# A Random Walk–Based Model for Identifying Semantic Orientation

Ahmed Hassan*
Microsoft Research

Amjad Abu-Jbara**
University of Michigan

Wanchen Lu†
University of Michigan

Dragomir Radev‡
University of Michigan

*Automatically identifying the sentiment polarity of words is a very important task that has been used as the essential building block of many natural language processing systems such as text classification, text filtering, product review analysis, survey response analysis, and on-line discussion mining. We propose a method for identifying the sentiment polarity of words that applies a Markov random walk model to a large word relatedness graph, and produces a polarity estimate for any given word. The model can accurately and quickly assign a polarity sign and magnitude to any word. It can be used both in a semi-supervised setting where a training set of labeled words is used, and in a weakly supervised setting where only a handful of seed words is used to define the two polarity classes. The method is experimentally tested using a gold standard set of positive and negative words from the General Inquirer lexicon. We also show how our method can be used for three-way classification which identifies neutral words in addition to positive and negative words. Our experiments show that the proposed method outperforms the state-of-the-art methods in the semi-supervised setting and is comparable to the best reported values in the weakly supervised setting. In addition, the proposed method is faster and does not need a large corpus. We also present extensions of our methods for identifying the polarity of foreign words and out-of-vocabulary words.*

* Microsoft Research, Redmond, WA, USA. E-mail: hassanam@microsoft.com. This research was performed while at the University of Michigan.
** Department of Electrical Engineering & Computer Science, University of Michigan, Ann Arbor, MI, USA. E-mail: amjbara@umich.edu.
† Department of Electrical Engineering & Computer Science, University of Michigan, Ann Arbor, MI, USA. E-mail: wanchlu@umich.edu.
‡ Department of Electrical Engineering & Computer Science and School of Information, University of Michigan, Ann Arbor, MI, USA. E-mail: radev@umich.edu.

Submission received: 15 November 2011; revised submission received: 10 May 2013; accepted for publication: 14 July 2013.

doi:10.1162/COLI_a_00192

## 1. Introduction

Identifying emotions and attitudes from unstructured text has a variety of possible applications. For example, there has been a large body of work for mining product reputation on the Web (Morinaga et al. 2002; Turney 2002). Morinaga et al. (2002) have shown how product reputation mining helps with marketing and customer relation management. The Google products catalog and many on-line shopping sites like Amazon.com provide customers not only with comprehensive information and reviews about a product, but also with faceted sentiment summaries. Such systems are all supported by a sentiment lexicon, some even in multiple languages.

Another interesting application is mining on-line discussions. An enormous number of discussion groups exist on the Web. Millions of users post content to these groups covering pretty much every possible topic. Tracking a participant attitude toward different topics and toward other participants is a very important task that makes use of sentiment lexicons. 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 these applications would benefit from an automatic way of identifying semantic orientation of words.

In this article, we study the task of automatically identifying the semantic orientation of any word by analyzing its relations to other words, Automatically classifying words as positive, negative, or neutral enables us to automatically identify the polarity of larger pieces of text. This could be a very useful building block for systems that mine surveys, product reviews, and on-line discussions. We apply a Markov random walk model to a large semantic relatedness 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, Inui, and Okumura 2005) and a weakly supervised (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 weakly supervised setting, only a handful of seeds are used to define the two polarity classes.

Our proposed method can be used both in a semi-supervised and in a weakly supervised setting. Empirical experiments on a labeled set of positive and negative words show that the proposed method outperforms the state-of-the-art methods in the semi-supervised setting. The results in the weakly supervised setting are comparable to the best reported values. The proposed method has the advantages that it is faster and does not need a large training corpus.

The rest of the article is structured as follows. In Section 2, we review related work on word polarity and subjectivity classification and note applications of the random walk and hitting times framework. Section 3 presents our method for identifying word polarity. We describe how the proposed method can be extended to cover foreign languages in Section 4, and out-of-vocabulary words in Section 5. Section 6 describes our experimental set-up. We present our conclusions in Section 7.

## 2. Related Work

### 2.1 Identifying Word Polarity

Hatzivassiloglou and McKeown (1997) proposed a method for identifying the 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 data set for experiments. Their approach works only with adjectives because there is nothing wrong with conjunctions of nouns or verbs with opposite polarities ("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 number 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 data set is roughly one hundred billion words (Turney and Littman 2003).

Takamura et al. (2005) propose 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 energy 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 (Miller 1995) 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 using only adjectives.

Hu and Liu (2004) propose another method that uses WordNet. They 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 until they label all possible words.

### 2.2 Building Sentiment Lexicons

A number of other methods 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, and 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, whereas others may be interpreted differently.

Mohammad, Dunne, and Dorr (2009) utilize the marking theory, which states that overtly marked words such as *dishonest, unhappy*, and *impure* tend to have negative semantic orientations whereas their unmarked counterparts (*honest*, *happy*, and *pure*) tend to have positive semantic orientation. They use a set of 11 antonym-generating affix patterns to generate overtly marked words and their counterparts from the *Macquarie Thesaurus*. After obtaining a set of 2,600 seeds by the affix patterns, they expand the sentiment lexicon using a *Roget*-like thesaurus. Their method does not require seed sentiment words or WordNet, but still needs a comprehensive thesaurus. The idea of the marking theory is language-dependent and cannot be applied from one language to another.

Contrasting the dictionary based approaches that rely on resources such as WordNet, Velikovich et al. (2010) investigated the viability of learning sentiment lexicons semi-automatically from the Web. Kanayama and Nasukawa (2006) use syntactic features and context coherency (i.e., the tendency for same polarities to appear successively) to detect polar clauses.

### 2.3 Random Walk–Based Methods

Closest to our work in its methodology is probably the line of research on semi-supervised graphical methods for sentiment classification. Rao and Ravichandran (2009) build a lexical graph similar to ours. The graph is constructed of both unlabeled and labeled nodes, each node representing a word that can be either positive or negative, and each edge representing some semantic relatedness that can be constructed using resources like WordNet or other thesaurus. They evaluate two semi-supervised methods: Mincut (including its variant, Randomized Mincut) and label propagation. The general idea of label propagation is defining a probability distribution over the positive and negative classes for each node in the graph. A Markov random walk is performed on the graph to recover this distribution for the unlabeled nodes.

Additionally, Rao and Ravichandra (2009) and Blair-Goldensohn et al. (2008) use a similar label propagation method on a lexical graph built from WordNet, where a small set of words with known polarities are used as seeds. Brody and Elhadad (2010) use label propagation over a graph constructed of adjectives only.

Velikovich et al. (2010) compare label propagation with a Web-based method and conclude that label propagation is not suitable when the whole Web is used as a background corpus, because the constructed graph is very noisy and contains many dense subgraphs, unlike the lexical graph constructed from WordNet.

Random walk–based methods have been studied in the context of many other NLP tasks. For example, Kok and Brockett (2010) construct a graph from bilingual parallel corpora, where each node represents a phrase and two nodes are connected by an edge if they are aligned in a phrase table. Then they compute hitting time of random walks to learn paraphrases.

Our work is different from previous random walk methods in that it uses the mean hitting time as the criterion for assigning polarity labels. Our experiments showed that this achieves better results than methods that use label propagation.

### 2.4 Subjectivity Analysis

Subjectivity analysis is another research line that is closely related to our work. The main task in subjectivity analysis is to identify text that presents opinion as opposed to objective text that present factual information (Wiebe 2000). Text could be either words, phrases, sentences, or other chunks. Wiebe et al. (2001) list a number of applications of subjectivity analysis such as classifying e-mails and mining reviews. For example, to analyze movie reviews, Pang and Lee (2004) apply Mincut to a graph constructed from individual sentences as nodes to determine whether a sentence is subjective or objective. Each node (sentence) has an individual subjectivity score obtained from a first-pass classifier using sentence features and linguistic knowledge. Edges are weighted by a similarity metric of how likely it is that the two sentences will be in the same subjectivity class. All sentences to be classified are represented as unlabeled nodes and the only two labeled nodes represent the subjective and objective classes. A Mincut algorithm is then performed on the constructed graph to obtain the subjectivity classes for individual sentences. The authors also integrate the subjectivity classification of isolated sentences to document level sentiment analysis.

There are two main categories of work on subjectivity analysis. In the first category, subjective words and phrases are identified without considering their context (Hatzivassiloglou and Wiebe 2000; Wiebe 2000; Banea, Mihalcea, and Wiebe 2008). In the second category, the context of subjective text is used (Nasukawa and Yi 2003; Riloff and Wiebe 2003; Yu and Hatzivassiloglou 2003; Popescu and Etzioni 2005). Wiebe and Mihalcea (2006a) studied the association of word subjectivity and word sense. They showed that different subjectivity labels can be assigned to different senses of the same word. Wiebe, Wilson, and Cardie (2005) described MPQA, a corpus of news articles from a wide variety of news sources manually annotated for opinions and other private states (i.e., beliefs, emotions, sentiments, speculations) directed for studying opinions and emotions in language.

In addition, there has been a large body of work on labeling subjectivity of WordNet words. Wiebe and Mihalcea (2006b) label word senses in WordNet as subjective or objective, utilizing the MPQA corpus. They show that subjectivity information for WordNet senses can improve word sense disambiguation tasks for subjectivity ambiguous words.

Su and Markert (2009) propose a semi-supervised minimum cut framework to label word sense entries in WordNet with subjectivity information. Their method requires less training data other than the sense definitions and relational structure of WordNet.

### 2.5 Word Polarity Classification for Foreign Languages

Word sentiment and subjectivity has also been studied for languages other than English. Jijkoun and Hofmann (2009) describe a method for creating a non-English subjectivity

lexicon based on an English lexicon, an on-line translation service, and Wordnet. Mihalcea and Banea (2007) use bilingual resources such as a bilingual dictionary or a parallel corpus to generate subjectivity analysis resources for foreign languages. Rao and Ravichandran (2009) adapt their label propagation model to Hindi using Hindi WordNet and French using a French thesaurus.

## 3. Approach

We use a Markov random walk model to identify the 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 connected if they are related. Different sources of information are used to decide whether two words are related. For example, the synonyms of a word are all semantically related to it. The intuition behind connecting semantically related words is that those words tend to have similar polarities. 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. Thus, if the word $w$ is positive then the average time it takes a random walk starting at $w$ to hit a positive node should be much less than the average time it takes a random walk starting at $w$ to hit a negative node. If $w$ doesn't have a clear polarity and we would like to say that it is neutral, we expect that the positive hitting time and negative hitting time to not have a significant difference.

We describe how we 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 are used as indicators of the relatedness of words. One such 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 interlinked 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 (POS) 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, with two opposite polarities when used as an adjective and as a noun.

Several other methods can be used to link words. For example, we can use other WordNet relations: hypernyms, similar to, and so forth. Another source of links between words is co-occurrence statistics from a corpus. Following the method presented

in Hatzivassiloglou and McKeown (1997), we can connect words if they appear together in a conjunction in the corpus. This method is only applicable to adjectives. If two adjectives are connected by "and," 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 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 times in 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, Ghahramani, and Lafferty (2003).

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

$$P_{t+1|t}(j|i) = Wij / \sum_{k} W_{ik} \tag{1}$$

where $k$ represents all nodes in the neighborhood of $i$. $P_{t+1|t}(j|i)$ denotes the transition probability from node $i$ at step $t$ to node $j$ at time step $t+1$. We note that the matrix of weights $W_{ij}$ is symmetric whereas the matrix of transition probabilities $P_{t+1|t}(j|i)$ is not necessarily symmetric because of the node outdegree 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 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 $S$.

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

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

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

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

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

### 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. The ratio between the two hitting times is then 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 when the confidence level is low. It also means that our method can be easily extended from two-way classification (i.e., positive or negative) to three-way classification (positive, negative, or neutral). This can be done by setting a threshold $\gamma$ on the ratio of positive and negative hitting time, and classifying a word to positive or negative only when the two hitting times have a significant difference; otherwise we classify it to neutral.

When the relatedness graph is very large, computing hitting time as described earlier may be very time consuming. The graph constructed from the English WordNet

**Algorithm 1** 3-class word polarity using random walks (parameter $\gamma : 0 < \gamma < 1$)

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

and synsets contains 155,000 nodes and 117,000 edges. To overcome this problem, we propose a Monte Carlo–based algorithm (Algorithm 1) for estimating it.

In the case of binary classification, where each word must be either positive or negative, if $h(w|S^+)$ is greater than $h(w|S^-)$, the word is classified as negative and positive otherwise. This can be achieved by setting parameter $\gamma = 1$ in Algorithm 1.

## 4. Foreign Word Polarity

As we mentioned earlier, a large body of research has focused on identifying the semantic orientation of words. This work has almost exclusively dealt with English and uses several language-dependent resources. When we try to apply these methods to other languages, we run into the problem of the lack of resources in other languages when compared with English. For example, the General Inquirer lexicon (Stone et al. 1966) has thousands of English words labeled with semantic orientation. Most of the literature has used it as a source of labeled seeds or for evaluation. Such lexicons are not readily available in other languages.

As we showed earlier, WordNet (Miller 1995) has been used for this task. However, even though W have been built for other languages, their coverage is relatively limited when compared to the English WordNet. The current release of English WordNet (WordNet 3.0) includes over 155K words and over 117K synsets. Looking at the resources for other languages, the Arabic WordNet (Black et al. 2006; Elkateb et al. 2006a, 2006b) contains only 11K synsets; the Hindi WordNet (Jha et al. 2001; Narayan et al. 2002) contains 32K synsets; Euro WordNet (Vossen 1997) contains 23K synsets in Spanish, 15K in German, and 22K in French, among other European languages. In some cases, accuracy was traded for coverage. For example, the current release of the Japanese WordNet has 57K synsets but contains errors in as many as 5% of the entries.[1]

In this section, we show how we can extend the methods presented earlier to predict the semantic orientation of foreign words. The proposed method is based on creating

1 http://nlpwww.nict.go.jp/wn-ja/index.en.html.

a multilingual network of words that represents both English and foreign words. The network has English–English connections, as well as Foreign–Foreign connections and English–Foreign connections. This allows us to benefit from the richness of the resources built for the English language and at the same time utilize resources specific to foreign languages. We define a random walk model over the multilingual network and predict the semantic orientation of any given word by comparing the mean hitting time of a random walk starting from it to a positive and a negative set of seed English words.

We use Arabic and Hindi in our experiments. We compare the performance of several methods using the foreign language resources only, and the multilingual network that has both English and foreign words. We show that bootstrapping from languages with dense resources such as English is useful for improving the performance on other languages with limited resources.

### 4.1 Multilingual Word Network

We build a network $G(V, E)$ where $V = V_{en} \cup V_{fr}$ is the union of the sets of English and Foreign words. $E$ is a set of edges connecting nodes in $V$. There are three types of connections: English–English connections, Foreign–Foreign connections, and English–Foreign connections. For the English–English connections, we use the same methodology as in Section 3.

Foreign–Foreign connections are created in a similar way to the English connections. Some foreign languages have lexical resources based on the design of the Princeton English WordNet. For example: Euro WordNet (Vossen 1997), Arabic WordNet (Black et al. 2006; Elkateb et al. 2006a, 2006b), and the Hindi WordNet (Jha et al. 2001; Narayan et al. 2002). We also use co-occurrence statistics similar to the work of Hatzivassiloglou and McKeown (1997).

Finally, to connect foreign words to English words, we use a Foreign to English dictionary. For every word in a list of foreign words, we look up its meaning in a dictionary and add an edge between the foreign word and every other English word that appeared as a possible meaning for it. If there is no comprehensive enough dictionary available, constructing a multilingual word network like a translation graph (Etzioni et al. 2007) may be a resolution.

### 4.2 Foreign Word Semantic Orientation Prediction

We use the multilingual network described previously to predict the semantic orientation of words based on the mean hitting time to two sets of positive and negative seeds. Given two lists of seed English words with known polarity, we define two sets of nodes $S^+$ and $S^-$ representing those seeds. For any given word $w$, we calculate the mean hitting time between $w$ and the two seed sets $h(w|S^+)$ and $h(w|S^-)$. If $h(w|S^+)$ is greater than $h(w|S^-)$, the word is classified as negative; otherwise it is classified as positive. We used the list of labeled seeds from Hatzivassiloglou and McKeown (1997) and Stone et al. (1966).

## 5. Out-of-Vocabulary Words

We observed that a significant portion of the text used on-line in discussions, comments, product reviews, and so on, contains words that are not defined in WordNet or in

standard dictionaries. We call these words Out-of-Vocabulary (OOV) words. Table 6 later in this article shows some OOV word examples. To show the importance of OOV word polarity identification, we calculated the proportion of OOV words in three corpora used for sentiment studies: a set of movie reviews, a set of on-line discussions from a political forum, and a set of randomly sampled tweets. For each word in the data, we look it up in two standard English dictionaries, together containing 160K unique words. Table 1 shows the statistics.

OOV words have a high chance of being polarized because people tend to use informal language or special acronyms to emphasize their attitudes or impress the audience. Therefore, being able to automatically identify the polarity of OOV words will essentially benefit real-world applications.

Consider the graph $G(W, E)$ described in Section 3.1. So far, the only resource we use to construct the graph is WordNet synsets. The first step in our approach to OOV word polarity identification is to find the words in WordNet that are related to an OOV word. Next, we add the OOV words to our graph by creating a new node for each OOV word and adding an edge between each OOV word and each of its related words. Once we have constructed the extended network, we use the random walk model described in Section 3.2 to predict the polarity of each OOV word.

### 5.1 Mining OOV Word Relatedness from the Web

There are several alternative methods of linking words in the graph. Agirre et al. (2009) studied the strengths and weaknesses of different approaches to term similarity and relatedness. They noticed that lexicographical methods such as the WordNet suffer from the limitation of lexicon coverage, which is the case here with OOV words. To overcome this limitation, we use a Web-based distributional approach to find the set of related words to each OOV word. We perform a Web search using the OOV word as a search query and retrieve the top $S$ search results. We extract the textual content of the retrieved results and tokenize it. After removing all the stop words, we compute the number of times each word co-occurs with the OOV word in the same document. We rank the words based on their co-occurrence frequency and return the top $R$ words as the set of related words to the given OOV word.

We experimented with three different variants of this approach. In the first variant, the frequency values of the co-occurring words are normalized by the lengths of the

**Table 1**
Proportion of OOV words in some corpora used for real world applications. (Numbers in parentheses exclude words whose first letters are capitalized because they are likely to refer to named entities.)

| corpus | source | # of words | Percentage of OOV |
|---|---|---|---|
| Movie reviews | 3,411 customer reviews from IMDb for the movie *The Dark Knight (2008)* | 10.7 M (9.5 M) | 5.3 (2.7) |
| Political forum | 23K sentences from `www.politicalforum.com` on various topics | 381 K (348 K) | 8 (6) |
| tweets | 0.6M random English tweets from `twitter.com`. (We count a tweet as in English if at least half of the words are English dictionary words. Tags and symbols were removed.) | 7.1 M (5.9 M) | 30 (27) |

documents that contributed to the count of each word. The intuition here is that longer documents contain more words and hence the probability that a word in the that document is related to the search query (i.e., the OOV word) is lower than when the document is shorter.

In the second variant, we only consider the words that appear in the proximity of the OOV word (i.e., within $d$ words around the OOV word) when we compute the co-occurrence frequency. The intuition here is that words that appear near the OOV word are more likely to be semantically related than the words that appear far away.

In the third variant, instead of searching the entire Web, we limit the search to social text. In the experiments described subsequently, we search for the OOV words in tweets posted on Twitter.[2] The intuition here is that searching the entire Web is likely to return results that do not necessarily contain opinionated text—particularly because many words have different senses. In contrast, the text written in a social context is more likely to carry sentiment and express emotions. This helps us find better related words that suit our task.

### 5.2 Word Network Extension with OOV Words

To extend the graph to include OOV words, we start with the graph $G(W, E)$ constructed from WordNet synsets. For each OOV word that does not exist in $G$, we create a new node $w$. We set the part of speech of $w$ to *unspecified*. Then we use the Web-based method described in the previous section to find a set of words that are most related to $w$. Finally, we create a link between each OOV word and each of its related words. To predict the polarity of an OOV word, we use the same random walk model described earlier.

## 6. Experiments

We performed experiments on the gold-standard data set for positive/negative words from the General Inquirer lexicon (Stone et al. 1966). The data set contains 4,206 words, 1,915 of which are positive and 2,291 of which are negative. Some of the ambiguous words were removed, as in Turney (2002) and Takamura, Inui, and Okumura (2005). Some examples of positive/negative words are listed in Table 2.

We use WordNet (Miller 1995) as a source of synonyms and hypernyms for the word relatedness graph. We used the Reuters Corpus, Volume 1 (Lewis et al. 2004) to generate co-occurrence statistics in the experiments that used them. We used 10-fold cross-validation for all tests. We evaluate our results in terms of accuracy. Statistical significance was tested using a two-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 described in Takamura, Inui, and Okumura (2005), the shortest path method described in Kamps et al. (2004), a re-implementation of the label propagation and Mincut methods described in Rao and Ravichandran (2009), and the bootstrapping method described in Hu and Liu (2004).

2 http://www.twitter.com.

**Table 2**
Examples of positive and negative words.

| Positive | | Negative | |
|---|---|---|---|
| able | adjective | abandon | verb |
| acceptable | adjective | abuse | verb |
| admire | verb | burglar | noun |
| amazing | adjective | chaos | noun |
| careful | adjective | contagious | adjective |
| ease | noun | corruption | noun |
| guide | verb | lie | verb |
| inspire | verb | reluctant | adjective |
| truthful | adjective | wrong | adjective |

### 6.1 Comparison 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, Inui, and Okumura (2005). Table 3 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, and using glosses did not help at all.

We also compare our method to a re-implementation of the label propagation (LP) method. Our method outperforms the LP method in both the 10-fold cross-validation set-up and when only 14 seeds are used.

We also compare our method to the SO-PMI method. Turney and Littman (2002) propose two methods for predicting the semantic orientation of words. They use Latent Semantic Analysis (SO-LSA) and Pointwise Mutual Information (SO-PMI) for measuring the statistical association between any given word and a set of 14 seed words. They describe this method as unsupervised because they only use 14 seeds as paradigm words that define the semantic orientation rather than train the model (Turney 2002).

**Table 3**
Accuracy for SO-PMI with different data set sizes, the spin model, the label propagation model, and the random walks model for 10-fold cross-validation and 14 seeds.

| – | CV | 14 seeds |
|---|---|---|
| SO-PMI ($1 \times 10^7$) | – | 61.3 |
| SO-PMI ($2 \times 10^9$) | – | 76.1 |
| SO-PMI ($1 \times 10^{11}$) | – | **82.8** |
| Spin Model | 91.5 | 81.9 |
| Label Propagation | 88.40 | 74.83 |
| Random Walks | **93.1** | 82.1 |

The SO-PMI value can be calculated as follows:

$$\text{SO-PMI}(w) = \log \frac{hits_{w,pos} \times hits_{neg}}{hits_{w,neg} \times hits_{pos}} \quad (5)$$

where $w$ is a word with unknown polarity, $hits_{w,pos}$ is the number of hits returned by a commercial search engine when the search query is the given word and the disjunction of all positive seed words. $hits_{pos}$ is the number of hits when we search for the disjunction of all positive seed words. $hits_{w,neg}$, and $hits_{neg}$ are defined similarly.

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 data set sizes, the spin model, and the proposed method using only 14 seeds is shown in Table 3. We observe that the random walk method outperforms SO-PMI when SO-PMI uses data sets of sizes $1 \times 10^7$ and $2 \times 10^9$ words. The performance of SO-PMI and the random walk methods are comparable when SO-PMI uses a very large data set ($1 \times 10^{11}$ words). The performance of the spin model approach is also comparable to the other two 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. 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 4,000 queries to a commercial search engine.

We also compare our method with 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 because our method is mainly interested in classifying adjectives.

The performance of the spin model, the bootstrapping method, the shortest path method, the LP method, the Mincut method, and the random walk method for only adjectives is shown in Table 4. We notice from the table that the random walk method outperforms the spin model, the bootstrapping method, the shortest path method, the LP method, and the Mincut 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, its accuracy will drop to around 61%.

*6.1.1 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 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 $m$ as it varies from 5 to 50. We use a network built from WordNet synonyms and hypernyms only. The number of samples $k$ was set to $1,000$.

**Table 4**
Accuracy for adjectives only for the spin model, the bootstrap method, and the random walk model.

| Method | Spin Model | Bootstrap | Shortest Path | LP | Mincut | Random Walks |
|---|---|---|---|---|---|---|
| Accuracy | 83.6 | 72.8 | 68.8 | 84.8 | 73.8 | 88.8 |

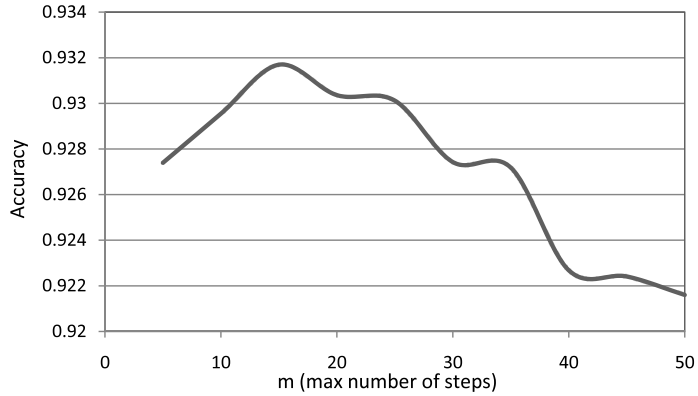


**Figure 1**
The effect of varying the maximum number of steps ($m$) on accuracy ($k$ = 1,000).

We perform 10-fold cross-validation using the General Inquirer lexicon. We observe that the maximum number of steps $m$ has very little impact on performance until it rises above 30. At that point, the performance drops by no more than 1%, and then it no longer changes 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 data set 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 with the diameter of the graph, the random walk that starts at ambiguous words (which are hard to classify) have the chance of moving until 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 are hence ignored.

Next, we study the effect of the number 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

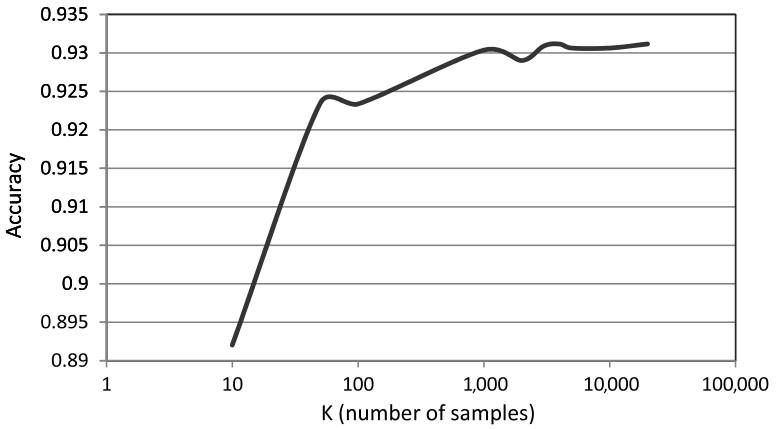


**Figure 2**
The effect of varying the number of samples ($k$) on accuracy.

settings as in the previous experiment. The only difference is that we fix $m$ at 15 and vary $k$ from 10 to 20,000 (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 1,000, 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 1,000.

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

*6.1.2 Other Experiments.* We now measure the performance of the random walk method when the system is allowed to abstain from classifying the words for which it has 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 3 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 accuracy greater than 99% for 10-fold cross validation and 92% with 14 seed words. This may be compared with the work described in Takamura, Inui, and Okumura (2005), where they achieve the 92% level when they only consider the top 1,000 words (28%).

Figure 4 shows a learning curve displaying how the performance of both the proposed method and the LP 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 size of 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 such as *love*, *harm*, *friend*, *enemy*, the performance for nouns and verbs improves considerably at the cost of a small drop in the

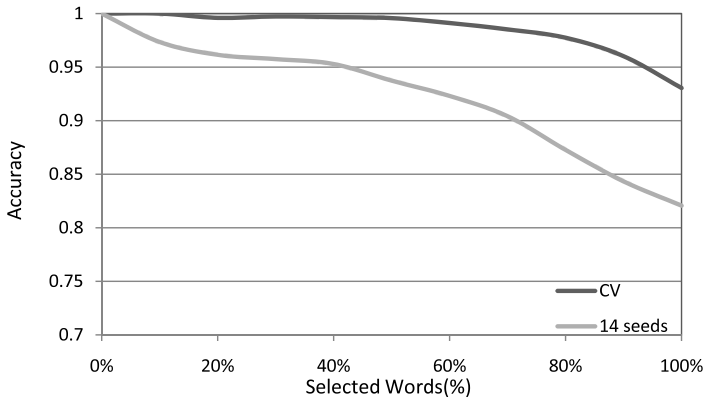


**Figure 3**
Accuracy for words with high confidence measure.

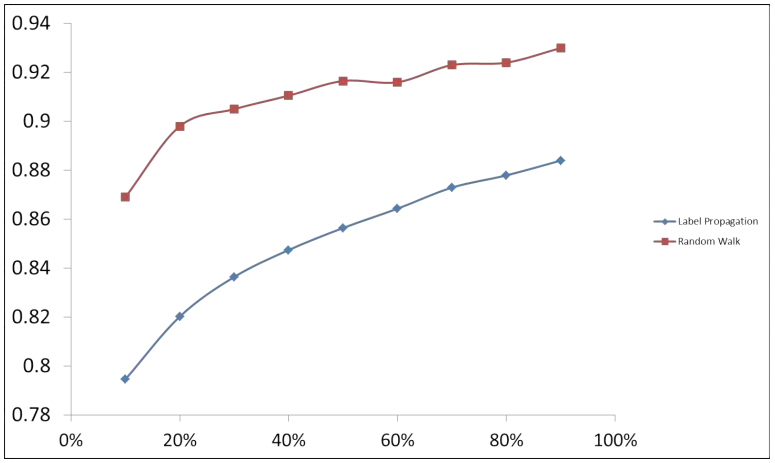


**Figure 4**
The effect of varying the number of seeds on accuracy.

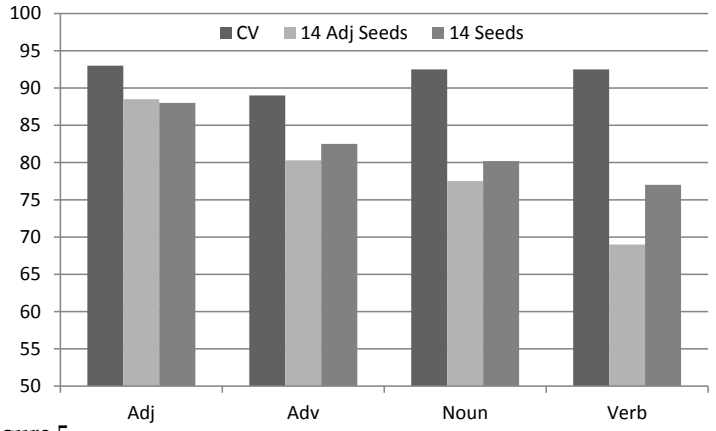


**Figure 5**
Accuracy for different parts of speech.

performance on adjectives. Finally, we tried adding edges to the network from glosses and co-occurrence statistics but we did not get any statistically significant improvement. Some of the words that were very weakly linked benefited from adding new types of links and they were correctly predicted. Others were misled by the noise and were incorrectly classified. 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 have more than one sense, possibly with different semantic 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 connections in the thesaurus. A possible solution to this might be to identify those words and add more links to them from glosses of co-occurrence statistics in the corpus.

*6.1.3 General Purpose Three-Way Classification.* The experiments described so far all use the General Inquirer lexicon, which contains a well-established gold standard data set of positive and negative words. However, in realistic applications, a general purpose

**Table 5**
Accuracy for three classes on a general purpose list of 2,000 words.

| Class | Positive | Negative | Neutral | Overall |
|---|---|---|---|---|
| Accuracy | 68.0 | 82.1 | 80.6 | 77.9 |

list of words will frequently have neutral words that don't express sentiment polarity. To evaluate the effectiveness of the random walk method in distinguishing polarized words from neutral words, we constructed a data set of 2,000 words randomly picked from a standard English dictionary[3] and hand labeled them with three classes: positive, negative, and neutral. Among the 2,000 words, 494 were labeled positive, 491 negative, and 1,015 neutral. The distribution among different parts of speech is 532 adjectives, 335 verbs, 1,051 nouns, and 82 others.

We used the semi-supervised setting with the General Inquirer lexicon polarized word list as the training set. Because the 2,000 test set has some portion of polarized words overlapping with the training set, we excluded the words that appear in the test set from the training set. We performed Algorithm 2 in Section 3.4 with parameters $\gamma = 0.8, m = 15, k = 1,000$. The overall accuracy as well as the precision for each class is shown in Table 5. We can see that the accuracy of the positive class is much lower than the negative class, due to the many positive words classified as neutral. This means that the average confidence of negative words is higher than positive words. One factor that could have caused this is the bias originating from the training set. Because there are more negative seeds than positive ones, the constructed graph has an overall bias towards the negative class.

### 6.2 Foreign Words

In addition to the English data we described earlier, we constructed a labeled set of 300 Arabic and 300 Hindi words for evaluation. For every language, we asked two native speakers to examine a large amount of text and identify a set of positive and negative words. We also used an Arabic–English and a Hindi–English dictionary to generate Foreign–English links.

We compare our results with two baselines. The first is the SO-PMI method described in Turney and Littman (2003). We used the same seven positive and seven negative seeds as Turney and Littman (2003).

The second baseline constructs a network of only foreign words as described earlier. It uses mean hitting time to find the semantic association of any given word. We used 10-fold cross-validation for this experiment. We will refer to this system as HT-FR.

Finally, we build a multilingual network and use the hitting time as before to predict semantic orientation. We used the English words from Stone et al. (1966) as seeds and the labeled foreign words for evaluation. We will refer to this system as HT-FR-EN.

Figure 6 compares the accuracy of the three methods for Arabic and Hindi. We notice that the SO-PMI and the hitting time–based methods perform poorly on both Arabic and Hindi. This is clearly evident when we consider that the accuracy of the two systems on English was 83%, and 93%, respectively (Turney and Littman 2003; Hassan

3 Very infrequent words were filtered out by setting a threshold on the inverse document frequency of the words in a corpus.

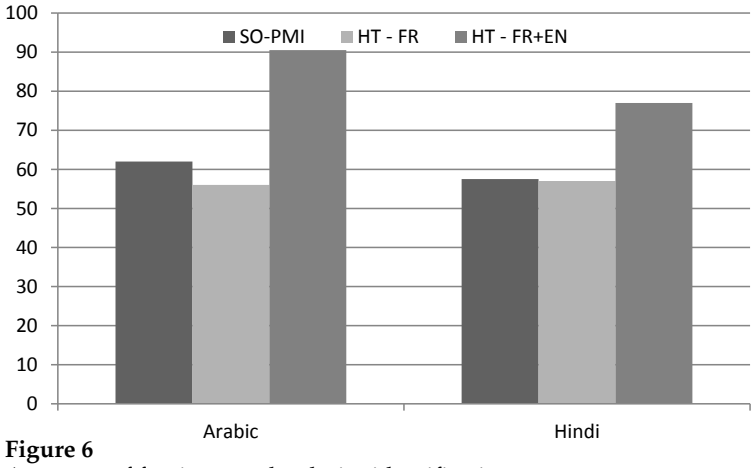


**Figure 6**
Accuracy of foreign word polarity identification.

and Radev 2010). This supports our hypothesis that state-of-the-art methods, designed for English, perform poorly on foreign languages due to the limited amount of resources in them. The figure also shows that the proposed method, which combines resources from both English and foreign languages, performs significantly better. Finally, we studied how much improvement is achieved by including links between foreign words from global WordNets. We found out that it improves the performance by 2.5% and 4% for Arabic and Hindi, respectively.

### 6.3 OOV Words

We created a labeled set of 300 positive and negative OOV words. We asked a native English speaker to examine a large number of threads posted on several on-line forums and identify OOV words and label them with their polarities. Some examples of positive/negative OOV words are listed in Table 6.

The baseline we use for OOV words is the SO-PMI method with the same 14 seeds as in Turney and Littman (2003). The calculation of SO-PMI is given in Equation (5).

We used the approach described in Section 5 to automatically label the words. We used the words of the General Inquirer lexicon as labeled seeds. We set the maximum number of steps $m$ to 15 and the number of samples $k$ to 1,000. We experimented with

**Table 6**
Examples of positive and negative OOV words.

| Positive | | Negative | |
|---|---|---|---|
| Word | Meaning | Word | Meaning |
| beautimous | beautiful and fabulous | disastrophy | a catastrophy and a disaster |
| gr8 | great | banjaxed | ruined |
| buffting | attractive | ijit | idiot |

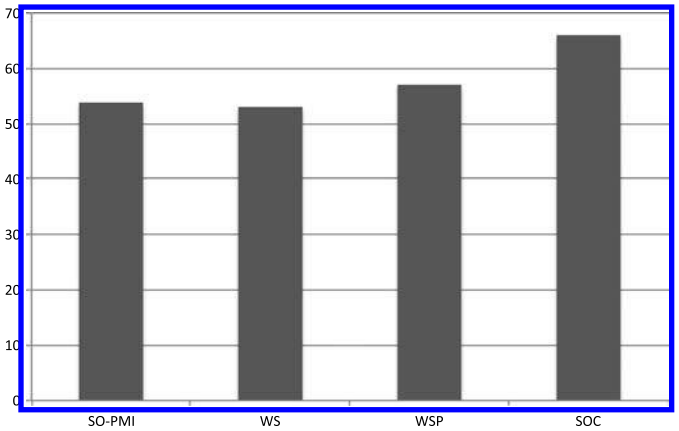


**Figure 7**
Accuracy of different methods in predicting OOV words polarity.

the three variants we proposed for extracting the related words as described in Section 5. We give the experimental set-up for each variant here:

1. Search the entire Web (WS): We used Yahoo search[4] to execute the search queries. For each OOV word, we retrieve the top 500 results and use them to extract the related words.
2. Search the entire Web and limit the extraction of related words to the proximity of the OOV word (WSP): We fix the proximity of a given OOV word to 15 words before and 15 words after the OOV word (we experimented with different ranges but no significant changes were observed).
3. Limit the search to social content (SOC): We limit the search for OOV words to tweets posted on Twitter. We use the Twitter search API to submit the search queries. For each OOV word, we retrieve 10,000 tweets. Each tweet is maximum of 140 characters long.

Figure 7 shows the results of the three methods compared with the baseline SO-PMI. The results show that extracting related words from tweets gives the best accuracy. This corroborated our intuition that using social content is more likely to provide sentiment-related words. The baseline SO-PMI and WS obtain very similar accuracy. This agrees with the comparable performance of the two methods in the earlier experiment on the General Inquirer lexicon.

The three variant methods for obtaining related words have a tunable parameter $R$, the number of related words extracted for each OOV word. We observe that $R$ has a non-negligible effect on the prediction accuracy. The results shown in Figure 8 correspond to $R = 90$. To better understand the impact of varying this parameter, we ran the experiment that uses Twitter to extract related words several times using different values for $R$. Figure 8 shows how the accuracy of polarity prediction changes as $R$ changes.

4 http://www.yahoo.com.

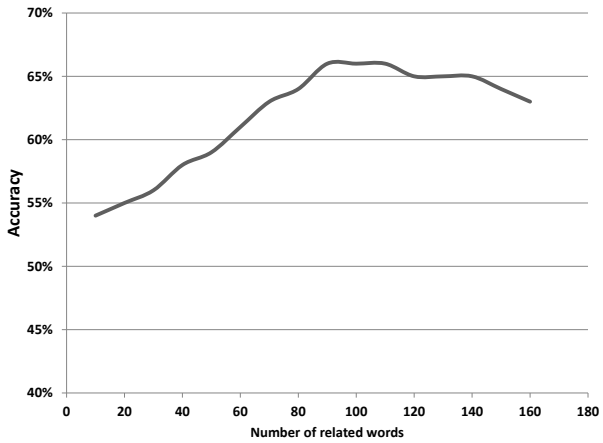


**Figure 8**
The effect of varying the number of extracted related words on accuracy.

## 7. 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 a weakly supervised 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. We also extended the proposed method to cover the problem of predicting the semantic orientation of foreign words. All previous work on this task has almost exclusively focused on English. Applying off-the-shelf methods developed for English to other languages does not work well because of the limited amount of resources available in foreign languages compared with English. We show that the proposed method can predict the semantic orientation of foreign words with high accuracy and outperforms state-of-the-art methods limited to using language specific resources. Finally, we further extended the method to cover out-of-vocabulary words. These words do not exist in WordNet and are not defined in the standard dictionaries of the language. We proposed using a Web-based approach to add the OOV words to our words network based on co-occurrence statistics, then use the same random walk model to predict the polarity. We showed that this method can predict the polarity of OOV words with good accuracy.

**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 contained 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


Agirre, Eneko, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 19–27, Stroudsburg, PA.

Andreevskaia, Alina and Sabine Bergler. 2006. Mining WordNet for fuzzy sentiment: Sentiment tag extraction from WordNet glosses. In *EACL'06*, pages 209–216.

Banea, Carmen, Rada Mihalcea, and Janyce Wiebe. 2008. A bootstrapping method for building subjectivity lexicons for languages with scarce resources. In *LREC'08*, pages 2,764–2,767.

Black, W., S. Elkateb, H. Rodriguez, M. Alkhalifa, P. Vossen, A. Pease, and C. Fellbaum. 2006. Introducing the Arabic WordNet project. In *Third International WordNet Conference*, pages 295–299.

Blair-Goldensohn, Sasha, Tyler Neylon, Kerry Hannan, George A. Reis, Ryan McDonald, and Jeff Reynar. 2008. Building a sentiment summarizer for local service reviews. In *NLP in the Information Explosion Era*.

Brody, Samuel and Noemie Elhadad. 2010. An unsupervised aspect-sentiment model for online reviews. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 804–812, Los Angeles, CA.

Elkateb, S., W. Black, H. Rodriguez, M. Alkhalifa, P. Vossen, A. Pease, and C. Fellbaum. 2006a. Building a WordNet for Arabic. In *Fifth International Conference on Language Resources and Evaluation*, pages 29–34.

Elkateb, S., W. Black, P. Vossen, D. Farwell, H. Rodriguez, A. Pease, and M. Alkhalifa. 2006b. Arabic WordNet and the challenges of Arabic. In *Arabic NLP/MT Conference*, pages 15–24.

Esuli, Andrea and Fabrizio Sebastiani. 2005. Determining the semantic orientation of terms through gloss classification. In *CIKM'05*, pages 617–624.

Esuli, Andrea and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *LREC'06*, pages 417–422.

Etzioni, Oren, Kobi Reiter, Stephen Soderl, and Marcus Sammer. 2007. Lexical translation with application to image search on the Web. In *Proceedings of Machine Translation Summit XI*.

Hassan, Ahmed and Dragomir R. Radev. 2010. Identifying text polarity using random walks. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 395–403, Uppsala.

Hatzivassiloglou, Vasileios and Kathleen R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *EACL'97*, pages 174–181.

Hatzivassiloglou, Vasileios and Janyce Wiebe. 2000. Effects of adjective orientation and gradability on sentence subjectivity. In *COLING*, pages 299–305.

Hu, Minqing and Bing Liu. 2004. Mining and summarizing customer reviews. In *KDD'04*, pages 168–177.

Jha, S., D. Narayan, P. Pande, and P. Bhattacharyya. 2001. A WordNet for Hindi. In *International Workshop on Lexical Resources in Natural Language Processing*.

Jijkoun, Valentin and Katja Hofmann. 2009. Generating a non-English subjectivity lexicon: Relations that matter. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 398–405, Athens.

Kamps, Jaap, Maarten Marx, Robert J. Mokken, and Maarten De Rijke. 2004. Using WordNet to measure semantic orientations of adjectives. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*, pages 1115–1118.

Kanayama, Hiroshi and Tetsuya Nasukawa. 2006. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *EMNLP'06*, pages 355–363.

Kim, Soo-Min and Eduard Hovy. 2004. Determining the sentiment of opinions. In *COLING*, pages 1,367–1,373.

Kok, Stanley and Chris Brockett. 2010. Hitting the right paraphrases in good time. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 145–153, Los Angeles, CA.

Lewis, D. D., Y. Yang, T. Rose, and F. Li. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397.

Mihalcea, Rada and Carmen Banea. 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.
Miller, George A. 1995. Wordnet: A lexical database for English. *Communications of ACM*, 38(11):39–41.
Mohammad, Saif, Cody Dunne, and Bonnie Dorr. 2009. Generating high-coverage semantic orientation lexicons from overtly marked words and a thesaurus. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2*, EMNLP '09, pages 599–608, Stroudsburg, PA.
Morinaga, Satoshi, Kenji Yamanishi, Kenji Tateishi, and Toshikazu Fukushima. 2002. Mining product reputations on the Web. In *KDD'02*, pages 341–349.
Narayan, Dipak, Debasri Chakrabarti, Prabhakar Pande, and P. Bhattacharyya. 2002. An experience in building the Indo WordNet—a WordNet for Hindi. In *First International Conference on Global WordNet*.
Nasukawa, Tetsuya 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.
Norris, J. 1997. *Markov Chains*. Cambridge University Press.
Pang, Bo and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, ACL '04, Stroudsburg, PA.
Popescu, Ana-Maria and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *HLT-EMNLP'05*, pages 339–346.
Rao, Delip and Deepak Ravichandran. 2009. Semi-supervised polarity lexicon induction. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 675–682, Athens.
Riloff, Ellen and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. In *EMNLP'03*, pages 105–112.
Stone, Philip, Dexter Dunphy, Marchall Smith, and Daniel Ogilvie. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. The MIT Press.
Su, Fangzhong and Katja Markert. 2009. Subjectivity recognition on word senses via semi-supervised mincuts. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 1–9, Stroudsburg, PA.
Szummer, Martin and Tommi Jaakkola. 2002. Partially labeled classification with Markov random walks. In *NIPS'02*, pages 945–952.
Takamura, Hiroya, Takashi Inui, and Manabu Okumura. 2005. Extracting semantic orientations of words using spin model. In *ACL'05*, pages 133–140.
Tong, Richard M. 2001. An operational system for detecting and tracking opinions in on-line discussion. Workshop note, *SIGIR 2001 Workshop on Operational Text Classification*.
Turney, Peter and Michael Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21:315–346.
Turney, Peter D. 2002. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *ACL'02*, pages 417–424.
Velikovich, Leonid, Sasha Blair-Goldensohn, Kerry Hannan, and Ryan McDonald. 2010. The viability of Web-derived polarity lexicons. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 777–785, Los Angeles, CA.
Vossen, P. 1997. Eurowordnet: A multilingual database for information retrieval. In *DELOS Workshop on Cross-Language Information Retrieval*, pages 5–7.
Wiebe, Janyce. 2000. Learning subjective adjectives from corpora. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and the Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 735–740.
Wiebe, Janyce, 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.
Wiebe, Janyce and Rada Mihalcea. 2006a. Word sense and subjectivity. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 1,065–1,072, Sydney.
Wiebe, Janyce and Rada Mihalcea. 2006b. Word sense and subjectivity. In *Proceedings*

*of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 1,065–1,072, Stroudsburg, PA.

Wiebe, Janyce, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2-3):165–210.

Yu, Hong and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *EMNLP'03*, pages 129–136.

Zhu, Xiaojin, Zoubin Ghahramani, and John Lafferty. 2003. Semi-supervised learning using Gaussian fields and harmonic functions. In *ICML'03*, pages 912–919.