Semi-Supervised Polarity Lexicon Induction

Delip Rao[*]
Department of Computer Science
Johns Hopkins University
Baltimore, MD
delip@cs.jhu.edu

Deepak Ravichandran
Google Inc.
1600 Amphitheatre Parkway
Mountain View, CA
deepakr@google.com

Abstract

We present an extensive study on the problem of detecting polarity of words. We consider the polarity of a word to be either positive or negative. For example, words such as *good, beautiful*, and *wonderful* are considered as positive words; whereas words such as *bad, ugly*, and *sad* are considered negative words. We treat polarity detection as a semi-supervised label propagation problem in a graph. In the graph, each node represents a word whose polarity is to be determined. Each weighted edge encodes a relation that exists between two words. Each node (word) can have two labels: positive or negative. We study this framework in two different resource availability scenarios using WordNet and OpenOffice thesaurus when WordNet is not available. We report our results on three different languages: English, French, and Hindi. Our results indicate that label propagation improves significantly over the baseline and other semi-supervised learning methods like Mincuts and Randomized Mincuts for this task.

1 Introduction

Opinionated texts are characterized by words or phrases that communicate positive or negative sentiment. Consider the following example of two movie reviews[1] shown in Figure 1. The positive review is peppered with words such as *enjoyable, likeable, decent, breathtakingly* and the negative

Figure 1: Movie Reviews with positive (left) and negative (right) sentiment.

comment uses words like *ear-shattering, humorless, unbearable*. These terms and prior knowledge of their polarity could be used as features in a supervised classification framework to determine the sentiment of the opinionated text (E.g., (Esuli and Sebastiani, 2006)). Thus lexicons indicating polarity of such words are indispensable resources not only in automatic sentiment analysis but also in other natural language understanding tasks like textual entailment. This motivation was seen in the *General Enquirer* effort by Stone et al. (1966) and several others who manually construct such lexicons for the English language.[2] While it is possible to manually build these resources for a language, the ensuing effort is onerous. This motivates the need for automatic language-agnostic methods for building sentiment lexicons. The importance of this problem has warranted several efforts in the past, some of which will be reviewed here.

We demonstrate the application of graph-based semi-supervised learning for induction of polarity lexicons. We try several graph-based semi-

[*]Work done as a summer intern at Google Inc.

[1]Source: *Live Free or Die Hard*, rottentomatoes.com

[2]The *General Inquirer* tries to classify English words along several dimensions, including polarity.

Proceedings of the 12th Conference of the European Chapter of the ACL, pages 675–682,
Athens, Greece, 30 March – 3 April 2009. ©2009 Association for Computational Linguistics

supervised learning methods like Mincuts, Randomized Mincuts, and Label Propagation. In particular, we define a graph with nodes consisting of the words or phrases to be classified either as positive or negative. The edges between the nodes encode some notion of similarity. In a transductive fashion, a few of these nodes are labeled using seed examples and the labels for the remaining nodes are derived using these seeds. We explore natural word-graph sources like WordNet and exploit different relations within WordNet like synonymy and hypernymy. Our method is not just confined to WordNet; any source listing synonyms could be used. To demonstrate this, we show the use of OpenOffice thesaurus – a free resource available in several languages.[3]

We begin by discussing some related work in Section 2 and briefly describe the learning methods we use, in Section 3. Section 4 details our evaluation methodology along with detailed experiments for English. In Section 5 we demonstrate results in French and Hindi, as an example of how the method could be easily applied to other languages as well.

2 Related Work

The literature on sentiment polarity lexicon induction can be broadly classified into two categories, those based on corpora and the ones using WordNet.

2.1 Corpora based approaches

One of the earliest work on learning polarity of terms was by Hatzivassiloglou and McKeown (1997) who deduce polarity by exploiting constraints on conjoined adjectives in the Wall Street Journal corpus. For example, the conjunction "and" links adjectives of the same polarity while "but" links adjectives of opposite polarity. However the applicability of this method for other important classes of sentiment terms like nouns and verbs is yet to be demonstrated. Further they assume linguistic features specific to English.

Wiebe (2000) uses Lin (1998a) style distributionally similar adjectives in a cluster-and-label process to generate sentiment lexicon of adjectives.

In a different work, Riloff et al. (2003) use manually derived pattern templates to extract subjective nouns by bootstrapping.

[3] http://www.openoffice.org

Another corpora based method due to Turney and Littman (2003) tries to measure the semantic orientation $O(t)$ for a term t by

$$O(t) = \sum_{t_i \in S^+} PMI(t, t_i) - \sum_{t_j \in S^-} PMI(t, t_j)$$

where S^+ and S^- are minimal sets of polar terms that contain prototypical positive and negative terms respectively, and $PMI(t, t_i)$ is the pointwise mutual information (Lin, 1998b) between the terms t and t_i. While this method is general enough to be applied to several languages our aim was to develop methods that exploit more structured sources like WordNet to leverage benefits from the rich network structure.

Kaji and Kitsuregawa (2007) outline a method of building sentiment lexicons for Japanese using structural cues from HTML documents. Apart from being very specific to Japanese, excessive dependence on HTML structure makes their method brittle.

2.2 WordNet based approaches

These approaches use lexical relations defined in WordNet to derive sentiment lexicons. A simple but high-precision method proposed by Kim and Hovy (2006) is to add all synonyms of a polar word with the same polarity and its antonyms with reverse polarity. As demonstrated later, the method suffers from low recall and is unsuitable in situations when the seed polar words are too few – not uncommon in low resource languages.

In line with Turney's work, Kamps et. al. (2004) try to determine sentiments of adjectives in WordNet by measuring relative distance of the term from exemplars, such as "good" and "bad". The polarity orientation of a term t is measured as follows

$$O(t) = \frac{d(t, \mathsf{good}) - d(t, \mathsf{bad})}{d(\mathsf{good}, \mathsf{bad})}$$

where $d(.)$ is a WordNet based relatedness measure (Pedersen et al., 2004). Again they report results for adjectives alone.

Another relevant example is the recent work by Mihalcea et. al. (2007) on multilingual sentiment analysis using cross-lingual projections. This is achieved by using bridge resources like dictionaries and parallel corpora to build sentence subjectivity classifiers for the target language (Romanian). An interesting result from their work is that

only a small fraction of the lexicon entries preserve their polarities under translation.

The primary contributions of this paper are :

- An application of graph-based semi-supervised learning methods for inducing sentiment lexicons from WordNet and other thesauri. The label propagation method naturally allows combining several relations from WordNet.
- Our approach works on all classes of words and not just adjectives
- Though we report results for English, Hindi, and French, our methods can be easily replicated for other languages where WordNet is available.[4] In the absence of WordNet, any thesaurus listing synonyms could be used. We present one such result using the OpenOffice thesaurus – a freely available multilingual resource scarcely used in NLP literature.

3 Graph based semi-supervised learning

Most natural language data has some structure that could be exploited even in the absence of fully annotated data. For instance, documents are similar in the terms they contain, words could be synonyms of each other, and so on. Such information can be readily encoded as a graph where the presence of an edge between two nodes would indicate a relationship between the two nodes and, optionally, the weight on the edge could encode strength of the relationship. This additional information aids learning when very few annotated examples are present. We review three well known graph based semi-supervised learning methods – mincuts, randomized mincuts, and label propagation – that we use in induction of polarity lexicons.

3.1 Mincuts

A mincut of a weighted graph $G(V, E)$ is a partitioning the vertices V into V_1 and V_2 such that sum of the edge weights of all edges between V_1 and V_2 is minimal (Figure 2).

Mincuts for semi-supervised learning proposed by Blum and Chawla (2001) tries to classify datapoints by partitioning the similarity graph such that it minimizes the number of similar points being labeled differently. Mincuts have been used in semi-supervised learning for various tasks, including document level sentiment analysis (Pang and Lee, 2004). We explore the use of mincuts for the task of sentiment lexicon learning.

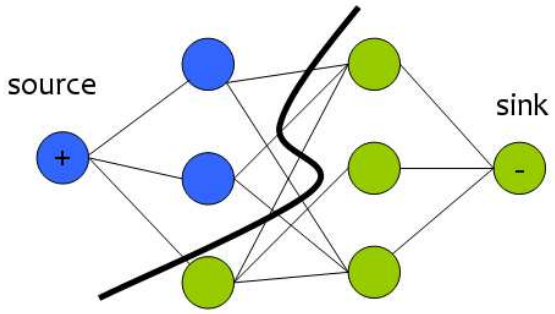

Figure 2: Semi-supervised classification using mincuts

3.2 Randomized Mincuts

An improvement to the basic mincut algorithm was proposed by Blum et. al. (2004). The deterministic mincut algorithm, solved using max-flow, produces only one of the several possible mincuts. Some of these cuts could be skewed thereby negatively effecting the results. As an extreme example consider the graph in Figure 3a. Let the nodes with degree one be labeled as positive and negative respectively, and for the purpose of illustration let all edges be of the same weight. The graph in Figure 3a. can be partitioned in four equal cost cuts – two of which are shown in (b) and (c). The mincut algorithm, depending on the implementation, will return only one of the extreme cuts (as in (b)) while the desired classification might be as shown in Figure 3c.

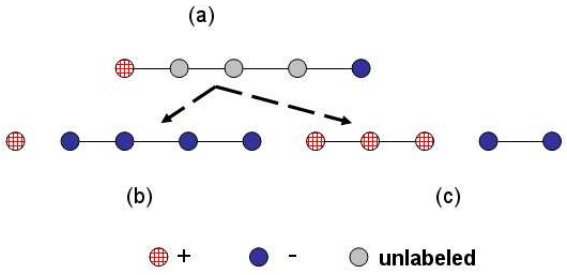

Figure 3: Problem with mincuts

The randomized mincut approach tries to address this problem by randomly perturbing the adjacency matrix by adding random noise.[5] Mincut is then performed on this perturbed graph. This is

[4] As of this writing, WordNet is available for more than 40 world languages (http://www.globalwordnet.org)

[5] We use a Gaussian noise $\mathcal{N}(0, 1)$.

repeated several times and unbalanced partitions are discarded. Finally the remaining partitions are used to deduce the final classification by majority voting. In the unlikely event of the voting resulting in a tie, we refrain from making a decision thus favoring precision over recall.

3.3 Label propagation

Another semi-supervised learning method we use is label propagation by Zhu and Ghahramani (2002). The label propagation algorithm is a transductive learning framework which uses a few examples, or seeds, to label a large number of unlabeled examples. In addition to the seed examples, the algorithm also uses a relation between the examples. This relation should have two requirements:

1. It should be transitive.
2. It should encode some notion of relatedness between the examples.

To name a few, examples of such relations include, synonymy, hypernymy, and similarity in some metric space. This relation between the examples can be easily encoded as a graph. Thus every node in the graph is an example and the edge represents the relation. Also associated with each node, is a probability distribution over the labels for the node. For the seed nodes, this distribution is known and kept fixed. The aim is to derive the distributions for the remaining nodes.

Consider a graph $G(V, E, W)$ with vertices V, edges E, and an $n \times n$ edge weight matrix $W = [w_{ij}]$, where $n = |V|$. The label propagation algorithm minimizes a quadratic energy function

$$\mathcal{E} = \frac{1}{2} \sum_{(i,j) \in E} w_{ij}(y_i - y_j)^2$$

where y_i and y_j are the labels assigned to the nodes i and j respectively.[6] Thus, to derive the labels at y_i, we set $\frac{\partial}{\partial y_i}\mathcal{E} = 0$ to obtain the following update equation

$$y_i = \frac{\sum_{(i,j)\in E} w_{ij} y_j}{\sum_{(i,j)\in E} w_{ij}}$$

In practice, we use the following iterative algorithm as noted by Zhu and Ghahramani (2002). A $n \times n$ stochastic transition matrix T is derived by row-normalizing W as follows:

$$T_{ij} = P(j \rightarrow i) = \frac{w_{ij}}{\sum_{k=1}^{n} w_{kj}}$$

where T_{ij} can be viewed as the transition probability from node j to node i. The algorithm proceeds as follows:

1. Assign a $n \times C$ matrix Y with the initial assignment of labels, where C is the number of classes.
2. Propagate labels for all nodes by computing $Y = TY$
3. Row-normalize Y such that each row adds up to one.
4. Clamp the seed examples in Y to their original values
5. Repeat 2-5 until Y converges.

There are several points to be noted. First, we add a special label "DEFAULT" to existing set of labels and set $P(\mathsf{DEFAULT} \mid node = u) = 1$ for all unlabeled nodes u. For all the seed nodes s with class label L we define $P(\mathsf{L} \mid node = s) = 1$. This ensures nodes that cannot be labeled at all[7] will retain $P(\mathsf{DEFAULT}) = 1$ thereby leading to a quick convergence. Second, the algorithm produces a probability distribution over the labels for all unlabeled points. This makes this method specially suitable for classifier combination approaches. For this paper, we simply select the most likely label as the predicted label for the point. Third, the algorithm eventually converges. For details on the proof for convergence we refer the reader to Zhu and Ghahramani (2002).

4 Evaluation and Experiments

We use the General Inquirer (GI)[8] data for evaluation. General Inquirer is lexicon of English words hand-labeled with categorical information along several dimensions. One such dimension is called valence, with 1915 words labeled "Positiv" (sic) and 2291 words labeled "Negativ" for words with positive and negative sentiments respectively. Since we want to evaluate the performance of the

[6] For binary classification $y_k \in \{-1, +1\}$.

[7] As an example of such a situation, consider a disconnected component of unlabeled nodes with no seed in it.

[8] http://www.wjh.harvard.edu/∼inquirer/

algorithms alone and not the recall issues in using WordNet, we only consider words from GI that also occur in WordNet. This leaves us the distribution of words as enumerated in Table 1.

PoS type	No. of Positives	No. of Negatives
Nouns	517	579
Verbs	319	562
Adjectives	547	438

Table 1: English evaluation data from General Inquirer

All experiments reported in Sections 4.1 to 4.5 use the data described above with a 50-50 split so that the first half is used as seeds and the second half is used for test. Note that all the experiments described below did not involve any parameter tuning thus obviating the need for a separate development test set. The effect of number of seeds on learning is described in Section 4.6.

4.1 Kim-Hovy method and improvements

Kim and Hovy (2006) enrich their sentiment lexicon from WordNet as follows. Synonyms of a positive word are positive while antonyms are treated as negative. This basic version suffers from a very poor recall as shown in the Figure 4 for adjectives (see iteration 1). The recall can be improved for a slight trade-off in precision if we re-run the above algorithm on the output produced at the previous level. This could be repeated iteratively until there is no noticeable change in precision/recall. We consider this as the best possible F1-score produced by the Kim-Hovy method. The classwise F1 for this method is shown in Table 2. We use these scores as our baseline.

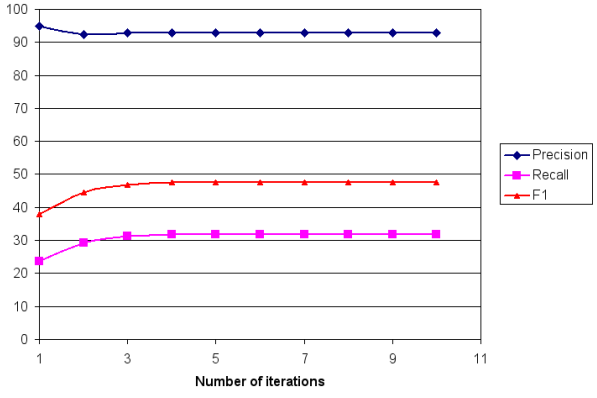

Figure 4: Kim-Hovy method

PoS type	P	R	F1
Nouns	92.59	21.43	34.80
Verbs	87.89	38.31	53.36
Adjectives	92.95	31.71	47.28

Table 2: Precision/Recall/F1-scores for Kim-Hovy method

4.2 Using prototypes

We now consider measuring semantic orientation from WordNet using prototypical examples such as "good" and "bad" similar to Kamps et al. (2004). Kamps et. al., report results only for adjectives though their method could be used for other part-of-speech types. The results for using prototypes are listed in Table 3. Note that the seed data was fully unused except for the examples "good" and "bad". We still test on the same test data as earlier for comparing results. Also note that the recall need not be 100 in this case as we refrain from making a decision when $d(t, \mathsf{good}) = d(t, \mathsf{bad})$.

PoS type	P	R	F1
Nouns	48.03	99.82	64.86
Verbs	58.12	100.00	73.51
Adjectives	57.35	99.59	72.78

Table 3: Precision/Recall/F1-scores for prototype method

4.3 Using mincuts and randomized mincuts

We now report results for mincuts and randomized mincuts algorithm using the WordNet synonym graph. As seen in Table 4, we only observed a marginal improvement (for verbs) over mincuts by using randomized mincuts.

But the overall improvement of using graph-based semi-supervised learning methods over the Kim-Hovy and Prototype methods is quite significant.

4.4 Using label propagation

We extract the synonym graph from WordNet with an edge between two nodes being defined iff one is a synonym of the other. When label propagation is performed on this graph results in Table 5 are observed. The results presented in Tables 2-5 need deeper inspection. The iterated Kim-Hovy method suffers from poor recall. However both mincut methods and the prototype method by

	P	R	F1
Nouns			
Mincut	68.25	100.00	81.13
RandMincut	68.32	99.09	80.08
Verbs			
Mincut	72.34	100.00	83.95
RandMincut	73.06	99.02	84.19
Adjectives			
Mincut	73.78	100.00	84.91
RandMincut	73.58	100.00	84.78

Table 4: Precision/Recall/F1-scores using mincuts and randomized mincuts

PoS type	P	R	F1
Nouns	82.55	58.58	58.53
Verbs	81.00	85.94	83.40
Adjectives	84.76	64.02	72.95

Table 5: Precision/Recall/F1-scores for Label Propogation

Kamps et. al., have high recall as they end up classifying every node as either positive or negative. Note that the recall for randomized mincut is not 100 as we do not make a classification decision when there is a tie in majority voting (refer Section 3.2). Observe that the label propagation method performs significantly better than previous graph based methods in precision. The reason for lower recall is attributed to the lack of connectivity between plausibly related nodes, thereby not facilitating the "spread" of labels from the labeled seed nodes to the unlabeled nodes. We address this problem by adding additional edges to the synonym graph in the next section.

4.5 Incorporating hypernyms

The main reason for low recall in label propagation is that the WordNet synonym graph is highly disconnected. Even nodes which are logically related have paths missing between them. For example the positive nouns *compliment* and *laud* belong to different synonym subgraphs without a path between them. But incorporating the hypernym edges the two are connected by the noun *praise*. So, we incorporated hypernyms of every node to improve connectivity. Performing label propagation on this combined graph gives much better results (Table 6) with much higher recall and even slightly better precision. In Table 6., we do not report results for adjectives as WordNet does not define hypernyms for adjectives. A natural question to ask is if we can use other WordNet relations too. We will defer this until section 6.

PoS type	P	R	F1
Nouns	83.88	99.64	91.08
Verbs	85.49	100.00	92.18
Adjectives	N/A	N/A	N/A

Table 6: Effect of adding hypernyms

4.6 Effect of number of seeds

The results reported in Sections 4.1 to 4.5 fixed the number of seeds. We now investigate the performance of the various methods on the number of seeds used. In particular, we are interested in performance under conditions when the number of seeds are few – which is the motivation for using semi-supervised learning in the first place. Figure 5 presents our results for English. Observe that Label Propagation performs much better than our baseline even when the number of seeds is as low as ten. Thus label propagation is especially suited when annotation data is extremely sparse.

One reason for mincuts performing badly with few seeds is because they generate degenrate cuts.

5 Adapting to other languages

In order to demonstrate the ease of adaptability of our method for other languages, we used the Hindi WordNet[9] to derive the adjective synonym graph. We selected 489 adjectives at random from a list of 10656 adjectives and this list was annotated by two native speakers of the language. The annotated list was then split 50-50 into seed and test sets. Label propagation was performed using the seed list and evaluated on the test list. The results are listed in Table 7.

Hindi	P	R	F1
	90.99	95.10	93.00

Table 7: Evaluation on Hindi dataset

WordNet might not be freely available for all languages or may not exist. In such cases building graph from an existing thesaurus might also suffice. As an example, we consider French. Although the French WordNet is available[10], we

[9] http://www.cfilt.iitb.ac.in/wordnet/webhwn/

[10] http://www.illc.uva.nl/EuroWordNet/consortium-ewn.html

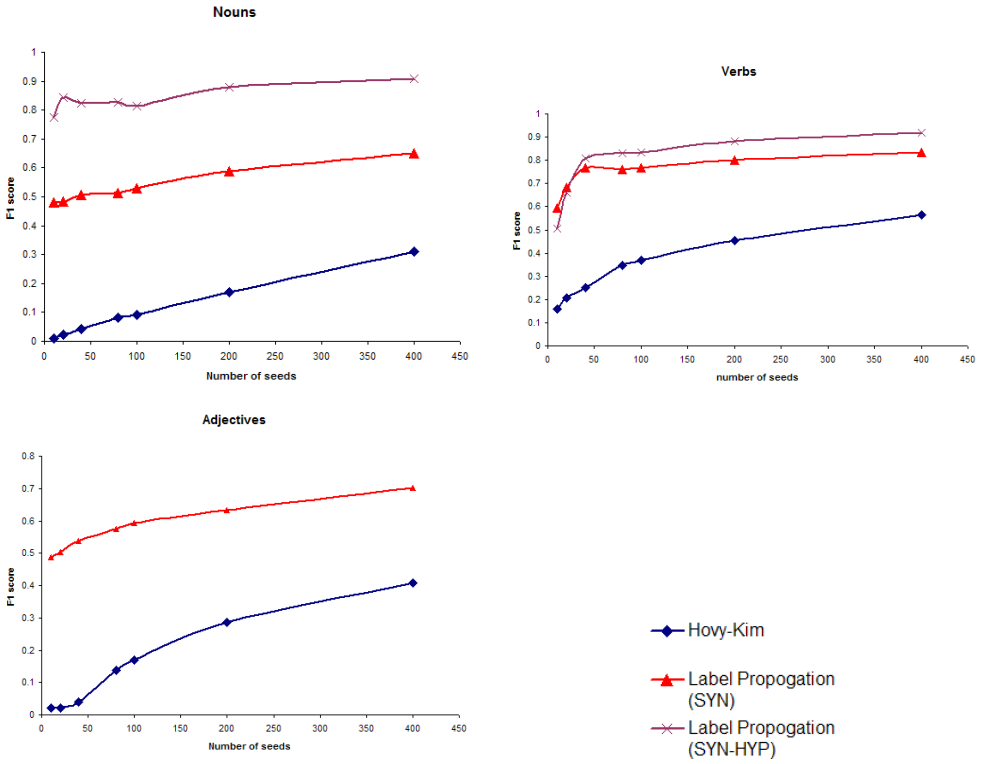

Figure 5: Effect of number of seeds on the F-score for Nouns, Verbs, and Adjectives. The X-axis is number of seeds and the Y-axis is the F-score.

found the cost prohibitive to obtain it. Observe that if we are using only the synonymy relation in WordNet then any thesaurus can be used instead. To demonstrate this, we consider the OpenOffice thesaurus for French, that is freely available. The synonym graph of French adjectives has 9707 vertices and 1.6M edges. We manually annotated a list of 316 adjectives and derived seed and test sets using a 50-50 split. The results of label propagation on such a graph is shown in Table 8.

French	P	R	F1
	73.65	93.67	82.46

Table 8: Evaluation on French dataset

The reason for better results in Hindi compared to French can be attributed to (1) higher inter-annotator agreement ($\kappa = 0.7$) in Hindi compared that in French ($\kappa = 0.55$).[11] (2) The Hindi experiment, like English, used WordNet while the French experiment was performed on graphs derived from the OpenOffice thesaurus due lack of freely available French WordNet.

[11]We do not have κ scores for English dataset derived from the Harvard Inquirer project.

6 Conclusions and Future Work

This paper demonstrated the utility of graph-based semi-supervised learning framework for building sentiment lexicons in a variety of resource availability situations. We explored how the structure of WordNet could be leveraged to derive polarity lexicons. The paper combines, for the first time, relationships like synonymy and hypernymy to improve label propagation results. All of our methods are independent of language as shown in the French and Hindi cases. We demonstrated applicability of our approach on alternative thesaurus-derived graphs when WordNet is not freely available, as in the case of French.

Although our current work uses WordNet and other thesauri, in resource poor situations when only monolingual raw text is available we can perform label propagation on nearest neighbor graphs derived directly from raw text using distributional similarity methods. This is work in progress.

We are also currently working on the possibility of including WordNet relations other than synonymy and hypernymy. One relation that is interesting and useful is antonymy. Antonym edges cannot be added in a straight-forward way to the

graph for label propagation as antonymy encodes negative similarity (or dissimilarity) and the dissimilarity relation is not transitive.

References

[Blum and Chawla2001] Avrim Blum and Shuchi Chawla. 2001. Learning from labeled and unlabeled data using graph mincuts. In *Proc. 18th International Conf. on Machine Learning*, pages 19–26.

[Blum et al.2004] Blum, Lafferty, Rwebangira, and Reddy. 2004. Semi-supervised learning using randomized mincuts. In *Proceedings of the ICML*.

[Esuli and Sebastiani2006] Andrea Esuli and Fabrizio Sebastiani. 2006. Determining term subjectivity and term orientation for opinion mining. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 193–200.

[Hatzivassiloglou and McKeown1997] Vasileios Hatzivassiloglou and Kathleen McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the ACL*, pages 174–181.

[Kaji and Kitsuregawa2007] Nobuhiro Kaji and Masaru Kitsuregawa. 2007. Building lexicon for sentiment analysis from massive collection of HTML documents. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1075–1083.

[Kamps et al.2004] Jaap Kamps, Maarten Marx, R. ort. Mokken, and Maarten de Rijke. 2004. Using WordNet to measure semantic orientation of adjectives. In *Proceedings of LREC-04, 4th International Conference on Language Resources and Evaluation*, volume IV.

[Kim and Hovy2006] Soo-Min Kim and Eduard H. Hovy. 2006. Identifying and analyzing judgment opinions. In *Proceedings of the HLT-NAACL*.

[Lin1998a] Dekang Lin. 1998a. Automatic retrieval and clustering of similar words. In *Proceedings of COLING*, pages 768–774.

[Lin1998b] Dekang Lin. 1998b. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference in Machine Learning*, pages 296–304.

[Mihalcea et al.2007] Rada Mihalcea, Carmen Banea, and Janyce Wiebe. 2007. Learning multilingual subjective language via cross-lingual projections. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 976–983.

[Pang and Lee2004] Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL*, pages 271–278.

[Pedersen et al.2004] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet::similarity - measuring the relatedness of concepts. In *Proceeding of the HLT-NAACL*.

[Riloff et al.2003] Ellen Riloff, Janyce Wiebe, and Theresa Wilson. 2003. Learning subjective nouns using extraction pattern bootstrapping. In *Proceedings of the 7th Conference on Natural Language Learning*, pages 25–32.

[Stone et al.1966] Philip J. Stone, Dexter C. Dunphy, Marshall S. Smith, and Daniel M. Ogilvie. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press.

[Turney and Littman2003] Peter D. Turney and Michael L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21(4):315–346.

[Wiebe2000] Janyce M. Wiebe. 2000. Learning subjective adjectives from corpora. In *Proceedings of the 2000 National Conference on Artificial Intelligence*. AAAI.

[Zhu and Ghahramani2002] Xiaojin Zhu and Zoubin Ghahramani. 2002. Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University.