Semi-Supervised Approaches for Sentiment Analysis for Citing Sentence

Abstract

We apply two semi-supervised learning approaches for building better citing sentences sentiment analyzer based on limited training data. First of all, we extract several kinds of text-based information from each citing sentence as the content feature. Second, we used the network information for each citing paper and cited paper as the meta feature. Third, we set up an expeiment for verifying that meta features are helpful for sentiment classification. We will show that the introduction of meta features results in about 2% accuracy improvement over the state-of-the-art approach. Finally, we compare the performance of two classical semi-supervised learning approaches for predicting sentiment for the targeting dataset.

1 Introduction

Sentiment analysis has become an interesting topic in natural language processing and text mining. In QA Systems (Oh et al., 2012), sentiment mining can improve the relevance and usefulness of the re-ranked top-k answers. In recommender systems (Leung et al., 2006), mining the sentiment of user comments can increase the quality of the items that the service providers recommend, thus enhances user satisfaction. In the financial area, mining news or reports from the social media such as Reuters and New York Times also gives some insights to experts for predicting stock prices, thus make more profitable investment. In scientific research, analysis of citing sentences plays an important role (Bonzi, 1982) in enhancing publication and scientist ranking.

Sentiment classification for citing sentence is a binary text classification problem, where the category set $\mathcal{C} = \{positive, negative\}$. In the information retrieval field, text classification is a classic problem that has been substantially studied (Joachims, 1998; Sebastiani, 2002) and the state-of-the-art classifiers have achieved very high performance on textual web corpus. However, the main scope of traditional text classifiers is content topics (i.e. the texts are supposed be classified by their topics). When changing the scope to broader targets such as sentiment classification, the traditional classification approach based on the Bag-of-Words model does not perform well (Pang et al., 2002). Despite the synonym problem of Bag-of-Words model. In the task of sentiment classification, features extracted from the syntactic and semantic structure are necessary because people tend to express their polarity opinion by sentence or paragraph as a whole rather than a single word or phrase. A very simple example is the negation modifier, which will completely inverse the sentence sentiment. Although this problem can be partially solved by introducing n-gram features such as 3-grams or 4-grams, the problem stands still when the distance between modifier and target becomes very large. Also, the n-gram model will suffer from serious curse-of-dimensionality problem. As a result, more natural language processing such as POS tagging, syntax parsing and dependency parsing (Nakagawa et al., 2010; Yessenalina et al., 2010) is needed for extract features for sentiment classification.

Compared to the sentiment classification on user comment data such as IMDB movie reviews (Dodds, 2006), where users indicate the sentiment in a straightforward way, sentiment classification on paper citations is a harder task (Athar, 2011). The author of paper tends to express the opinion on the cited paper implicitly due to respectfulness and objectiveness. As a result, features of citing sentences need to be more indicative and sensitive to the subtle statements. Besides the difficulties, citing sentences have some good properties that can be taken advantage of when doing sentiment classification. First of all, unlike online comments such as movie reviews, sentences in research paper are better formed for extracting syntactic structure, whereas online comments are always poor written because of misspellings and slang. Second, citation sentence have a large set of meta-data such as author, citation network, collaboration network and citation number. These have not been used into sentiment analysis by previous studies. These two advantages result in a gain over the current state-of-the-art performance as will be shown in our experiment. Considering the methodology, current studies on sentiment classification are based on supervised learning approaches(Wilbur et al., 2006; Athar, 2011), where training data need to be generated by human labeling. However, with the growing number of research paper along years as shown in Figure [1]1, its impossible to manually label so many citing sentences for building a reliable supervised learning model for much larger testing data set. Focusing on this problem, in this paper, we propose a semi-supervised learning approach by using a small training data generated by rule-based approach for adapting large data set with relatively high performance. In addition, we propose several citation-based features that have not been evaluated by previous studies.

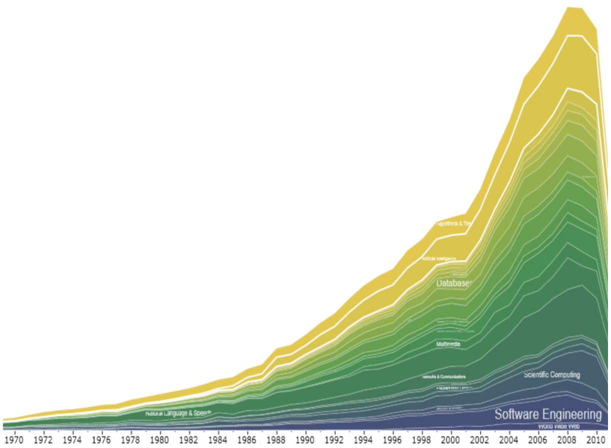

Figure 1: Trend for publication volume in computer science from 1970 to 2010 by Microsoft Academic Search

2 Related Work

Our work is related to three areas — sentiment mining, bibliographic search and semi-supervised learning. Early work on sentiment mining mainly focused on product reviews. Dave et al. (Dave et al., 2003) classify opinions of product reviews into three categories (poor, mixed, good) by identifying product features. This paper also makes several evolutions of classification approaches, where their method shows better performance for larger scale off-line web analysis such as query clustering. Cardie et al. (Cardie et al., 2003) mine opinions in the answer set for broader question prospective to improve the performance of the Question-Answering system. In their approach, sentiment is hierarchical. Polarity opinions were assigned prefixes indicating sentiment degree such as “strong” and “weak”. Das et al. (Das and Chen, 2007) use classification blending method for predicting investor sentiment on the stock message board. Pang et al. (Pang et al., 2002) evaluate several classical machine learning methods (Naive Bayes, Maximum Entropy, Support Vector Machines) on sentiment classification for movie reviews, which states the drawback of the traditional classifiers over the sentiment classification task.

Citation sentiment analysis is a relatively new topic that has recently raised the attention of researchers. Teufel et al. (Teufel et al., 2006b; Teufel et al., 2006a) is the earliest attempt to consider using sentiment classification methods to improve the performance of impact factor estimation and citation indexing. Athar (Athar, 2011) performs deeper analysis of sentence structure feature. In the following work by Athar (Athar and Teufel, 2012a; Athar and Teufel, 2012b), is mainly focused on exploring more features from in-corpus information, such as the context of the citing sentences. Unlike Athar’s approach, in this paper, we go in a different direction by exploring the out of corpus information (meta feature). In our experiment, we implement the algorithm described in (Athar, 2011) as baseline. And result shows that these new features can achieve better performance over Athar’s approach.

Bibliographic search is used for improving the efficiency of finding relevant papers and experts by researchers. Currently, services such as Microsoft Academic Search, Google scholar, DBLP, ArnetMiner and AAN are widely used for scientific study. Also, because the structure of both paper and author space are network, there are series of research trying to apply network mining techniques to bibliographic search. Leicht et al.(Leicht et al., 2007) proposed a Expectation-

[1]http://academic.research.microsoft.com/

Maximization (EM) approach for finding hidden community among research papers. Radev et al. (Radev et al., 2009a) introduces several classical graph based analysis matrices such as cluster diameter and PageRank. One important application of our work in bibliographic search is to make the H-index for authors more accurate. H-index (Hirsch, 2005) proposed by Hirsch is an important criterion in ranking scientist by their research impacts. Paper citation sentiment analysis in this field provides a rational view in deciding the quality of edges. For example, we can reduce or remove the edge if the corresponding citing sentence casts a negative opinion towards the cited paper. This will prevent the H-index calculating algorithm from giving too much score on the papers which were frequently criticised.

Semi-supervised learning is often used when training data are too short compared to the testing data as is the case in our study. Currently, two approaches are widely used, the first is self-training, the other is co-training (Zhu, 2005). Riloff et al. (Riloff et al., 2003) extract subjective nouns using self-training approach that wraps a Naive Bayes classifier, which gain a high performance on both precision and recall. Collins et al. (Collins and Singer, 1999) used a co-training approach by summarizing a small set of basic and global rules as "seeds" and applying these rules into an iterative "training" procedure to discover more local rules. These "seeds" are the only supervision, and are generated manually by analyzing the give data. In our experiments, we compare both of the two approaches using the combine feature (content feature and meta feature). The rest of the paper is organized as follows: In section 3 we will introduce the features of each citing sentence in detail. Section 4 will introduce our whole algorithm framework, including the setting up of two semi-supervised approaches. Section 5 will introduce the target data set in this paper, including some statistics that is useful in our experiment. Section 6 will show the experimental results, including the comparison of baseline feature selection (Athar, 2011) and feature selection with meta feature included. Also, the performance of two semi-supervised learning approach will be illustrated. Section 7 will state current drawbacks of our system and the tentative solutions we will work on.

3 Feature Study

3.1 Content-Based Feature

Term Appearance Feature: This feature is basically derived from the traditional bag of words model, where each term is one dimension of the document vector. The difference is we used the POS tagging approach for annotating each term as a suffix for disambiguation purpose.

Polarity Feature: For each citation, we assign four features indicating the counts of sentence polarity. They are '#SP#'–(Strong Positive), '#WP#'–(Weak Positive), '#WN#'–(Weak Negative) and '#SN#'–(Strong Negative). The polarity words are pre-compiled from the ontology of MPQA Opinion Corpus [2]. The format of this ontology is:

type=weaksubj len=1 word1=abandoned pos1=adj stemmed1=n priorpolarity=negative
type=weaksubj len=1 word1=abandonment pos1=noun stemmed1=n priorpolarity=negative
type=weaksubj len=1 word1=abandon pos1=verb stemmed1=y priorpolarity=negative
type=strongsubj len=1 word1=abase pos1=verb stemmed1=y priorpolarity=negative
type=strongsubj len=1 word1=abasement pos1=anypos stemmed1=y priorpolarity=negative
type=strongsubj len=1 word1=abash pos1=verb stemmed1=y priorpolarity=negative

In our implementation, we use the term matching method for counting the frequency of each type of polarity. Here is one issue needs to be aware. When the negation (such as "not", "n't") appears, both the polarity and degree will be inversed. For example, if the polarity of "abandoned" is "weak negative", the polarity of its negation "not abandoned" is "strong positive"

Dependency Feature: We use the Stanford Dependency Parser (De Marneffe et al., 2006) for discovering all the dependency instances in the sentence.

Negation Feature: The negation modifier will completely inverse the sentiment of a sentence. In our approach, we apply a simple rule based approach for annotating negation to a specific term. That is, a word is annotated a negation if and only if it is a sibling of negation modifier and modified

[2] http://www.cs.pitt.edu/mpqa/

by adjective modifier or adverb modifier. Also, the parent mode which modified by negation will be annotated a negation. For example, when the dependency tree is present in Figure 2, where the original input is "*The mean approach didn't perform that well.*"

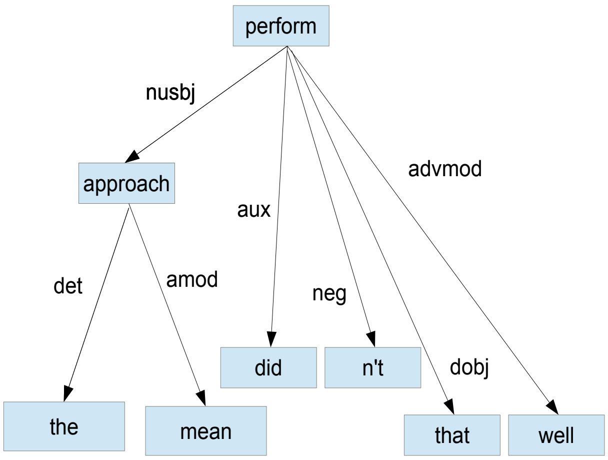

Figure 2: Example of Dependency Tree and Negation Annotation

As shown in Figure 2, the term "well" is the adverb modifier of its parent node "perform", and it's the sibling of negation modifier "n't". Also, the parent node "perform" is modify the the negation modifier "n't". As the result, both term "perform" and "well" will be added a suffix indicating negation as "perform<neg>" and "well<neg>".

3.2 Meta Feature

Citation Count: Number of citations of the cited paper

Self Citation: This is a boolean feature, indicates whether the paper is self-cited.

co-author feature: this is also a boolean feature that indicates whether the author of the citing paper has ever co-authored a paper with the author of the cited paper. this feature is somewhat similar to the self-citation feature.

In the implementation, we use a pipelined framework that takes the raw citing sentence stream with the paper id of both citing and cited paper as input and generates refined features. Again, we take the input "*The mean approach didn't perform that well.*" as the example for showing the workflow in Figure 3

Feature Name	Type	Description
term & POS tag	content	all the terms appended by their POS tags
polarity	content	4-dimension, indicates the number of different types of polarities
dependency	content	word dependency
negation	content	term who were negated are appended by suffix <neg>
citation	meta	number of citations of the cited paper
self	meta	a boolean indicates whether it is a self citation
co-author	meta	whether the author of the cited paper has been coauthored with the writer

Table 1: Features

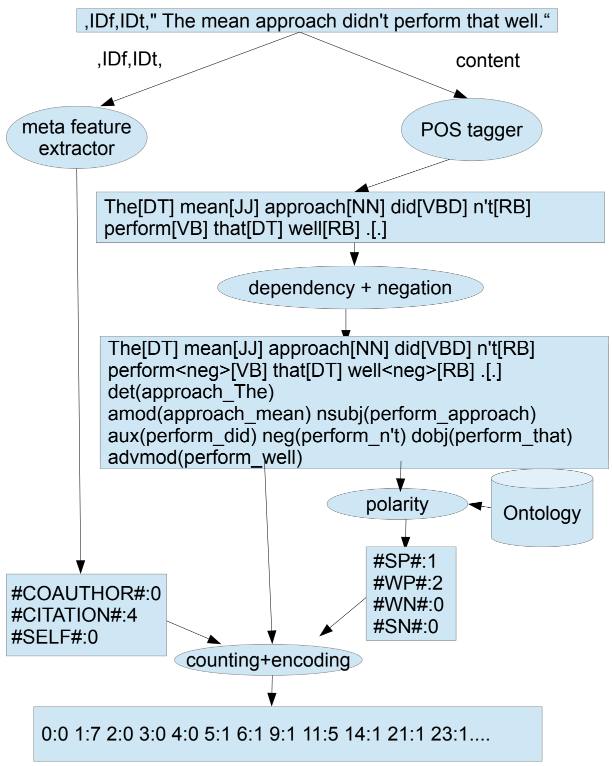

Figure 3: Pipeline for feature generation

4 Methodology

This section mainly talks about the details of the semi-supervised learning.

4.1 Self-Training

Given the seed data, self-training will give prediction on each test case. Given an input case x, different from the hard classifier that give prediction $f(x) \in \mathcal{C}$, in the self training classifier, the system output will be soft that also produces the probability for each label, where $f_s(x) = (y, p(y|x)) \in \mathcal{C} \times [0, 1]$ By choosing the classified data who has dominate probability over they other candidates as new training data. The training set will be enriched for the second run of training and testing. The self-training procedure runs iteratively until most data have been classified with high confidence(98% in our experimental setting), The algorithm is described in Algorithm 1:

Algorithm 1: Self-Training

Input:
$Seeds\ \mathcal{S} = \{(s_1, c_1), (s_2, c_2), \cdots, (s_m, c_m)\}$
$Test\ data\ \mathcal{T} = \{x_1, x_2, \cdots, x_n\}$
Procedure ALGORITHM

```
while #S/#T < σ do
    Train a classifier: f ← S
    for x ∈ T do
        if f_p(x) = p(y|x) > θ then
            S ← (x, y(x))
        end
    end
end
```

Where θ is the threshold for adding the test case into training data, which is 0.95 for positive cases and 0.7 for negative cases. σ is the threshold for ending the iteration loop of learning, which controls the number of new training data. When the training data is large enough compared to the test data (in our experiment $\sigma = 0.7$), the iteration will stop.

4.2 Co-Training

The assumption for co-training is there are at least two kinds of feature are uncorrelated. In our study, we assume the content features and meta feature are independent as they reflect two different aspects of citation. In the co-training approach, citation features are incorporated iteratively by their co-relation to the class. The co-relation is defined by a function with binary inputs $h(c, x) \sim \mathcal{C} \times \mathcal{X} \to [0, 1]$, where h can be interpreted as the conditional probability $p(c|x)$ of class c given a fact that feature x is seen in the case. $\mathcal{X}$ here is the feature space. This approach is called decision list, where each decision $x \to c$ are sorted by the co-relation function h. In addition, the final label of a given input is:

$$c_x = argmax_{x \in \mathcal{X}, c \in \mathcal{C}} h(c, x) \quad (1)$$

Where the label of the input case is defined by its most "confident" feature. In practice, the co-relation function h is defined by counting with Laplace smoothing:

$$h(c, x) = \frac{Count(c, x) + \alpha}{Count(x) + m\alpha} \quad (2)$$

Where $Count(c, x)$ is the number of co-occurrence of (c, x) pair, α is the Laplace smoother, m is the number of classes. In our case, $m = 2$ because our task is binary classification.

Given a set of "seed" corpus, we first calculate the co-relation between features and classes. Then add these features into the feature set, and the labels are predicted by the logic in Equation (1).
After tuning the parameter, we select $\alpha = 0.1$ and $\theta = 0.8$ as our fixed parameter in the experiment.

5 Data Set

The data set we use is AAN (ACL Anthology Network(Radev et al., 2009b)), which contains the citation summaries, paper texts, citation network, co-author network, author lists. The data is complete for generating all the features as shown in Section 3. Table2 shows some useful statistics.
After compiling the original file, we get 15,014 targeting sentences from the data set. Also, in the experiment, we use a threshold on the citation length to filter the citations with too many words. To decide this threshold, we did a simple statistic that shows the distribution of the citation over its length as shown in Figure 4
From Figure 4, we can see the citation distribution doesn't simply follow a normal distribution, where the mean is 38. In our experiment, we set our citation threshold as 80, where the actual number of citations in our experiment is 14,386 (95.8% of the total volume).

Algorithm 2: Co-Training

Input:

$Seed\ Features:\ \mathcal{S} = \{(s_1, c_1), (s_2, c_2), \cdots, (s_m, c_m)\}$

$Test\ data:\ \mathcal{T} = \{t_1, t_2, \cdots, t_n\}$

$Batch\ Number: n = 5$

$Class\ Number: m = 2$

Procedure ALGORITHM

while *true* **do**

1.Extract feature pairs from the current training set $\mathcal{S}$

2.Distribute the features into $Set_{content}$ and Set_{meta} by their types

3.Count the occurrence of $Count(c, x)$ for each (c, x) pair

4.Use features in $Set_{content}$ to predict labels, using Equation (1)

5.Add top $n \times m$ in Set_{meta} with $\max_c h(c, x) > \theta$ into the Set_{meta}

6.Use features in Set_{meta} to predict labels, using Equation (1)

7.Add top $n \times m$ in $Set_{content}$ with $\max_c h(c, x) > \theta$ into the $Content - S$

8.$n = n + 5$

9.**if** $Set_{content}$ *and* Set_{meta} *are not change* **then**

break

end

end

Item	Number
Number of papers	18,290
Number of authors	14,799
Number of paper citations	84,237
Number of author collaborations	57,499

Table 2: Data Set

As the main approach of this paper is a semi-supervised learning, training data doesn't need to be large. However, because we also compare the performance with meta features included, we still need a traditional supervised learning tool to complete this task. As the result, we split our experiment into two tasks. In the first task, we manually label 1833 citing sentences with 1423 positive cases and 410 negative cases. Then we split the labeled data into 1000 training data and 833 validation data. In the second task, we compare the performance of two semi-supervised learning approach, where we manually label 100 citations as training seeds, which is a much smaller training set compared to the first task. The main classifier for the two tasks we use is support vector machines (SVM) (Joachims, 1998; Chang and Lin, 2011), a well-known classifier currently gain the best classification results.

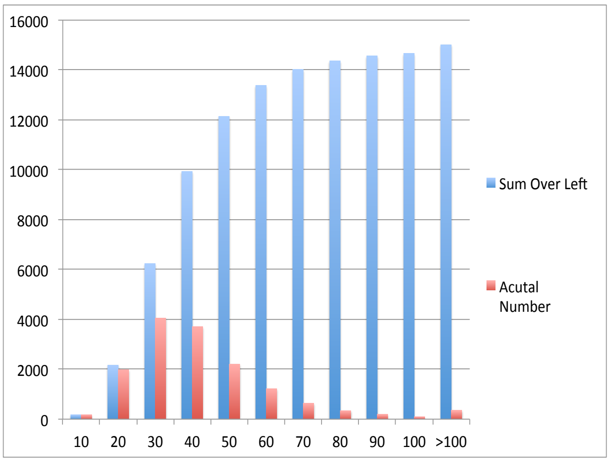

Figure 4: Distribution of Citation Length

6 Experiment

6.1 Feature Study

Unlike Athar's (Athar, 2011) approach, we define our task as a binary classification problem rather than three classification problem (remove the label "objective"). The reason for doing this is the class "objective" can be assigned if the possibilities can't reach the threshold for negative or positive label. Also, the purpose for sentiment analysis on citation sentence is to identify the "fake" edges that express criticism, not the objective sentiment. As the result, we treat "objective" sentiment as "positive" sentiment in our study. For the binary classification problem, we use accuracy as our general evaluation metric. For better presenting the result of the different feature combinations, we assign each combination an id as shown in Table 3.

Feature Combination	id
Bag-Of-Words	1
POS	2
POS+Dep	3
POS+Dep+Neg	4
POS+Dep+Neg+co-author	5
POS+Dep+Neg+coauthor+citation number	6
POS+Dep+Neg+meta	7

Table 3: Feature Combinations

Table 4 shows the experimental result of different feature settings.

id	Accuracy	Precision	Recall	F-score
1	0.758	0.735	0.808	0.77
2	0.787	0.742	0.812	0.775
3	0.791	0.747	0.84	0.79
4	0.805	0.753	0.872	0.808
5	0.813	0.762	0.88	0.817
6	0.821	0.771	0.910	0.835
7	0.825	0.774	0.911	0.837

Table 4: Experimental Results for Feature Study

From the results we can see that by adding meta features, the performance is better than the content only(state-of-the-art) approach. After observing the data, we find the content based classifier tends to classify some objective statements into negative label because of some polarities. However, some negative polarities are used for describing the hardness of the problem rather than the cited paper. By introducing meta features, one can reduce the negative effect of the polarities post on the classification task. That's why the new features result in better performance.

6.2 Semi-Supervised Learning

To achieve the effect of lacking training data, we randomly select 100 labeled case from the training data. Then we run both the self-training approach and co-training approach as described in Section 4. The experimental result is shown in Table 5

Method	self-training	co-training
Accuracy	0.793	0.769
Precision	0.792	0.768
Recall	0.97	0.859
F-score	0.873	0.812

Table 5: Experimental Result for Semi-Supervised Learning

Both of the two approaches converge very quickly (3 iterations for self-training and 6 times for co-training). From the table we can see that the performance of self-training outperform that of co-training. One possible explanation is that the core classification for self-training is SVM, which can better handle the high dimensional data, while the decision list approach works better for low dimensional data with less sparcity. Also, due to the unbalance of distribution of positive and negative label, we use the different thresholds (positive is higher) to prevent updating dominated by positive features or cases.

7 Conclusion & Future Work

This paper mainly is on sentiment analysis for citing sentences with limited training data. The experimental results show that the meta feature can enhance classification performance. We consider enhancing our work in the following aspects. 1) Analysis of the inner relationship between the meta features and sentiment. 2) Apply more graph mining techniques on paper or author network to generalize to enhance the meta features 3) Try to apply richer data source on citation analysis. Although in some other data sources, the distribution might be heterogeneous, transferable approaches could tentatively find bridge between these data.

References

[Athar and Teufel2012a] A. Athar and S. Teufel. 2012a. Context-enhanced citation sentiment detection. *Montréal, Canada*, pages 597–601.

[Athar and Teufel2012b] A. Athar and S. Teufel. 2012b. Detection of implicit citations for sentiment detection. In *Association for Computational Linguistics*, page 18.

[Athar2011] A. Athar. 2011. Sentiment analysis of citations using sentence structure-based features. In *Proceedings of the ACL*, pages 81–87.

[Bonzi1982] S. Bonzi. 1982. Characteristics of a literature as predictors of relatedness between cited and citing works. *Journal of the American Society for Information Science*, 33(4):208–216.

[Cardie et al.2003] C. Cardie, J. Wiebe, T. Wilson, and D. Litman. 2003. Combining low-level and summary representations of opinions for multi-perspective question answering. In *Working Notes-New Directions in Question Answering (AAAI Spring Symposium Series)*, pages 1–13.

[Chang and Lin2011] C.C. Chang and C.J. Lin. 2011. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27.

[Collins and Singer1999] M. Collins and Y. Singer. 1999. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 100–110.

[Das and Chen2007] S.R. Das and M.Y. Chen. 2007. Yahoo! for amazon: Sentiment extraction from small talk on the web. *Management Science*, 53(9):1375–1388.

[Dave et al.2003] K. Dave, S. Lawrence, and D.M. Pennock. 2003. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, pages 519–528. ACM.

[De Marneffe et al.2006] M.C. De Marneffe, B. MacCartney, and C.D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454.

[Dodds2006] K. Dodds. 2006. Popular geopolitics and audience dispositions: James bond and the internet movie database (imdb). *Transactions of the Institute of British Geographers*, 31(2):116–130.

[Hirsch2005] J.E. Hirsch. 2005. An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences of the United states of America*, 102(46):16569.

[Joachims1998] T. Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. *Machine learning: ECML-98*, pages 137–142.

[Leicht et al.2007] E.A. Leicht, G. Clarkson, K. Shedden, and M.E.J. Newman. 2007. Large-scale structure of time evolving citation networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 59(1):75–83.

[Leung et al.2006] C.W.K. Leung, S.C.F. Chan, and F. Chung. 2006. Integrating collaborative filtering and sentiment analysis: A rating inference approach. In *Proceedings of The ECAI 2006 Workshop on Recommender Systems*, pages 62–66. Citeseer.

[Nakagawa et al.2010] T. Nakagawa, K. Inui, and S. Kurohashi. 2010. Dependency tree-based sentiment classification using crfs with hidden variables. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 786–794. Association for Computational Linguistics.

[Oh et al.2012] J.H. Oh, K. Torisawa, C. Hashimoto, T. Kawada, S. De Saeger, and J.K.Y. Wang. 2012. Why question answering using sentiment analysis and word classes. In *Proceedings of EMNLP-CoNLL 2012: Conference on Empirical Methods in Natural Language Processing and Natural Language Learning*.

[Pang et al.2002] 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.

[Radev et al.2009a] D.R. Radev, M.T. Joseph, B. Gibson, and P. Muthukrishnan. 2009a. A bibliometric and network analysis of the field of computational linguistics. *Journal of the American Society for Information Science and Technology*, 1001:48109–1092.

[Radev et al.2009b] Dragomir R. Radev, Pradeep Muthukrishnan, and Vahed Qazvinian. 2009b. The acl anthology network corpus.

[Riloff et al.2003] E. Riloff, J. Wiebe, and T. Wilson. 2003. Learning subjective nouns using extraction pattern bootstrapping. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 25–32. Association for Computational Linguistics.

[Sebastiani2002] F. Sebastiani. 2002. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

[Teufel et al.2006a] Simone Teufel, Advaith Siddharthan, and Dan Tidhar. 2006a. An annotation scheme for citation function. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, SigDIAL '06, pages 80–87, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Teufel et al.2006b] Simone Teufel, Advaith Siddharthan, and Dan Tidhar. 2006b. Automatic classification of citation function. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 103–110, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Wilbur et al.2006] W.J. Wilbur, A. Rzhetsky, and H. Shatkay. 2006. New directions in biomedical text annotation: definitions, guidelines and corpus construction. *BMC bioinformatics*, 7(1):356.

[Yessenalina et al.2010] Ainur Yessenalina, Yisong Yue, and Claire Cardie. 2010. Multi-level structured models for document-level sentiment classification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1046–1056, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Zhu2005] X. Zhu. 2005. Semi-supervised learning literature survey.