Extending the *Loughran and McDonald Financial Sentiment Words List* from 10-K Corporate Fillings using Social Media Texts

Marcelo Sardelich, Dimitar Kazakov
University of York
Heslington, York YO10 5GH, UK
{marcelo.sardelich, dimitar.kazakov}@york.ac.uk

Abstract

This article describes a novel text corpora and sentiment lexicon for financial text mining. The text corpora comprises social media messages, specifically, comments on stocks by *Yahoo Message Board* service users. The messages contains the user opinion and is labelled by the user with an overall sentiment label. This novel dataset with 74,641 messages covering 492 stocks over a period of two years is made publicly available. State-of-the-art methods are used to extract terms that convey positive and negative connotation from each message of the corpora. Then, each message is represented as a vector of these terms and sentiment classifiers are trained. The best combination of text representation weights and classifier model achieves 91.4% accuracy in the test set. We then use this sentiment classifier to build a sentiment lexicon, which contains words associated with positive and negative sentiments. We show that this lexicon is useful to extend previously proposed words lists, which were manually crafted from 10-K or 10-Q financial documents, and is able to capture the sentiment of terms from the formal and informal language of financial stock markets. Our novel financial domain text corpora and sentiment lexicon constitute valuable language resources to help advance the work on financial narrative processing.

Keywords: financial texts corpora, domain-specific sentiment lexicon, supervised learning, neural networks

1. Introduction

The sentiment classification of texts is a Natural Language Processing task that has increasingly attracted attention of the research community in recent years. Broadly speaking, we can group the sentiment classification into two approaches: On the one hand, those employing supervised (Pang et al., 2002; Pang and Lee, 2008; Glorot et al., 2011; Socher et al., 2013) or semi-supervised machine learning methods (Dasgupta and Ng, 2009; Zhou et al., 2013; Ponomareva and Thelwall, 2013), and on the other hand, those using unsupervised learning (Turney, 2002).

Lexicon-based sentiment classification is performed by retrieving information from *sentiment word lists* or *sentiment lexicon*, i.e. a database of words with positive and negative annotations. The main challenge of this approach is to compile the word list while avoiding any time-consuming human intervention. In other words, the goal would be to learn the sentiment words lists rather than compiling it manually. The techniques developed to build a sentiment word lists can be arranged in three broad categories: *Dictionary-based*, *Corpus-based* and *Emoticon-based*. The former method starts with a seed of initial words that contains at least one positive and one negative word. Then, the seed is bootstrapped, e.g. using WordNet (Miller, 1995) *synsets* as in Hu and Liu (2004);Hassan and Radev (2010);Rao and Ravichandran (2009). The *Corpus-based* technique is similar to the *Dictionary-based* one, however, it attempts to bootstrap the seed using a domain specific corpus. This method largely exploits *grammatical coherences*[1] of a given language (see, for example, the early studies in Hatzivassiloglou and McKeown (1997) and posterior advancements in Kanayama and Nasukawa (2006)). One of the main drawbacks of this method is the limited occurrence of linguistic conventions in a given corpus. Finally, the *Emoticon-based* methods are grounded on the fact that Emotion icons (*Emoticons*), such as `-)`, `:)` and `:-(` have an advantage of summarizing feelings. Therefore, they are useful to automatically assign a sentiment label to a given text. This method is employed in Go et al. (2009) and in Davies and Ghahramani (2011).

As a matter of fact, many publicly available language resources for sentiment classification, e.g. *Sentiment140* (Go et al., 2009), *Bing Liu Sentiment Lexicon* (Liu, 2012), *MPQA Sentiment Lexicon* (Wilson et al., 2005), *Harvard Dictionary* (the *General Inquirer*) (Stone et al., 1966) and *VADER* (Hutto and Gilbert, 2014), are built based on three fundamental methods discussed above, named *Dictionary-based*, *Corpus-based* or *Emoticon-based*.

Although these resources are effective for sentiment classification in the general contexts of customer reviews, they are of limited use for the financial domain corpora, such as US 10-K/10-Q corporate fillings, conference press releases or social media content related to stock markets. For instance, as stressed in Loughran and Mcdonald (2011): "Almost three-fourths of the words identified as negative by the widely used *Harvard Dictionary* are words typically not considered negative in financial contexts."

This work focuses on building a sentiment lexicon specific for texts from the financial domain. Three main contributions are made to the existing literature. First, we propose a novel sentiment lexicon for words in financial contexts. This sentiment lexicon is learnt from user posts of the *Yahoo Message Board* applying a supervised learning approach. In this regard, our work is helpful to extend

[1] Grammatical coherence can be understood as linguistic conventions on connectives such as *and*, *or* and *neither nor*. To illustrate, from the text "*The service is good and staff is friendly*" we could infer that "friendly" and "good" have the same sentiment connotation even without knowing *a priori* the sentiment of each word individually.

the manually annotated *Loughran and McDonald Financial Sentiment Dictionary* of Loughran and Mcdonald (2011). Second, the method we propose to build a sentiment lexicon from a text sentiment classifier can be utilized as a general method to similar problems, regardless the corpus domain. Third, we make the sentiment annotated dataset used to build the sentiment lexicon publicly available as an additional language resource.

2. Financial Domain Dataset

2.1. Description and Characteristics

Until *Yahoo*'s recent acquisition by Verizon, the company provided a financial message board service covering a broad range of individual stocks. When discussing a given stock, users could annotate their posts with one of the following fixed five sentiment labels: *Buy*, *Strong Buy*, *Sell*, *Strong Sell* and *Hold*.

Aiming to make use of this sentiment annotation, we collected raw HTML content from each stock message board. Then, we parsed this content extracting tags that contain relevant information. Finally, we converted the parsed HTML content into open JSON (JavaScript Object Notation) format. This step converted the unstructured message board content (HTML) into structured data (JSON).

In total, we collected 4.9GB of Python serialized JSON objects by sending web requests through 8 parallel processes during two consecutive weeks[2]. Messages published in 2014 and 2015 were collected for a list of 492 stocks[3]. Below, we show two samples from the JSON dataset for *IBM* and *Exxon Mobil* stocks (the field `message_sentiment` describes the label):

```
{'is_reply': True,
 'message_sentiment': 'Buy',
 'message_title': 'IBM profit machine
     slows; layoffs planned',
 'timestamp': 1366340436.652},

{'is_reply': False,
 'message_sentiment': 'Strong Sell',
 'message_title': "Bloomberg: Crude
     Oil Erases Advance on OPEC's
     Reduced Demand Forecast",
 'timestamp': 1421371595.252},
```

We aggregate the messages of each stock into three classes. In the **POS** (positive) class, we group all messages originally labelled as *Buy*, *Strong Buy*. The **NEG** (negative) class receives all *Sell*, *Strong Sell* messages. Finally, all residual messages are assigned to the class **NEUTRAL**.

Our strategy to collapse the messages into the coarse-grained classes **POS** and **NEG**, regardless whether it is a *Strong* message or not, is grounded on the fact that without the aggregation each class would be underrepresented and few labels would be left to discriminate, for example, between *Sell* and *Strong Sell*. The same follows to the *Hold* messages, i.e. few samples would be left to discriminate this specific class.

One interesting characteristic of the users' behaviour is their general optimism regrading the stock market. Based on the distribution of **POS**, **NEG** and **NEUTRAL** labels of Table 1, we see that our data has a strong bias towards messages with positive tone. We perceive this bias as a behavioural manifestation of the overconfidence and excessive optimism investors see in the stock markets, as described in Shiller (2000).

Label	Number of Samples	Percentage
POS	46,981	63
NEG	20,610	28
NEUTRAL	7,050	9
total	74,641	100

Table 1: Dataset sentiment labels distribution.

Finally, a closer look at some random samples reveals a certain degree of noise in the annotated labels. For example, the text: "*All the cards on the table today!*" is labelled by the user as **POS**. However, without the label most annotators would probably consider the message neutral. We presume that this "labelling mismatch" happens because some message board users tend to mistaken the message true connotation for their own judgment about the future performance of the company. That being said, potentially, the user that posted this message was betting the market would go up and not exactly the fact that, from a linguistic viewpoint, "*all cards on the table*" is an utterance with neutral sentiment.

2.2. Pre-Processing and Wrangling

Our pre-processing phase starts by filtering out all messages with the following characteristics: duplicate title, without any sentiment annotation and reply messages.

After this phase, we end up with $74,641$ messages that are dumped separately to a JSON file for each stock ticker.

Before training the model described in Sec. 3. we carry out the following additional pre-processing steps:

1. A simple lexical normalization to convert Out-Of-Vocabulary (OOV) words to its canonical form. The normalization treats the following cases: Repeated words (e.g. convert from "going up up up" to "going up"). Repeated symbols (e.g. convert from "AMAZING!!!!!" to "AMAZING!"). This task is pipelined and executed before the Part-Of-Speech (POS) tagging task.

2. Spell checking using *GNU Aspell*. The words that are still not recognized are filtered out.

3. We ignore terms that appears in less than three message titles.

[2] We relied on data parallelization techniques where each process/thread took care of one stock independently.

[3] The list of stocks was compiled based on all constituents of the Standard & Poor's 500 Index (S&P500) as in 2017. Subsequently, stocks with no messages were disregarded, hence, reducing the initial universe of 500 stocks.

3. Methodology

3.1. Document Representation

We use a sparse vector space model to represent each message of our dataset. However, we expand each message in a base of Semantic Orientation[4] (SO) occurring in it, rather than the standard "bag of words" model, according to which the message is represented by the set of words or *n-grams* occurrences (weighted or not). Our sentiment classifier is thus trained on what might be called "bag of Semantic Orientation (SO)".
The motivation to use the "bag of Semantic Orientation (SO)" representation resides in two facts:

1. The SO keywords are Part-of-Speech (POS) tag patterns that work as good indicators of explicit opinions. For instance, the tag pattern `JJ` (adjective) + `NNS` (noun) + `<any tag>` extracts "economic concerns" from the message "*Stocks tank on global economic concerns*". Since we represent each text of our corpora in this SO base, we can, at inference time, predict the sentiment of each SO keyword separately in order to build our sentiment lexicon.

2. The SO tag patterns are handy to disregard messages that do not convey any connotation and appropriate in our context of binary sentiment classification, i.e. only **POS** or **NEG** classes. In other words, the SO tag patterns constitute a simple algorithmic way to filter texts without explicit polarity out.

3.2. Semantic Orientation Tag Patterns Extraction

To build the SO base representing each text described in the subsection above, we extract the same Part-of-Speech (POS) tag patterns proposed in Turney (2002). Table 2 replicates these tag patterns, and the respective *TGrep2* (Rohde, 2001) expressions we used in our code.
The Part-of-Speech (POS) tagging is performed using the tagger proposed in Toutanova et al. (2003).

Tgrep 2 expression	POS Tag pattern
`(JJ.(NN|NNS))`	JJ + NN or NS
`(RB.(JJ!.(NN|NNS)))`	RB + JJ + not NN, not NNS
`(RBR.(JJ!.(NN|NNS)))`	RBR + JJ + not NN, not NNS
`(RBS.(JJ!.(NN|NNS)))`	RBS + JJ + not NN, not NNS
`(JJ.(JJ!.(NN|NNS)))`	JJ + JJ + not NN, not NNS
`(NN.(JJ!.(NN|NNS)))`	NN + JJ + not NN, not NNS
`(NS.(JJ!.(NN|NNS)))`	NS + JJ + not NN, not NNS
`(RB.(VB|VBD|VBN|VBG))`	RB + VB or VBD or VBN or VBG
`(RBR.(VB|VBD|VBN|VBG))`	RBR + VB or VBD or VBN or VBG
`(RBS.(VB|VBD|VBN|VBG))`	RBS + VB or VBD or VBN or VBG

Table 2: Extracted POS tag patterns using *TGrep2* expressions.

3.3. Sentiment Lexicon Learning and Compilation

Up to this point, we have not made use of any sentiment annotation of our dataset. That said, we could learn the polarity of each tag pattern using a totally unsupervised approach. One such approach is the *SO-PMI* method proposed by Turney (2002) and, for example, extensively discussed in Taboada et al. (2011). This approach uses search engines hit counts to calculate the Pointwise Mutual Information (PMI) between a given "keyword" and two fixed strong opinion words, such as "good" and "bad", which are expected to have opposite sentiment polarity. The *SO-PMI* is the difference between the two PMI measures[5]. Sticking to our example, we would expect that the PMI between the word "bad" and "economic concerns" would be higher than the PMI between the word "good" and "economic concerns", what would make the text "*Stocks tank on global economic concerns*" more biased to a negative sentiment label than the positive one. Nonetheless, as pointed out in Taboada et al. (2006), search engines are living organisms, subjected to a constant updating process, making the *SO-PMI* measure highly unstable over time. Additionally, the goal of this study is to learn a domain-specific lexicon but, typically, search engines do not segregate queries to texts from a specific domain, which is in our case the financial markets domain.
As an alternative to unsupervised learning, we leverage the sentiment annotation of our dataset and train three supervised binary sentiment classifiers: Logistic Regression, Linear Support Vector Machine and Neural Network. All classifiers are trained to predict the probability of the positive sentiment label and except for the Neural Network classifier were implemented using the Scikit-learn library (Pedregosa et al., 2011). The Neural Network uses the Keras library (Chollet, 2015) and is trained using an architecture with one hidden dense layer and one final dense layer with only one neuron.
Below, we provide a detailed explanation of all steps leading from the dataset messages to our proposed Sentiment Lexicon compilation:

1. *SO Tag Pattern Extraction*: After performing the preprocessing steps described in Sec. 2., for each message we extract all possible tag patterns (terms) described in Table 2. We assign the set of all different tag patterns extracted from our dataset as the vocabulary set V. When performing this step we ended up with a vocabulary with $1,185$ entries, which constitutes the dimension of our sparse vector space model.

[4]Semantic Orientation is a measure of subjectivity and opinion of a given text: see the early works of Osgood (1952) and a more recent review in (Taboada et al., 2011).

[5]Note that the measure will be negative if the PMI ("distance") between a given keyword and "bad" is higher than between the keyword "good". In simple terms, negative(positive) measures are associated with negative(positive) sentiments.

2. *Instance Representation*: We represent each message in the base of terms V using three different weight schemes: Term-Frequency (TF), Term Frequency-Inverse Document Frequency[6] (TF-IDF) and One-hot, where the term representation is assigned to one if the term appears in the text and zero otherwise.

3. *Hyperparameter Selection*: We randomly split 85% of the data for hyperparameters optimization (training) with the remaining 15% "left out" as test set. The hyperparameter selection is performed in the training data using 10-fold cross-validation. The cross-validation is implemented using greedy-search, which to sweep all possible hyperparameters of the search space for each of the three classifiers and selects the best model for a given metric. Table 3 describes the hyperparameters space for each classifier. In total this step outputs 9 models, i.e. 3 (classifiers) times 3 text representations per classifier.

4. *Sentiment Lexicon Compilation*: At this stage, our models can be consumed to classify the binary sentiment of any text. However, in order to compile a sentiment lexicon as a handy language resource, we perform the following tasks:

   - First, at inference time, we predict the **POS** label probability $\{p_i\}_{i=1}^{1,185}$ for all the entries of our vocabulary V using the One-hot models. Technically, this step is implemented passing though our classifiers $1,185$ vectors. Each of these vectors have zero elements for all dimensions expect for the ith dimension corresponding to the lexicon V_i which has entry one.
   - Second, we introduce a cut-off probability for the decision boundary, i.e. the cut-off probability decides if a given term should be grouped in the positive or negative word lists. Thus, all terms V_i with probability p_i greater (less) than 0.60 (0.40) are classified as positive (negative). The remaining terms are filtered out ($0.40 \leq p_i \leq 0.60$).

Classifier	**Hyperparameters**
Logistic Regression	`regularization_type` = {l1, l2} , `C_regularization` = {0.10, 1, 10, 100}
Linear SVM	`C_regularization` = {0.10, 1, 10, 100}
Neural Network	`hidden_layer_n_neurons` = {64, 128}

Table 3: Hyperparameters space.

[6]The standard IDF weight scheme is employed in our work. This weight will lower the total TF weight of any term $V_i \in V$ that appears frequently in all instances of the dataset. For example, a term that appears in all instances (documents) will have a final TF-IDF weight equal to zero.

Our proposed "bag of Sentiment Orientations" representation address two main challenges. First, it filters out factual texts (neutral opinion). Second, it is our proposed solution to build a sentiment lexicon straight from a binary sentiment classifier. We make available the positive and negative word lists (sentiment lexicon) as a language resource, which can be found in the files `stocksenti-word-list-pos` and `stocksenti-word-list-neg` for the positive and negative sentiments, respectively.

4. Results

Table 4 shows the Sentiment classifiers test set accuracy for the 15% of our dataset samples that were "left out". The reported values are the accuracy for the best model selected during the cross-validation phase (training set) for each classifier.

Classifier	**TF**	**TF-IDF**	**One-Hot**
Logistic Regression	82.9	83.7	82.4
Linear SVM	82.8	83.0	82.7
Neural Network	91.3	90.8	91.4

Table 4: Test set accuracy for different classifiers and instance representations.

Our best classifier is the Neural Network and, for this classifier, the performance remains even when different text representations are used. For the other classifiers we can see that the TF-IDF representation performs the best. Based on these results, we selected the Neural Network model with One-hot representation to compile the Sentiment Lexicon using the approach described in the previous section. The final confusion matrix of the compiled Sentiment Lexicon can be found in Table 5.

Predicted/Actual	**POS**	**NEG**
POS	10,888	902
NEG	572	4,824

Table 5: Test set confusion matrix for the best classifier consumed to build the Sentiment Lexicon.

To evaluate our learnt sentiment lexicon, here named *StockSenti*, we perform two different strategies. To begin with, we assess how far our word lists are able to capture the sentiment of terms commonly used in the financial parlance, taking into account formal and informal language variations. In addition, we evaluate the effectiveness of our sentiment lexicon as a potential tool to extend the manually compiled *Loughran and McDonald Financial Sentiment Dictionary* of Loughran and Mcdonald (2011), which was built using exclusively corporate disclosures.

We provide many examples where our Sentiment Lexicon thrives in learning terms related positive and negative polarity for stock market texts. To exemplify, the term *next resistance* and *strong support* have high positive probability (0.82). On the contrary, *next support* is highly negative (with probability equal to 0.21). Further, the keywords *short squeeze*, *short covering* and *too cheap* are positive as *shorting opportunity*, *great short*, *too high*, *high price* and *buy puts* are negative[7].

Interestingly, our dictionary is able to capture some relationships between the economic environment and stock markets. For instance, *cheap oil* is classified as positive, in agreement with the average negative correlation between inflation and stock prices. Even phrases like *bad weather* (0.2 probability) that are less obvious to grasp[8] were correctly classified.

Particularity, all the examples cited above are misclassified by all publicly available dictionaries built using general corpora (*Sentiment140* (Go et al., 2009), *Bing Liu Sentiment Lexicon* (Liu, 2012), *MPQA Sentiment Lexicon* (Wilson et al., 2005), *Harvard Dictionary* (the *General Inquirer*) (Stone et al., 1966) and *VADER* (Hutto and Gilbert, 2014)).

Regarding the effectiveness in extending the *Loughran and McDonald Financial Sentiment Dictionary* of Loughran and Mcdonald (2011) we grouped in Table 6 a few examples of words that are not present in this financial domain dictionary and, thus, are potential candidates to extend the same.

Positive	**Negative**
solid quarter, extremely undervalued, green day, buyback program, strong cash, outperform recommendation, solid company, major upgrade, legislative inaction	strong sell-off, insider trading, unprofitable allocation, expensive debt, litigious fraud, profitless resources

Table 6: Examples of terms part of our sentiment lexicon.

5. Conclusion

This work makes available to the research community a novel text corpora and sentiment lexicon for financial text mining. Indisputably, both language resources are valuable to the studies of corporate disclosures, e.g. corporate press releases, annual reports and so forth.

The sentiment classifier built on top of the sentiment labelled *Yahoo Message Board* service covers a broad range of stocks and is effective in classifying the sentiment of terms common to the stock markets parlance. We extensively assessed different classifiers and text representations and the best combination of text representation weights and classifier model achieves 91.4% accuracy in the test set.

Additionally, we propose a method to build a sentiment lexicon from a sentiment classifier by representing each dataset instance (message) in a base of terms with high polarity, what we named "bag of Semantic Orientation".

We assessed the potential of our learnt sentiment lexicon to be utilized to extend manually annotated sentiment lexicons (crafted using 10-K or 10-Q financial documents). Not only our sentiment lexicon is effective to extend financial sentiment dictionaries, but also it is able to capture the sentiment of terms from the formal and informal language of financial stock markets.

[7] The reader not familiar with the words "long", "short", "support", "resistance", "covering" and "put"/"call" derivatives instruments is encourage to consult introductory capital markets books to gain specific domain knowledge.

[8] A closer look at the dataset reveals that the *bad weather* phrase was extracted from oil companies. In this case, the negative hint is a consequence of damages caused by hurricane seasons.

6. Bibliographical References

Chollet, F. (2015). Keras. https://github.com/fchollet/keras.

Dasgupta, S. and Ng, V. (2009). Mine the easy, classify the hard: a semi-supervised approach to automatic sentiment classification. In *ACL '09 Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 701–709. Association for Computational Linguistics, aug.

Davies, A. and Ghahramani, Z. (2011). Language-independent Bayesian sentiment mining of Twitter. In *5th SNA-KDD Workshop 11 (SNA-KDD 11)*.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach. In Lise Getoor et al., editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 513–520, New York, NY, USA, jun. ACM.

Go, A., Bhayani, R., and Huang, L. (2009). Twitter sentiment classification using distant supervision. Technical report, Stanford University (CS224N Project Report).

Hassan, A. and Radev, D. (2010). Identifying text polarity using random walks. In *Proceeding ACL '10 Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 395–403. Association for Computational Linguistics, jul.

Hatzivassiloglou, V. and McKeown, K. R. (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.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '04*, page 168, New York, New York, USA, aug. ACM Press.

Hutto, C. and Gilbert, E. (2014). VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. In *Procceddings of 8th International AAAI Conference on Weblogs and Social Media*, pages 216–225.

Kanayama, H. and Nasukawa, T. (2006). Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *EMNLP '06 Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 355–363. Association for Computational Linguistics, jul.

Liu, B. (2012). *Sentiment analysis and opinion mining*.

Loughran, T. and Mcdonald, B. (2011). When is a Liability not a Liability? Textual Analysis , Dictionaries , and 10-Ks. *The Journal of Finance*, 66(1):35–65.

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, nov.

Osgood, C. E. (1952). The nature and measurement of meaning. *Psychological Bulletin*, 49(3):197–237.

Pang, B. and Lee, L. (2008). Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, 2(1–2):1–135, jan.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - EMNLP '02*, volume 10, pages 79–86, Morristown, NJ, USA, jul. Association for Computational Linguistics.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Ponomareva, N. and Thelwall, M. (2013). Semi-supervised vs. Cross-domain Graphs for Sentiment Analysis. In Galia Angelova, et al., editors, *Proceedings of the Conference on Recent Advances in Natural Language Processing (RANLP'13)*, pages 571–578, Hissar, Bulgaria.

Rao, D. and Ravichandran, D. (2009). Semi-supervised polarity lexicon induction. In *EACL '09 Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 675–682. Association for Computational Linguistics, mar.

Rohde, D. L. T. (2001). TGrep2 User Manual.

Shiller, R. J. (2000). Measuring Bubble Expectations and Investor Confidence. *Journal of Psychology and Financial Markets*, 1(1):49–60, mar.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., and Christopher D. Manning Andrew Y. Ng, C. P. (2013). Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642.

Stone, P. J., Dunphy, D. C., and Smith, M. S. (1966). The General Inquirer: A Computer Approach to Content Analysis.

Taboada, M., Anthony, C., and Voll, K. (2006). Methods for Creating Semantic Orientation Databases. In *Proceeding of LREC-06, the 5th International Conference on Language Resources and Evaluation*, pages 427–432.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., and Stede, M. (2011). Lexicon-Based Methods for Sentiment Analysis. *Computational Linguistics*, 37(2):267–307, jun.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - NAACL '03*, volume 1, pages 173–180, Morristown, NJ, USA, may. Association for Computational Linguistics.

Turney, P. D. (2002). Thumbs up or thumbs down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, pages 417–424, Stroudsburg, PA, USA, jul. Association for Computational Linguistics.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing - HLT '05*, pages 347–354, Morristown, NJ, USA, oct. Association for Computational Linguistics.

Zhou, S., Chen, Q., and Wang, X. (2013). Active deep learning method for semi-supervised sentiment classification. *Neurocomputing*, 120:536–546, nov.