

Identifying Text Polarity Using Random Walks

Ahmed Hassan

University of Michigan Ann Arbor
Ann Arbor, Michigan, USA
hassanam@umich.edu

Dragomir Radev

University of Michigan Ann Arbor
Ann Arbor, Michigan, USA
radev@umich.edu

Abstract

Automatically identifying the polarity of words is a very important task in Natural Language Processing. It has applications in text classification, text filtering, analysis of product review, analysis of responses to surveys, and mining online discussions. We propose a method for identifying the polarity of words. We apply a Markov random walk model to a large word relatedness graph, producing a polarity estimate for any given word. A key advantage of the model is its ability to accurately and quickly assign a polarity sign and magnitude to any word. The method could be used both in a semi-supervised setting where a training set of labeled words is used, and in an unsupervised setting where a handful of seeds is used to define the two polarity classes. The method is experimentally tested using a manually labeled set of positive and negative words. It outperforms the state of the art methods in the semi-supervised setting. The results in the unsupervised setting is comparable to the best reported values. However, the proposed method is faster and does not need a large corpus.

1 Introduction

Identifying emotions and attitudes from unstructured text is a very important task in Natural Language Processing. This problem has a variety of possible applications. For example, there has been a great body of work for mining product reputation on the Web (Morinaga et al., 2002; Turney, 2002). Knowing the reputation of a product is very important for marketing and customer relation management (Morinaga et al., 2002). Manually handling reviews to identify reputation is a very costly, and

time consuming process given the overwhelming amount of reviews on the Web. A list of words with positive/negative polarity is a very valuable resource for such an application.

Another interesting application is mining online discussions. A threaded discussion is an electronic discussion in which software tools are used to help individuals post messages and respond to other messages. Threaded discussions include e-mails, e-mail lists, bulletin boards, newsgroups, or Internet forums. Threaded discussions act as a very important tool for communication and collaboration in the Web. An enormous number of discussion groups exists on the Web. Millions of users post content to these groups covering pretty much every possible topic. Tracking participant attitude towards different topics and towards other participants is a very interesting task. For example, Tong (2001) presented the concept of sentiment timelines. His system classifies discussion posts about movies as either positive or negative. This is used to produce a plot of the number of positive and negative sentiment messages over time. All those applications could benefit much from an automatic way of identifying semantic orientation of words.

In this paper, we study the problem of automatically identifying semantic orientation of any word by analyzing its relations to other words. Automatically classifying words as either positive or negative enables us to automatically identify the polarity of larger pieces of text. This could be a very useful building block for mining surveys, product reviews and online discussions. We apply a Markov random walk model to a large semantic word graph, producing a polarity estimate for any given word. Previous work on identifying the semantic orientation of words has addressed the problem as both a semi-supervised (Takamura et al., 2005) and an unsupervised (Turney and Littman, 2003) learning problem. In the semi-supervised setting, a training set of labeled words

is used to train the model. In the unsupervised setting, only a handful of seeds is used to define the two polarity classes. The proposed method could be used both in a semi-supervised and in an unsupervised setting. Empirical experiments on a labeled set of words show that the proposed method outperforms the state of the art methods in the semi-supervised setting. The results in the unsupervised setting are comparable to the best reported values. The proposed method has the advantages that it is faster and it does not need a large training corpus.

The rest of the paper is structured as follows. In Section 2, we discuss related work. Section 3 presents our method for identifying word polarity. Section 4 describes our experimental setup. We conclude in Section 5.

2 Related Work

Hatzivassiloglou and McKeown (1997) proposed a method for identifying word polarity of adjectives. They extract all conjunctions of adjectives from a given corpus and then they classify each conjunctive expression as either the same orientation such as “simple and well-received” or different orientation such as “simplistic but well-received”. The result is a graph that they cluster into two subsets of adjectives. They classify the cluster with the higher average frequency as positive. They created and labeled their own dataset for experiments. Their approach will probably work only with adjectives because there is nothing wrong with conjunctions of nouns or verbs with opposite polarities (e.g., “war and peace”, “rise and fall”, ..etc).

Turney and Littman (2003) identify word polarity by looking at its statistical association with a set of positive/negative seed words. They use two statistical measures for estimating association: Pointwise Mutual Information (PMI) and Latent Semantic Analysis (LSA). To get co-occurrence statistics, they submit several queries to a search engine. Each query consists of the given word and one of the seed words. They use the search engine near operator to look for instances where the given word is physically close to the seed word in the returned document. They present their method as an unsupervised method where a very small amount of seed words are used to define semantic orientation rather than train the model. One of the limitations of their method is that it requires a large

corpus of text to achieve good performance. They use several corpora, the size of the best performing dataset is roughly one hundred billion words (Turney and Littman, 2003).

Takamura et al. (2005) proposed using spin models for extracting semantic orientation of words. They construct a network of words using gloss definitions, thesaurus, and co-occurrence statistics. They regard each word as an electron. Each electron has a spin and each spin has a direction taking one of two values: up or down. Two neighboring spins tend to have the same orientation from an energetic point of view. Their hypothesis is that as neighboring electrons tend to have the same spin direction, neighboring words tend to have similar polarity. They pose the problem as an optimization problem and use the mean field method to find the best solution. The analogy with electrons leads them to assume that each word should be either positive or negative. This assumption is not accurate because most of the words in the language do not have any semantic orientation. They report that their method could get misled by noise in the gloss definition and their computations sometimes get trapped in a local optimum because of its greedy optimization flavor.

Kamps et al. (2004) construct a network based on WordNet synonyms and then use the shortest paths between any given word and the words ‘good’ and ‘bad’ to determine word polarity. They report that using shortest paths could be very noisy. For example, ‘good’ and ‘bad’ themselves are closely related in WordNet with a 5-long sequence “good, sound, heavy, big, bad”. A given word w may be more connected to one set of words (e.g., positive words), yet have a shorter path connecting it to one word in the other set. Restricting seed words to only two words affects their accuracy. Adding more seed words could help but it will make their method extremely costly from the computation point of view. They evaluate their method only using adjectives.

Hu and Liu (2004) use WordNet synonyms and antonyms to predict the polarity of words. For any word, whose polarity is unknown, they search WordNet and a list of seed labeled words to predict its polarity. They check if any of the synonyms of the given word has known polarity. If so, they label it with the label of its synonym. Otherwise, they check if any of the antonyms of the given word has known polarity. If so, they label it

with the opposite label of the antonym. They continue in a bootstrapping manner till they label all possible word. This method is quite similar to the shortest-path method proposed in (Kamps et al., 2004).

There are some other methods that try to build lexicons of polarized words. Esuli and Sebastiani (2005; 2006) use a textual representation of words by collating all the glosses of the word as found in some dictionary. Then, a binary text classifier is trained using the textual representation and applied to new words. Kim and Hovy (2004) start with two lists of positive and negative seed words. WordNet is used to expand these lists. Synonyms of positive words and antonyms of negative words are considered positive, while synonyms of negative words and antonyms of positive words are considered negative. A similar method is presented in (Andreevskaia and Bergler, 2006) where WordNet synonyms, antonyms, and glosses are used to iteratively expand a list of seeds. The sentiment classes are treated as fuzzy categories where some words are very central to one category, while others may be interpreted differently. Kanayama and Nasukawa (2006) use syntactic features and context coherency, the tendency for same polarities to appear successively, to acquire polar atoms.

Other related work is concerned with subjectivity analysis. Subjectivity analysis is the task of identifying text that present opinions as opposed to objective text that present factual information (Wiebe, 2000). Text could be either words, phrases, sentences, or any other chunks. There are two main categories of work on subjectivity analysis. In the first category, subjective words and phrases are identified without considering their context (Wiebe, 2000; Hatzivassiloglou and Wiebe, 2000; Banea et al., 2008). In the second category, the context of subjective text is used (Riloff and Wiebe, 2003; Yu and Hatzivassiloglou, 2003; Nasukawa and Yi, 2003; Popescu and Etzioni, 2005) Wiebe et al. (2001) lists a lot of applications of subjectivity analysis such as classifying emails and mining reviews. Subjectivity analysis is related to the proposed method because identifying the polarity of text is the natural next step that should follow identifying subjective text.

3 Word Polarity

We use a Markov random walk model to identify polarity of words. Assume that we have a network

of words, some of which are labeled as either positive or negative. In this network, two words are connecting if they are related. Different sources of information could be used to decide whether two words are related or not. For example, the synonyms of any word are semantically related to it. The intuition behind that connecting semantically related words is that those words tend to have similar polarity. Now imagine a random surfer walking along the network starting from an unlabeled word w . The random walk continues until the surfer hits a labeled word. If the word w is positive then the probability that the random walk hits a positive word is higher and if w is negative then the probability that the random walk hits a negative word is higher. Similarly, if the word w is positive then the average time it takes a random walk starting at w to hit a positive node is less than the average time it takes a random walk starting at w to hit a negative node.

In the rest of this section, we will describe how we can construct a word relatedness graph in Section 3.1. The random walk model is described in Section 3.2. Hitting time is defined in Section 3.3. Finally, an algorithm for computing a sign and magnitude for the polarity of any given word is described in Section 3.4.

3.1 Network Construction

We construct a network where two nodes are linked if they are semantically related. Several sources of information could be used as indicators of the relatedness of words. One such important source is WordNet (Miller, 1995). WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept (Miller, 1995). Synsets are inter-linked by means of conceptual-semantic and lexical relations.

The simplest approach is to connect words that occur in the same WordNet synset. We can collect all words in WordNet, and add links between any two words that occur in the same synset. The resulting graph is a graph $G(W, E)$ where W is a set of word / part-of-speech pairs for all the words in WordNet. E is the set of edges connecting each pair of synonymous words. Nodes represent word/pos pairs rather than words because the part of speech tags are helpful in disambiguating the different senses for a given word. For example,

the word “fine” has two different meanings when used as an adjective and as a noun.

Several other methods could be used to link words. For example, we can use other WordNet relations: hypernyms, similar to,...etc. Another source of links between words is co-occurrence statistics from corpus. Following the method presented in (Hatzivassiloglou and McKeown, 1997), we can connect words if they appear in a conjunctive form in the corpus. This method is only applicable to adjectives. If two adjectives are connected by “and” in conjunctive form, it is highly likely that they have the same semantic orientation. In all our experiments, we restricted the network to only WordNet relations. We study the effect of using co-occurrence statistics to connect words later at the end of our experiments. If more than one relation exists between any two words, the strength of the corresponding edge is adjusted accordingly.

3.2 Random Walk Model

Imagine a random surfer walking along the word relatedness graph G . Starting from a word with unknown polarity i , it moves to a node j with probability P_{ij} after the first step. The walk continues until the surfer hits a word with a known polarity. Seed words with known polarity act as an absorbing boundary for the random walk. If we repeat the number of random walks N times, the percentage of time at which the walk ends at a positive/negative word could be used as an indicator of its positive/negative polarity. The average time a random walk starting at w takes to hit the set of positive/negative nodes is also an indicator of its polarity. This view is closely related to the partially labeled classification with random walks approach in (Szummer and Jaakkola, 2002) and the semi-supervised learning using harmonic functions approach in (Zhu et al., 2003).

Let W be the set of words in our lexicon. We construct a graph whose nodes V are all words in W . The edges E correspond to relatedness between words. We define transition probabilities $P_{t+1|t}(j|i)$ from i to j by normalizing the weights of the edges out of node i , so:

$$P_{t+1|t}(j|i) = W_{ij} / \sum_k W_{ik} \quad (1)$$

where k represents all nodes in the neighborhood of i . $P_{t_2|t_1}(j|i)$ denotes the transition probability from node i at step t_1 to node j at time step t_2 . We note that the weights W_{ij} are symmetric and

the transition probabilities $P_{t+1|t}(j|i)$ are not necessarily symmetric because of the node out degree normalization.

3.3 First-Passage Time

The mean first-passage (hitting) time $h(i|k)$ is defined as the average number of steps a random walker, starting in state $i \neq k$, will take to enter state k for the first time (Norris, 1997). Let $G = (V, E)$ be a graph with a set of vertices V , and a set of edges E . Consider a subset of vertices $S \subset V$. Consider a random walk on G starting at node $i \notin S$. Let N_t denote the position of the random surfer at time t . Let $h(i|S)$ be the the average number of steps a random walker, starting in state $i \notin S$, will take to enter a state $k \in S$ for the first time. Let T^S be the first-passage for any vertex in S .

$$P(T_S = t | N_0 = i) = \sum_{j \in V} p_{ij} \times P(T_S = t - 1 | N_0 = j) \quad (2)$$

$h(i|S)$ is the expectation of T_S . Hence:

$$\begin{aligned} h(i|S) &= E(T_S | N_0 = i) \\ &= \sum_{t=1}^{\infty} t \times P(T_S = t | N_0 = i) \\ &= \sum_{t=1}^{\infty} t \sum_{j \in V} p_{ij} P(T_S = t - 1 | N_0 = j) \\ &= \sum_{j \in V} \sum_{t=1}^{\infty} (t - 1) p_{ij} P(T_S = t - 1 | N_0 = j) \\ &\quad + \sum_{j \in V} \sum_{t=1}^{\infty} p_{ij} P(T_S = t - 1 | N_0 = j) \\ &= \sum_{j \in V} p_{ij} \sum_{t=1}^{\infty} t P(T_S = t | N_0 = j) + 1 \\ &= \sum_{j \in V} p_{ij} \times h(j|S) + 1 \end{aligned} \quad (3)$$

Hence the first-passage (hitting) time can be formally defined as:

$$h(i|S) = \begin{cases} 0 & i \in S \\ \sum_{j \in V} p_{ij} \times h(j|S) + 1 & \text{otherwise} \end{cases} \quad (4)$$

3.4 Word Polarity Calculation

Based on the description of the random walk model and the first-passage (hitting) time above,

we now propose our word polarity identification algorithm. We begin by constructing a word relatedness graph and defining a random walk on that graph as described above. Let S^+ and S^- be two sets of vertices representing seed words that are already labeled as either positive or negative respectively. For any given word w , we compute the hitting time $h(w|S^+)$, and $h(w|S^-)$ for the two sets iteratively as described earlier. If $h(w|S^+)$ is greater than $h(w|S^-)$, the word is classified as negative, otherwise it is classified as positive. The ratio between the two hitting times could be used as an indication of how positive/negative the given word is. This is useful in case we need to provide a confidence measure for the prediction. This could be used to allow the model to abstain from classifying words with when the confidence level is low.

Computing hitting time as described earlier may be time consuming especially if the graph is large. To overcome this problem, we propose a Monte Carlo based algorithm for estimating it. The algorithm is shown in Algorithm 1.

Algorithm 1 Word Polarity using Random Walks

Require: A word relatedness graph G

- 1: Given a word w in V
 - 2: Define a random walk on the graph. the transition probability between any two nodes i , and j is defined as: $P_{t+1|t}(j|i) = W_{ij} / \sum_k W_{ik}$
 - 3: Start k independent random walks from w with a maximum number of steps m
 - 4: Stop when a positive word is reached
 - 5: Let $h^*(w|S^+)$ be the estimated value for $h(w|S^+)$
 - 6: Repeat for negative words computing $h^*(w|S^-)$
 - 7: **if** $h^*(w|S^+) \leq h^*(w|S^-)$ **then**
 - 8: Classify w as positive
 - 9: **else**
 - 10: Classify w as negative
 - 11: **end if**
-

4 Experiments

We performed experiments on the General Inquirer lexicon (Stone et al., 1966). We used it as a gold standard data set for positive/negative words. The dataset contains 4206 words, 1915 of which are positive and 2291 are negative. Some of the ambiguous words were removed like (Turney, 2002; Takamura et al., 2005).

We use WordNet (Miller, 1995) as a source of synonyms and hypernyms for the word relatedness graph. We used 10-fold cross validation for all tests. We evaluate our results in terms of accuracy. Statistical significance was tested using a 2-tailed paired t-test. All reported results are statistically significant at the 0.05 level. We perform experiments varying the parameters and the network. We also look at the performance of the proposed method for different parts of speech, and for different confidence levels. We compare our method to the Semantic Orientation from PMI (SO-PMI) method described in (Turney, 2002), the Spin model (Spin) described in (Takamura et al., 2005), the shortest path (short-path) described in (Kamps et al., 2004), and the bootstrapping (bootstrap) method described in (Hu and Liu, 2004).

4.1 Comparisons with other methods

This method could be used in a semi-supervised setting where a set of labeled words are used and the system learns from these labeled nodes and from other unlabeled nodes. Under this setting, we compare our method to the spin model described in (Takamura et al., 2005). Table 2 compares the performance using 10-fold cross validation. The table shows that the proposed method outperforms the spin model. The spin model approach uses word glosses, WordNet synonym, hypernym, and antonym relations, in addition to co-occurrence statistics extracted from corpus. The proposed method achieves better performance by only using WordNet synonym, hypernym and similar to relations. Adding co-occurrence statistics slightly improved performance, while using glosses did not help at all.

We also compare our method to the SO-PMI method presented in (Turney, 2002). They describe this setting as unsupervised (Turney, 2002) because they only use 14 seeds as paradigm words that define the semantic orientation rather than train the model. After (Turney, 2002), we use our method to predict semantic orientation of words in the General Inquirer lexicon (Stone et al., 1966) using only 14 seed words. The network we used contains only WordNet relations. No glosses or co-occurrence statistics are used. The results comparing the SO-PMI method with different dataset sizes, the spin model, and the proposed method using only 14 seeds is shown in Table 2. We no-

Table 1: Accuracy for adjectives only for the spin model, the bootstrap method, and the random walk model.

spin-model	bootstrap	short-path	rand-walks
83.6	72.8	68.8	88.8

tice that the random walk method outperforms SO-PMI when SO-PMI uses datasets of sizes 1×10^7 and 2×10^9 words. The performance of SO-PMI and the random walk methods are comparable when SO-PMI uses a very large dataset (1×10^{11} words). The performance of the spin model approach is also comparable to the other 2 methods. The advantages of the random walk method over SO-PMI is that it is faster and it does not need a very large corpus like the one used by SO-PMI. Another advantage is that the random walk method can be used along with the labeled data from the General Inquirer lexicon (Stone et al., 1966) to get much better performance. This is costly for the SO-PMI method because that will require the submission of almost 4000 queries to a commercial search engine.

We also compare our method to the bootstrapping method described in (Hu and Liu, 2004), and the shortest path method described in (Kamps et al., 2004). We build a network using only WordNet synonyms and hypernyms. We restrict the test set to the set of adjectives in the General Inquirer lexicon (Stone et al., 1966) because this method is mainly interested in classifying adjectives. The performance of the spin model method, the bootstrapping method, the shortest path method, and the random walk method for only adjectives is shown in Table 1. We notice from the table that the random walk method outperforms both the spin model, the bootstrapping method, and the shortest path method for adjectives. The reported accuracy for the shortest path method only considers the words it could assign a non-zero orientation value. If we consider all words, the accuracy will drop to around 61%.

4.1.1 Varying Parameters

As we mentioned in Section 3.4, we use a parameter m to put an upper bound on the length of random walks. In this section, we explore the impact

Table 2: Accuracy for SO-PMI with different dataset sizes, the spin model, and the random walks model for 10-fold cross validation and 14 seeds.

-	CV	14 seeds
SO-PMI (1×10^7)	-	61.3
SO-PMI (2×10^9)	-	76.1
SO-PMI (1×10^{11})	-	82.8
Spin Model	91.5	81.9
Random Walks	93.1	82.1

of this parameter on our method’s performance.

Figure 1 shows the accuracy of the random walk method as a function of the maximum number of steps m . m varies from 5 to 50. We use a network built from WordNet synonyms and hypernyms only. The number of samples k was set to 1000. We perform 10-fold cross validation using the General Inquirer lexicon. We notice that the maximum number of steps m has very little impact on performance until it rises above 30. When it does, the performance drops by no more than 1%, and then it does not change anymore as m increases. An interesting observation is that the proposed method performs quite well with a very small number of steps (around 10). We looked at the dataset to understand why increasing the number of steps beyond 30 negatively affects performance. We found out that when the number of steps is very large, compared to the diameter of the graph, the random walk that starts at ambiguous words, that are hard to classify, have the chance of moving till it hits a node in the opposite class. That does not happen when the limit on the number of steps is smaller because those walks are then terminated without hitting any labeled nodes and hence ignored.

Next, we study the effect of the random of samples k on our method’s performance. As explained in Section 3.4, k is the number of samples used by the Monte Carlo algorithm to find an estimate for the hitting time. Figure 2 shows the accuracy of the random walks method as a function of the number of samples k . We use the same settings as in the previous experiment. the only difference is that we fix m at 15 and vary k from 10 to 20000 (note the logarithmic scale). We notice that the performance is badly affected, when the value of k is very small (less than 100). We also notice that

after 1000, varying k has very little, if any, effect on performance. This shows that the Monte Carlo algorithm for computing the random walks hitting time performs quite well with values of the number of samples as small as 1000.

The preceding experiments suggest that the parameter have very little impact on performance. This suggests that the approach is fairly robust (i.e., it is quite insensitive to different parameter settings).

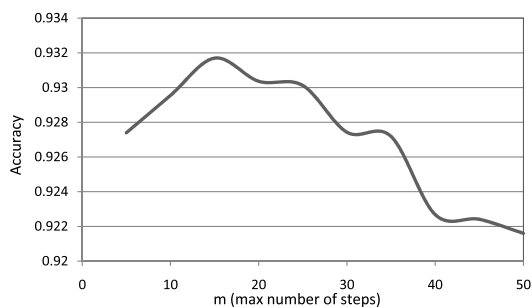


Figure 1: The effect of varying the maximum number of steps (m) on accuracy.

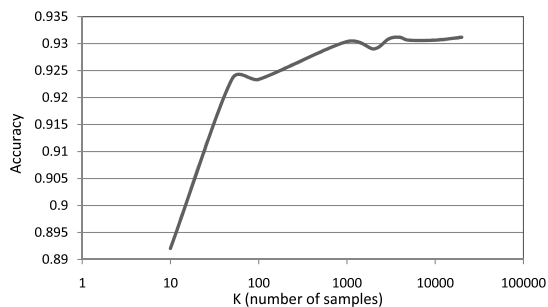


Figure 2: The effect of varying the number of samples (k) on accuracy.

4.1.2 Other Experiments

We now measure the performance of the proposed method when the system is allowed to abstain from classifying the words for which it have low confidence. We regard the ratio between the hitting time to positive words and hitting time to negative words as a confidence measure and evaluate the top words with the highest confidence level at different values of threshold. Figure 4 shows the accuracy for 10-fold cross validation and for using only 14 seeds at different thresholds. We notice that the accuracy improves by abstaining from classifying the difficult words. The figure shows

that the top 60% words are classified with an accuracy greater than 99% for 10-fold cross validation and 92% with 14 seed words. This may be compared to the work described in (Takamura et al., 2005) where they achieve the 92% level when they only consider the top 1000 words (28%).

Figure 3 shows a learning curve displaying how the performance of the proposed method is affected with varying the labeled set size (i.e., the number of seeds). We notice that the accuracy exceeds 90% when the training set size rises above 20%. The accuracy steadily increases as the labeled data increases.

We also looked at the classification accuracy for different parts of speech in Figure 5. we notice that, in the case of 10-fold cross validation, the performance is consistent across parts of speech. However, when we only use 14 seeds all of which are adjectives, similar to (Turney and Littman, 2003), we notice that the performance on adjectives is much better than other parts of speech. When we use 14 seeds but replace some of the adjectives with verbs and nouns like (love, harm, friend, enemy), the performance for nouns and verbs improves considerably at the cost of losing a little bit of the performance on adjectives. We had a closer look at the results to find out what are the reasons behind incorrect predictions. We found two main reasons. First, some words are ambiguous and has more than one sense, possible with different orientations. Disambiguating the sense of words given their context before trying to predict their polarity should solve this problem. The second reason is that some words have very few connection in thesaurus. A possible solution to this might be identifying those words and adding more links to them from glosses of co-occurrence statistics in corpus.

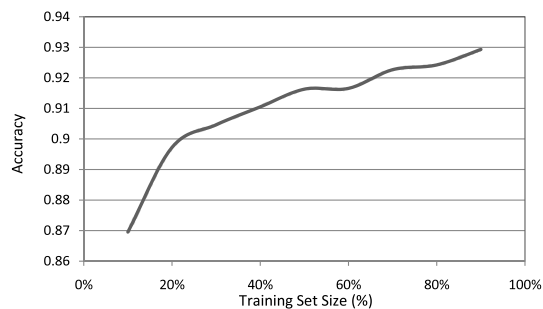


Figure 3: The effect of varying the number of seeds on accuracy.

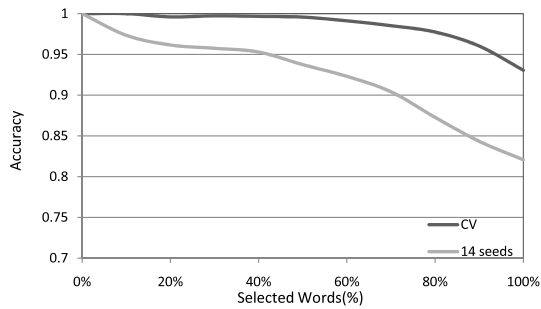


Figure 4: Accuracy for words with high confidence measure.

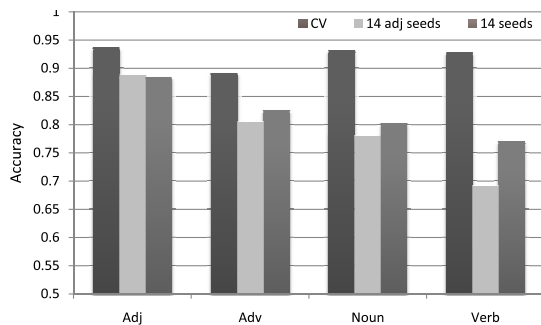


Figure 5: Accuracy for different parts of speech.

5 Conclusions

Predicting the semantic orientation of words is a very interesting task in Natural Language Processing and it has a wide variety of applications. We proposed a method for automatically predicting the semantic orientation of words using random walks and hitting time. The proposed method is based on the observation that a random walk starting at a given word is more likely to hit another word with the same semantic orientation before hitting a word with a different semantic orientation. The proposed method can be used in a semi-supervised setting where a training set of labeled words is used, and in an unsupervised setting where only a handful of seeds is used to define the two polarity classes. We predict semantic orientation with high accuracy. The proposed method is fast, simple to implement, and does not need any corpus.

Acknowledgments

This research was funded by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), through the U.S. Army Research Lab. All statements of fact, opinion or conclusions con-

tained herein are those of the authors and should not be construed as representing the official views or policies of IARPA, the ODNI or the U.S. Government.

References

- Alina Andreevskaia and Sabine Bergler. 2006. Mining wordnet for fuzzy sentiment: Sentiment tag extraction from wordnet glosses. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2006)*.
- Carmen Banea, Rada Mihalcea, and Janyce Wiebe. 2008. A bootstrapping method for building subjectivity lexicons for languages with scarce resources. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*.
- Andrea Esuli and Fabrizio Sebastiani. 2005. Determining the semantic orientation of terms through gloss classification. In *Proceedings of the 14th Conference on Information and Knowledge Management (CIKM 2005)*, pages 617–624.
- Andrea Esuli and Fabrizio Sebastiani. 2006. Sentimentwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of the 5th Conference on Language Resources and Evaluation (LREC 2006)*, pages 417–422.
- 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*, pages 174–181.
- Vasileios Hatzivassiloglou and Janyce Wiebe. 2000. Effects of adjective orientation and gradability on sentence subjectivity. In *COLING*, pages 299–305.
- Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.
- Jaap Kamps, Maarten Marx, Robert J. Mokken, and Maarten De Rijke. 2004. Using wordnet to measure semantic orientations of adjectives. In *National Institute for*, pages 1115–1118.
- Hiroshi Kanayama and Tetsuya Nasukawa. 2006. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, pages 355–363.
- Soo-Min Kim and Eduard Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of the 20th international conference on Computational Linguistics (COLING 2004)*, pages 1367–1373.

- George A. Miller. 1995. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41.
- Satoshi Morinaga, Kenji Yamanishi, Kenji Tateishi, and Toshikazu Fukushima. 2002. Mining product reputations on the web. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 341–349.
- Tatsuya Nasukawa and Jeonghee Yi. 2003. Sentiment analysis: capturing favorability using natural language processing. In *K-CAP '03: Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77.
- J. Norris. 1997. Markov chains. Cambridge University Press.
- Ana-Maria Popescu and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 339–346.
- Ellen Riloff and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 105–112.
- Philip Stone, Dexter Dunphy, Marchall Smith, and Daniel Ogilvie. 1966. The general inquirer: A computer approach to content analysis. *The MIT Press*.
- Martin Szummer and Tommi Jaakkola. 2002. Partially labeled classification with markov random walks. In *Advances in Neural Information Processing Systems*, pages 945–952.
- Hiroya Takamura, Takashi Inui, and Manabu Okumura. 2005. Extracting semantic orientations of words using spin model. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 133–140.
- Richard M. Tong. 2001. An operational system for detecting and tracking opinions in on-line discussion. Workshop note, SIGIR 2001 Workshop on Operational Text Classification.
- Peter Turney and Michael Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21:315–346.
- Peter D. Turney. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 417–424.
- Janyce Wiebe, Rebecca Bruce, Matthew Bell, Melanie Martin, and Theresa Wilson. 2001. A corpus study of evaluative and speculative language. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*, pages 1–10.
- Janyce Wiebe. 2000. Learning subjective adjectives from corpora. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 735–740.
- Hong Yu and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 129–136.
- Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *In ICML*, pages 912–919.