0.110	0.889	1
	AVAYA-Unconstrained	0.892	0.905	0.898	0.834	0.865	0.849	0.539	0.219	0.311	0.874	2
	AVAYA-Constrained	0.882	0.911	0.896	0.844	0.843	0.843	0.493	0.225	0.309	0.870	3
	<i>Mean of submissions</i>	0.837	0.745	0.773	0.745	0.656	0.677	0.159	0.240	0.115	0.725	-
SMS	GUMTLT (top)	0.814	0.924	0.865	0.908	0.896	0.902	0.286	0.050	0.086	0.884	1
	AVAYA-Unconstrained	0.815	0.871	0.842	0.853	0.896	0.874	0.448	0.082	0.138	0.858	3
	AVAYA-Constrained	0.777	0.875	0.823	0.859	0.852	0.856	0.364	0.076	0.125	0.839	4
	<i>Mean of submissions</i>	0.734	0.722	0.710	0.807	0.663	0.698	0.144	0.184	0.099	0.704	-

Table 3: Contextual Polarity Disambiguation (Task A) Results

For example given the bold-faced context in the sentence:

*@User Criminals killed Sadat, and in the process they killed **Egypt**. . . **they** destroyed the future of young & old Egyptians..*

the extracted POS-path feature would be:

{NNP} dobj <{VBD} conj <{VBD}  
ccomp <{VBD} root <{TOP}

while the Sentiment-POS path would be:

{^/pos}{V/neg}{V/neg}{V/neg}{TOP}.

Paths with depth greater than 4 dependency relations were truncated to reduce feature sparsity. In addition to these detailed path features, we include two binary features to indicate if any part of the path contains subject or object relations.

## 7.2 Model Parameters and Training

Like with message polarity classification, the contextual polarity disambiguation systems rely on LBLINEAR’s L2 regularized logistic regression for model training. Both constrained and unconstrained models use identical parameters of cost  $c = 1$

and weights  $w_{positive} = 1$ ,  $w_{negative} = 2$ , and  $w_{neutral} = 1$ . They vary only in the choice of polarity lexicon. The constrained model uses the MPQA subjectivity lexicon, while the unconstrained model uses the expanded dictionary derived via computation of PMI, which ultimately differentiates these models through the variation in the sentiment path and word polarity features.

## 8 Experiments and Results

In this section we report results for the series of Sentiment Analysis in Twitter tasks at SemEval 2013. Please refer to refer to Wilson et al. (2013) for the exact details about the corpora, evaluation conditions, and methodology.

We submitted polarity output for the Message Polarity Classification (task B) and the Contextual Polarity Disambiguation (task A). For each task we submitted system output from our constrained and unconstrained models. As stated above, the constrained models made use of only the training data released for the task, whereas the unconstrained models trained on additional tweets. Each subtask had two test sets one comprised of tweets and the other comprised of SMS messages. Final task 2

	S	G	Message / Context
1	+	/	<i>Going to Helsinki tomorrow or on the day after tomorrow,yay!</i>
2	/	+	<i>Eric Decker catches his second TD pass from Manning. This puts Broncos up 31-7 with 14:54 left in the 4th.</i>
3	-	/	<i>So, crashed a wedding reception and Andy Lee's bro was in the bridal party. How'd you spend your Saturday night? #longstory</i>
4	-	+	<i>Aiyo... <b>Dun worry</b> la, they'll let u change one... Anyway, sleep early, nite nite...</i>
5	+	-	<i><b>Sori</b> I haven't done anything for today's meeting.. pls pardon me. Cya guys later at 10am.</i>
6	+	-	<i>these PSSA's are just gonna be the icing to another horrible monday. #fmlll #ihateschool</i>

Table 4: Example Classification Errors: S=System, G=Gold, +=positive, -=negative, /=neutral. Bold-faced text indicates the span for contextual polarities.

evaluation is based on the average positive and negative F-score. Task B results are listed in table 2, and task A results are shown in table 3. For comparison these tables also include the top-ranked system in each category as well as the mean scores across all submissions.

## 9 Error Analysis

To better understand our systems' limitations we manually inspected misclassified output. Table 4 lists errors representative of the common issues uncovered in our error analysis.

Though some degree of noise is expected in sentiment analysis, we found several instances of annotation error or ambiguity where it could be argued that the system was actually correct. The message in #1 was annotated as neutral, whereas the presence of the word "yay" suggests an overall positive polarity. The text in #2 could be interpreted as positive, negative or neutral depending on the author's disposition.

Unseen vocabulary and unexpected usages were the largest category of error. For example in #3 "crashed" means to attend without an invitation instead of the more negative meaning associated with car accidents and airplane failures. Although POS features can disambiguate word senses, in this case more sophisticated features for word sense disambiguation could help. While the degradation in performance between the Tweet and SMS test sets might be explained by differences in medium, errors like those found in #4 and #5 suggest that this may have more to do with the dialectal differences between the predominantly American and British English found in the Tweet test set and the Colloquial Singaporean English (aka Singlish) found in the SMS test set. Error #6 illustrates both how hashtags composed of common words can easily become

a problem when assigning a polarity to a short context. Hashtag segmentation presents one possible path to reducing this source of error.

## 10 Conclusions and Future Work

The results and rankings reported in section 8 suggest that our systems were competitive in assigning sentiment across the varied tasks and data conditions. We performed particularly well in disambiguating contextual polarities finishing second overall on the Tweet test set. We hypothesize this performance is largely due to the expanded vocabulary obtained via unlabeled data and the richer syntactic context captured with dependency path representations.

Looking forward, we expect that term recall and unseen vocabulary will continue to be key challenges for sentiment analysis on social media. While larger amounts of data should assist in that pursuit, we would like to explore how a more iterative approach to self-training and lexicon expansion may provide a less noisy path to attaining such recall.

## 11 Acknowledgments

We would like to thank the organizers of SemEval 2013 and the Sentiment Analysis in Twitter task for their time and energy. We also would like to express our appreciation to the anonymous reviewers for their helpful feedback and suggestions.

## References

- Apoorv Agarwal, Boyi Xie, Iliia Vovsha, Owen Rambow, and Rebecca Passonneau. 2011. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Language in Social Media (LSM 2011)*.
- Luciano Barbosa and Junlan Feng. 2010. Robust sentiment detection on twitter from biased and noisy data.

- In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 36–44, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Jinho D. Choi and Andrew McCallum. 2013. Transition-based dependency parsing with selectional branching. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL'13)*.
- Sanjiv Das and Mike Chen. 2001. Yahoo! for amazon: extracting market sentiment from stock message boards. In *Proceedings of the 8th Asia Pacific Finance Association Annual Conference*.
- Dmitry Davidov, Oren Tsur, and Ari Rappaport. 2010a. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*.
- Dmitry Davidov, Oren Tsur, and Ari Rappaport. 2010b. Enhanced sentiment learning using twitter hashtags and smileys. In *Coling 2010*, pages 241–249.
- Marie-Catherine de Marneffe and Christopher D. Manning, 2012. *Stanford typed dependencies manual*. Stanford University, v2.0.4 edition, November.
- Andrea Esuli and Fabrizio Sebastiani. 2006. SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining. In *Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06)*.
- Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research*, 9:1871–1874.
- Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 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 ACL:HLT 2011*.
- Vasileios Hatzivassiloglou and Kathleen R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL 1997)*.
- Yulan He and Deyu Zhou. 2011. Self-training from labeled features for sentiment analysis. *Information Processing and Management*, 47(4):606–616.
- Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. 2011. Twitter Sentiment Analysis: The Good the Bad and the OMG! In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM 2011)*.
- Saif M. Mohammad and Peter D. Turney. 2011. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 59(000).
- Saif M. Mohammad. 2012. #emotional tweets. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (\*SEM)*.
- Philip V. Ogren, Philipp G. Wetzler, and Steven Bethard. 2008. ClearTK: A UIMA toolkit for statistical natural language processing. In *Towards Enhanced Interoperability for Large HLT Systems: UIMA for NLP workshop at Language Resources and Evaluation Conference (LREC '08)*, 5.
- Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*.
- Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.
- Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment Classification Using Machine Learning Techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*.
- Delip Rao and Deepak Ravichandran. 2009. Semi-supervised polarity lexicon induction. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*.
- H. J. Scudder. 1965. Probability of error of some adaptive pattern-recognition machine. *IEEE Transactions on Information Theory*, 11:363–371.
- Vikas Sindhwani and Prem Melville. 2008. Document-word co-regularization for semi-supervised sentiment analysis. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, ICDM '08*, pages 1025–1030.
- Oscar Täckström and Ryan McDonald. 2011. Semi-supervised latent variable models for sentence-level sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011)*.
- Andranik Tumasjan, Timm O. Sprenger, Philipp G. Sandner, and Isabell M. Weppe. 2010. Predicting elections with twitter what 140 characters reveal about political sentiment. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media (ICWSM 2010)*.
- Peter Turney. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual*

- Meeting of the Association for Computational Linguistics (ACL 2002).*
- Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39:165–210.
- Theresa Wilson, Janyce Wiebe, and Paul Hoffman. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*.
- Theresa Wilson, Zornitsa Kozareva, Preslav Nakov, Alan Ritter, Sara Rosenthal, and Veselin Stoyanov. 2013. SemEval-2013 task 2: Sentiment analysis in twitter. In *Proceedings of the 7th International Workshop on Semantic Evaluation*. Association for Computation Linguistics.
- Xiuzhen Zhang, Yun Zhou, James Bailey, and Kotagiri Ramamohanarao. 2012. Sentiment analysis by augmenting expectation maximisation with lexical knowledge. *Proceedings of the 13th International Conference on Web Information Systems Engineering (WISE2012)*.