

SentiSys at SemEval-2016 Task 4: Feature-Based System for Sentiment Analysis in Twitter

Hussam Hamdan

Aix-Marseille University
hamdan.hussam@gmail.com

Abstract

This paper describes our sentiment analysis system which has been built for Sentiment Analysis in Twitter Task of SemEval-2016. We have used a Logistic Regression classifier with different groups of features. This system is an improvement to our previous system Lsislif in Semeval-2015 after removing some features and adding new features extracted from a new automatic constructed sentiment lexicon.

1 Introduction

Sentiment analysis in Twitter is different from document level sentiment analysis. Normally, in document level, each document is classified as positive or negative, the document is long enough to obtain a good representation using only the existing words (bag-of-words). For example, in movie reviews we can get f-score of 85% using bag-of-words representation with SVM classifier while in Twitter it is about 60% according to our experiments in previous SemEval workshops. This lower performance in Twitter domain is not surprising if we know the limitations of such task when applied to Twitter:

- The size of a tweet is limited to 140 characters which leads to sparseness where the tweets do not provide enough word co-occurrence.
- The informal language and non-standard expressions.
- The numerous spelling errors.

For dealing with the previous limitations, we have decided to extend the bag-of-words representation. Therefore, many group of features have been extracted. Uni-gram, bi-gram and 3-grams of words features to capture the text of tweet and the context. Negation features to handle the negated context. Sentiment lexicons features can help the classification because it contains positive and negative words which can add a useful information about the polarity of a tweet, they also contain a lot of terms which may not appear in the training data which can be very useful. Semantic features as Brown clusters can also give a rich representation which can be useful for reducing the sparsity.

For evaluating our system, we have participated in SemEval-2016 competition for sentiment analysis in Twitter (message polarity subtask A)¹ (Nakov et al., 2016). Our system has been ranked six over 34, this system is derived from our previous system LsisLif which has been ranked third in SemEval-2015.

The rest of this chapter is organized as follows. Section 2 presents the problem formulation. Section 3 gives an overview of our proposed approach. The features we extracted for training the classifier are presented in Section 4. Our experiments are described in Section 5. The related work is presented in Section 6. The conclusion and future work are presented in Section 7.

2 Problem Formulation

Let $T = t_1, t_2, \dots, t_n$ be a collection of n tweets. Each tweet t_i will be represented by a subset of all

¹<http://alt.qcri.org/semEval2016/task4/>

possible features $F = f_1, f_2, \dots, f_m$ that can appear in t_i . The features can be single words, bigrams, n-grams, stemmed words or other syntactic or semantic features. If a feature f_i exists in a tweet t_j , the tweet can be represented as a vector of weighted features $t_j = (w_1, w_2, \dots, w_m)$ where w_i is the weight of the feature f_i in the tweet t_j . w_i can represent the presence or absence of the feature or the frequency or any other function of the feature frequency in the tweet.

Let us have three classes $C = c_1, c_2, c_3$ where c_1 represents the negative class, c_2 the neutral class and c_3 the positive class. Our task is to assign each tweet t_j to a class c_i .

3 Overview of the Proposed Approach

Our proposed approach for sentiment polarity classification consists of three steps:

1. We tokenize each tweet to get the feature space which contains the words, punctuations and emoticons that appear in the tweets.
2. We extend the feature space by extracting some features using different resources (Sentiment lexicons, Twitter dictionary) and some semantic features.
3. We train a supervised classifier to get a trained model in order to predict the sentiment of the new tweets.

The next section describes the features we have extracted.

4 Feature Extraction

Before extracting the features, we should tokenize the tweet. Tokenization is a challenging problem for Twitter text. Happytokenizer² is the tokenizer which we used. It can capture the words, emoticons and punctuations. For example, for this tweet:

”RT @ #happyfuncoding: this is a typical Twitter tweet :-)”

It returns the following terms:

{rt, @, #happyfuncoding, :, this, is, a, typical, twitter, tweet, :-)}

²<http://sentiment.christopherpotts.net/tokenizing.html>

We also replaced each web link by the word *url* and each user name by *user*. Then, several groups of features have been extracted to improve the bag-of-words representation.

4.1 Word ngrams

Unigram, bigram and 3-gram are extracted for each term in the tweet without any stemming or stop-word removing, all terms with occurrence less than 3 are removed from the feature space. Therefore, for this tweet:

”i’am going to chapel hill on sat. :)”

The feature vector produced by this group of feature will be:

{’i’m’, ’going’, ’to’, ’chapel’, ’hill’, ’on’, ’sat’, ’.’, ’:), ’i’m going’, ’going to’, ’to chapel’, ’chapel hill’, ’hill on’, ’on sat’, ’sat .’, ’. :), ’i’m going to’, ’going to chapel’, ’to chapel hill’, ’chapel hill on’, ’hill on sat’, ’on sat .’, ’sat . :)’}

4.2 Negation Features

The rule-based algorithm presented in Christopher Potts’ Sentiment Symposium Tutorial³ is implemented. This algorithm appends a negation suffix to all words that appear within a negation scope which is determined by a negation key and a punctuation or a connector belonging to [’’, ’;’, ’:’, ’!’, ’?’’, ’but’, ’—’, ’so’]. All the negated words are added to the feature space. For example, for this tweet:

”I am not happy”

The feature vector generated by the words n-gram features with negation features is:

{’i’am’, ’not’, ’happy_Neg’, ’happy’, ’i’am not’, ’not happy’, ’i’am not happy’}

happy_NEG is added by the negation features while the others are the ngrams features. Obviously, we have chosen to add the negated feature to the vector without removing the original feature *happy*.

4.3 Twitter Dictionary

We constructed a dictionary for the abbreviations and the slang words used in Twitter in order to overcome the ambiguity of these terms which may increase the similarity between two similar tweets written in two different ways. This dictionary maps

³<http://sentiment.christopherpotts.net/lingstruc.html>

certain Twitter expressions and emotion icons to their meaning or their corresponding sentiment. It contains about 125 terms collected from different pages on the Web. Table 1 shows a part of the dictionary.

Twitter expression	Meaning
:)	veryhappy
:)	veryhappy
b/c	Because
FWIW	For what it's worth
Gr8	Great
IMHO	In my honest opinion or in my humble opinion
J/K	Just kidding
LOL	Laughing out loud funny
OMG	Oh my God
PLZ	Please
ROFL	Rolling on the floor laughing
RTHX	Thanks for the retweet
hahaha	laughing funny
wow	amazing surprised

Table 1: A part of Twitter dictionary.

All terms presented in a tweet and in the Twitter dictionary are mapped to their corresponding terms in the dictionary and added to the feature space. For this tweet: "i'am going to chapel hill on sat. :)", the term *veryhappy* will be added to the tweet vector because the emoticon :) will be replaced by *veryhappy* as indicated in the dictionary.

4.4 Semantic Features

The semantic representation of a text may bring some important hidden information, which may result in a better document representation and a better classification system. Usually, the semantic features can help to overcome the problem of sparseness in short text. Externally resources may be important to get such representation.

4.4.1 Brown Dictionary Features

From over 56 million English tweets (837 million tokens), 1000 hierarchical clusters have been constructed over 217 thousand words (Owoputi et al., 2013). Table 2 shows an example of five clusters.

Cluster	Top words (by frequency)
A1	lmao lmfao lmaoo lmaooo hahahahaha lol cfu rofl loool lmfaoo lmfaooo lmaoooo lmbolololol
A2	haha hahaha hehe hahahaha hahah aha hehehe ahaha hah hahahah kk hahaa ahah
A3	yes yep yup nope yess yesss yessss ofcourse yeap likewise ycpp yesh yw yuup yus
A4	yeah yea nah naw yeahh nooo yeh noo noooo yeaa ikr nvm yeahhh nahh nooooo
A5	smh jk #fail #random #fact smfh #smh #winning #realtalk smdh #dead #justsaying

Table 2: Example Twitter word clusters: we list the most probable words, starting with the most probable, in descending order.

Note that in cluster A1, the term lololol (an extension of lol for "laughing out loud") is grouped with a large number of laughter acronyms.

Each word in the text is mapped to its cluster in Brown dictionary, 1000 features are added to feature space where each feature represents the number of words in the text belonging to each cluster.

4.5 Sentiment Lexicons

The system extracts four features from the manual constructed lexicons and six features from the automatic ones. For each sentence the number of positive words, the number of negative ones, the number of positive words divided by the number of negative ones and the polarity of the last word are extracted from manual constructed lexicons. In addition to the sum of the positive scores and the sum of the negative scores from the automatic constructed lexicons.

The manual lexicons are: MPQA Subjectivity Lexicon⁴ and Bing Liu Lexicon⁵. The automatic ones are: NRC Hashtag Sentiment Lexicon and our lexicon based on natural entropy measure (Hamdan et al., 2015c).

Thus, this feature group adds 20 features to the tweet vector, some of this features are integer numbers others are floats. The lexicons which we used are the following:

4.5.1 Manually Constructed Sentiment Lexicons

Two manual constructed lexicons have been exploited:

1. MPQA Subjectivity Lexicon

Multi-Perspective Question Answering Subjectivity Lexicon is maintained by (Wilson et al., 2005), a lexicon of over 8,000 subjectivity single-word clues, each clue is classified as positive or negative. This is a fragment illustrating this lexicon structure:

Strength	Length	Word	Part-of-speech	Stemmed	priorpolarity
weaksbj	1	abandoned	adj	n	negative
weaksbj	1	abandonment	noun	n	negative
weaksbj	1	abandon	verb	y	negative

2. Bing Liu Lexicon

A list of positive and negative opinion words or sentiment words for English (around 6800

⁴http://mpqa.cs.pitt.edu/lexicons/subj_lexicon/

⁵<http://www.cs.uic.edu/liub/FBS/sentiment-analysis.html#lexicon>

words). This list was compiled over many years starting from this paper (Hu and Liu, 2004a). These are the first three words in each class of this lexicon:

Positive	Negative
abound	abnormal
abounds	abolish
abundance	abominable

4.5.2 Automatic Constructed Sentiment Lexicons

Three automatic constructed lexicon have been exploited:

1. NRC Hashtag Sentiment Lexicon

NRC Hashtag Sentiment Lexicon (Mohammad, 2012) contains tweet terms with scores, positive score indicates association with positive sentiment, whereas negative score indicates association with negative sentiment. It has entries for 54,129 unigrams and 316,531 bigrams; the scores were computed using PMI over corpus of tweets. Here are the first and the last three lines of the unigrams NRC lexicon file:

Term	Score	#positive	#negative
#fabulous	7.526	2301	2
#excellent	7.247	2612	3
#superb	7.199	1660	2
ipad2	-6.615	1	1205
#dreadful	-6.764	1	1398
#unacceptable	-6.925	2	3284

Score is a real number indicates the sentiment score. *#positive* is the number of times the term co-occurred with a positive marker such as a positive emoticon or a positive hashtag. *#negative* is the number of times the term co-occurred with a negative marker such as a negative emoticon or a negative hashtag.

4.6 Our Sentiment Lexicon

PMI metric has been widely used to compute the semantic orientation of words in order to construct the automatic lexicons. Sentiment140 lexicon is constructed using semantic orientation on Sentiment140 corpus (Go et al., 2009), a collection of 1.6 million

tweets that contain positive and negative emoticons⁶. But this corpus is a balanced corpus, it contains the same number of positive and negative tweets. Therefore, semantic orientation can be rewritten as following:

$$SO(w) = PMI(w, +) - PMI(w, -) = \log\left(\frac{p(w,+)}{p(w).p(+)}\right) - \log\left(\frac{p(w,-)}{p(w).p(-)}\right) \quad (1)$$

As $p(+)=p(-)=0.5$ in the balanced corpus:

$$So(w) = 1 + \log(p(+|w)) - 1 - \log(p(-|w)) = \log(a/c) \quad (2)$$

where + stands for the positive class, - stands for negative class, a is the number of documents containing the word w in the positive class, c is the number of documents containing the word w in the negative class. Thus, the semantic orientation is positive if $a>c$ else it is negative. We should note that the probability of the classes does not affect the final semantic orientation score, therefore we propose another metric which depends on the distribution of the word over the classes which seems more relevant in the balanced corpus.

We constructed a lexicon from sentiment140 corpus, we calculated Natural Entropy (ne) score for each term in this manner:

$$ne(w) = 1 - (-(p(+|w).log(p(+|w)) - p(-|w).log(p(-|w)))) \quad (3)$$

where

$p(+|w)$: The probability of the positive class given the word w .

$p(-|w)$: The probability of the negative class given the word w .

The more uneven the distribution of documents where a term occurs, the larger the Natural Entropy of this term is. Thus, the entropy of the term can express the uncertainty of the classes given the term. One minus this degree of uncertainty boosts the terms that unevenly distributed between the two classes (Wu and Gu, 2014). ne score is always between 0 and 1, and it assigns a high score for the words unevenly distributed over the classes, but it cannot discriminate the positive words from the negative ones. Therefore, we have used the a and c for discriminating the positive words from the negative ones; if $a>c$ then the word is considered positive else it is considered negative.

⁶<http://help.sentiment140.com/for-students>

Using this lexicon instead of sentiment140 can improve the performance of a state-of-the-art sentiment classifier as shown in (Hamdan et al., 2015c).

5 Experiments and Results

5.1 Twitter Dataset

Twitter datasets have been provided by SemEval organizers since 2013 for message polarity classification subtask of sentiment analysis in Twitter (Nakov et al., 2013). The participants have been provided with training tweets annotated positive, negative or neutral. In addition to a script for downloading the tweets. After executing the given script, we got the whole training dataset which consists of 9684 tweets. The organizers have also provided a development set containing 1654 tweets for tuning a machine learner. Table 3 shows the distribution of each label in each dataset.

Data	All	Positive	Negative	Neutral
train	9684	3640	1458	4586
dev	1654	739	340	575
test-2016	-	-	-	-

Table 3: Sentiment labels distribution in the training, testing and development datasets in Twitter.

5.2 Experiment Setup

We trained the L1-regularized logistic regression classifier implemented in LIBLINEAR (Fan et al., 2008), we had also tested L2 regularization technique but it gives less performance than L1. The classifier is trained on the training dataset using the features in the previous section with the three polarities (positive, negative, and neutral) as labels. A weighting schema is adapted for each class, we use the weighting option $-w_i$ which enables a use of different cost parameter C for different classes. Since the training data is unbalanced, this weighting schema adjusts the probability of each label. Thus, we tuned the classifier in adjusting the cost parameter C of logistic regression, weight w_{pos} of positive class and weight w_{neg} of negative class. We used the development set for tuning the three parameters, all combinations of C in range [0.1 .. 4] by step of 0.1, w_{pos} in range [1 .. 8] by step of 0.1, w_{neg} in range [1 .. 8] by step of 0.1 are tested. The combination

$C=0.3$, $w_{pos}=7.6$, $w_{neg}=5.2$ have given the best F1-score for the development set and therefore it was selected for our experiments on test set 2016.

5.3 Results

The evaluation score used by the task organizers was the averaged F1-score of the positive and negative classes. In the SemEval-2016 competition, our submission is ranked six (59.8%) over 34 submissions while it was ranked third in SemEval-2015.

Table 4 shows the results of our experiments after removing a feature group at each run for the four test set 2016.

Run	Test-2016
All features	59.8
all-lexicons	56.9
all-ngram	58.1
all-brown	58.4

Table 4: The F1 score for each run, All features run exploits all features while the others remove a feature group at each run lexicons, n-gram and brown cluster, respectively.

The results show that the sentiment lexicons features are the most important ones which conforms with the conclusion in different studies (Hamdan et al., 2015c; Mohammad et al., 2013).

6 Related Work

There are two principally different approaches to opinion mining: lexicon-based and supervised. The lexicon-based approach goes from the word level in order to constitute the polarity of the text. This approach depends on a sentiment lexicon to get the word polarity score. While the supervised approach goes from the text level and learn a model which assigns a polarity score to the whole text, this approach needs a labeled corpus to learn the model.

6.1 Lexicon-Based Approach

Lexicon-based approaches decide the polarity of a document based on sentiment lexicons. The sentiment of a text is a function of the common words between the text and the sentiment lexicons.

Much of the first lexicon-based research has focused on using adjectives as indicators of the seman-

tic orientation of text (Hatzivassiloglou and McKeown, 1997; Hu and Liu, 2004b). (Taboada et al., 2011) proposed another function called SO-CAL (Semantic Orientation CALculator) which uses dictionaries of words annotated with their semantic orientation (polarity and strength), and incorporates intensification and negation.

Thus, the sentiment lexicon is the most important part of this approach. Three different ways can be used to construct such lexicons: Manual Approach, Dictionary-Based Approach and Corpus-Based Approach.

6.2 Supervised Approach

The supervised approach is a machine learning approach. Sentiment classification can be seen as a text classification problem (Pang et al., 2002; Liu, 2012).

The research papers in sentiment classification have mainly focused on the two steps: document representation and classification methods.

While some papers have extended the bag-of-word representation by adding different types of features (Pang et al., 2002; Mohammad et al., 2013; Hamdan et al., 2013; Hamdan et al., 2015c), others have proposed different weighting schemas to weight the features such as PMI, Information Gain and chi-square χ^2 (Martineau and Finin, 2009; Paltoglou and Thelwall, 2010; Deng et al., 2014). Recently, after the success of deep learning techniques in many classification systems, several studies have learned the features instead of extracting them (Socher et al., 2013; Severyn and Moschitti, 2015).

The work of (Pang et al., 2002) was the first to apply this approach to classify the movie reviews into two classes positive or negative. They tested several classifiers (Naive Bayes, SVM, Maximum entropy) with several features.

Later on, many studies have proposed different features and some feature selection methods to choose the best feature set. Many features have been exploited :

- Terms and their weights: The features are the unigrams or n-grams with the associated frequency or weight given by a weighting schema like TF-IDF or PMI.
- Part of Speech (POS): The words can indicate

different sentiment according to their parts of speech (POS). Some papers treated the adjectives as special features.

- Sentiment Lexicons: The words and expressions which express an opinion have been used to add additional features as the number of positive and negative terms.
- Sentiment Shifters: The terms that are used to change the sentiment orientation, from positive to negative or vice versa such as *not* and *never*. Taking into account these features can improve the sentiment classification.
- Semantic Features: The named entities, concepts and topics have been extracted to get the semantic of the text.

Many systems which have worked on feature extraction have achieved a state-of-the-art performance in many competitions like SemEval⁷. For example, (Mohammad et al., 2013) used SVM model with several types of features including terms, POS and sentiment lexicons in Twitter data set. (Hamdan et al., 2015a; Hamdan et al., 2015c; Hamdan et al., 2015b) have also proved the importance of feature extraction with logistic regression classifier in Twitter and reviews of restaurants and laptops. They extracted terms, sentiment lexicon and some semantic features like topics. And (Hamdan et al., 2013) has proposed to extract the concepts from DBpedia.

Recently, some research papers have applied deep learning techniques to sentiment classification. (Socher et al., 2013) proposed to use recursive neural network to capture the compositionality in the phrases, (Tang et al., 2014) combined the hand-crafted features with learned features. They used neural network for learning sentiment-specific word embedding, then they combined hand-crafted features with these word embedding to produce a state-of-the-art system in sentiment analysis in Twitter. (Kim, 2014) proposed a simple convolutional neural network with one layer of convolution which performs remarkably well. Their results add to the well-established evidence that unsupervised pre-training of word vectors is an important ingredient in deep learning for Natural language processing.

⁷<https://www.cs.york.ac.uk/semeval-2013/task2.html>

7 Conclusion and Future Work

In this paper, we tested the impact of combining several groups of features on the sentiment classification of tweets. A logistic regression classifier with weighting schema was used, the sentiment lexicon-based features seem to get the most influential effect with the combination.

As the sentiment lexicons features seem to be so important in sentiment classification, we think that it is important to orient our future work on this direction. Improving the automatic construction of sentiment lexicons may lead to an important improvement on sentiment classification. For example, taking the context in the consideration may help such process. Another important direction is using deep learning techniques which have recently proved their performance in several studies. Thus, we can learn the features instead of extracting them.

References

- Zhi-Hong Deng, Kun-Hu Luo, and Hong-Liang Yu. 2014. A study of supervised term weighting scheme for sentiment analysis. *Expert Systems with Applications*, 41(7):3506 – 3513.
- 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.
- Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter Sentiment Classification using Distant Supervision. *Processing*, pages 1–6.
- Hussam Hamdan, Frédéric Bechet, and Patrice Bellot. 2013. Experiments with DBpedia, WordNet and SentiWordNet as resources for sentiment analysis in micro-blogging. In *International Workshop on Semantic Evaluation SemEval-2013 (NAACL Workshop)*, Atlanta, Georgia (USA), April.
- Hussam Hamdan, Patrice Bellot, and Frédéric Bechet. 2015a. Lsislif: CRF and Logistic Regression for Opinion Target Extraction and Sentiment Polarity Analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 753–758, Denver, Colorado, June. Association for Computational Linguistics.
- Hussam Hamdan, Patrice Bellot, and Frédéric Bechet. 2015b. Lsislif: Feature Extraction and Label Weighting for Sentiment Analysis in Twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 568–573, Denver, Colorado, June. Association for Computational Linguistics.
- Hussam Hamdan, Patrice Bellot, and Frédéric Bechet. 2015c. Sentiment Lexicon-Based Features for Sentiment Analysis in Short Text. In *Proceeding of the 16th International Conference on Intelligent Text Processing and Computational Linguistics*, Cairo, Egypt.
- Vasileios Hatzivassiloglou and Kathleen R. McKeown. 1997. Predicting the Semantic Orientation of Adjectives. In *Proceedings of the Eighth Conference on European Chapter of the Association for Computational Linguistics*, EACL '97, pages 174–181, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Minqing Hu and Bing Liu. 2004a. Mining and Summarizing Customer Reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA. ACM.
- Minqing Hu and Bing Liu. 2004b. Mining and Summarizing Customer Reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA. ACM.
- Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. *CoRR*, abs/1408.5882.
- Bing Liu. 2012. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.
- Justin Martineau and Tim Finin. 2009. Delta TFIDF: An Improved Feature Space for Sentiment Analysis. In *ICWSM*.
- Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. NRCCanada: Building the State-of-the-Art in Sentiment Analysis of Tweets. In *Proceedings of the International Workshop on Semantic Evaluation, SemEval '13*.
- Saif Mohammad. 2012. #Emotional Tweets. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 246–255, Montréal, Canada, June. Association for Computational Linguistics.
- Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. SemEval-2013 Task 2: Sentiment Analysis in Twitter. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 312–320, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

- Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. 2016. SemEval-2016 Task 4: Sentiment Analysis in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '2016. Association for Computational Linguistics, June.
- Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of NAACL-HLT*, pages 380–390.
- Georgios Paltoglou and Mike Thelwall. 2010. A Study of Information Retrieval Weighting Schemes for Sentiment Analysis. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1386–1395, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs Up?: Sentiment Classification Using Machine Learning Techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 79–86, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Aliaksei Severyn and Alessandro Moschitti. 2015. UNITN: Training Deep Convolutional Neural Network for Twitter Sentiment Classification. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 464–469, Denver, Colorado, June. Association for Computational Linguistics.
- Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *EMNLP*.
- Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based Methods for Sentiment Analysis. *Comput. Linguist.*, 37(2):267–307, June.
- Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1555–1565, Baltimore, Maryland, June. Association for Computational Linguistics.
- Theresa Wilson, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005. OpinionFinder: A System for Subjectivity Analysis. In *Proceedings of HLT/EMNLP on Interactive Demonstrations*, HLT-Demo '05, pages 34–35, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Haibing Wu and Xiaodong Gu. 2014. Reducing Over-Weighting in Supervised Term Weighting for Sentiment Analysis. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 1322–1330.