

Identifying Sentiment Words Using an Optimization-based Model without Seed Words

Hongliang Yu¹, Zhi-Hong Deng^{2*}, Shiyinxue Li³

Key Laboratory of Machine Perception (Ministry of Education),
School of Electronics Engineering and Computer Science,
Peking University, Beijing 100871, China

¹yuhongliang324@gmail.com

²zhdeng@cis.pku.edu.cn

³rachellieinspace@gmail.com

Abstract

Sentiment Word Identification (SWI) is a basic technique in many sentiment analysis applications. Most existing researches exploit seed words, and lead to low robustness. In this paper, we propose a novel optimization-based model for SWI. Unlike previous approaches, our model exploits the sentiment labels of documents instead of seed words. Several experiments on real datasets show that WEED is effective and outperforms the state-of-the-art methods with seed words.

1 Introduction

In recent years, sentiment analysis (Pang et al., 2002) has become a hotspot in opinion mining and attracted much attention. Sentiment analysis is to classify a text span into different sentiment polarities, i.e. positive, negative or neutral. Sentiment Word Identification (SWI) is a basic technique in sentiment analysis. According to (Ku et al., 2006)(Chen et al., 2012)(Fan et al., 2011), SWI can be applied to many fields, such as determining critics opinions about a given product, tweeter classification, summarization of reviews, and message filtering, etc. Thus in this paper, we focus on SWI.

Here is a simple example of how SWI is applied to comment analysis. The sentence below is an movie review in IMDB database:

- **Bored** performers and a **lackluster** plot and script, do not make a **good** action movie.

In order to judge the sentence polarity (thus we can learn about the preference of this user), one must recognize which words are able to express sentiment. In this sentence, “bored” and “lackluster” are negative while “good” should be positive, yet

its polarity is reversed by “not”. By such analysis, we then conclude such movie review is a negative comment. But how do we recognize sentiment words?

To achieve this, previous supervised approaches need labeled polarity words, also called **seed words**, usually manually selected. The words to be classified by their sentiment polarities are called **candidate words**. Prior works study the relations between labeled seed words and unlabeled candidate words, and then obtain sentiment polarities of candidate words by these relations. There are many ways to generate word relations. The authors of (Turney and Littman, 2003) and (Kaji and Kitsuregawa, 2007) use statistical measures, such as point wise mutual information (PMI), to compute similarities in words or phrases. Kanayama and Nasukawa (2006) assume sentiment words successively appear in the text, so one could find sentiment words in the context of seed words (Kanayama and Nasukawa, 2006). In (Hassan and Radev, 2010) and (Hassan et al., 2011), a Markov random walk model is applied to a large word relatedness graph, constructed according to the synonyms and hypernyms in WordNet (Miller, 1995).

However, approaches based on seed words has obvious shortcomings. First, polarities of seed words are not reliable for various domains. As a simple example, “rise” is a neutral word most often, but becomes positive in stock market. Second, manually selection of seed words can be very subjective even if the application domain is determined. Third, algorithms using seed words have low robustness. Any missing key word in the set of seed words could lead to poor performance. Therefore, the seed word set of such algorithms demands high completeness (by containing common polarity words as many as possible).

Unlike the previous research work, we identify sentiment words without any seed words in this paper. Instead, the documents’ bag-of-words in-

*Corresponding author

formation and their polarity labels are exploited in the identification process. Intuitively, polarities of the document and its most component sentiment words are the same. We call such phenomenon as “sentiment matching”. Moreover, if a word is found mostly in positive documents, it is very likely a positive word, and vice versa.

We present an optimization-based model, called WEED, to exploit the phenomenon of “sentiment matching”. We first measure the importance of the component words in the labeled documents semantically. Here, the basic assumption is that important words are more sentiment related to the document than those less important. Then, we estimate the polarity of each document using its component words’ importance along with their sentiment values, and compare the estimation to the real polarity. After that, we construct an optimization model for the whole corpus to weigh the overall estimation error, which is minimized by the best sentiment values of candidate words. Finally, several experiments demonstrate the effectiveness of our approach. To the best of our knowledge, this paper is the first work that identifies sentiment words without seed words.

2 The Proposed Approach

2.1 Preliminary

We formulate the sentiment word identification problem as follows. Let $D = \{d_1, \dots, d_n\}$ denote

document set. Vector $\vec{l} = \begin{bmatrix} l_1 \\ \vdots \\ l_n \end{bmatrix}$ represents their

labels. If document d_i is a positive sample, then $l_i = 1$; if d_i is negative, then $l_i = -1$. We use the notation $C = \{c_1, \dots, c_V\}$ to represent candidate word set, and V is the number of candidate words. Each document is formed by consecutive words in C . Our task is to predict the sentiment polarity of each word $c_j \in C$.

2.2 Word Importance

We assume each document $d_i \in D$ is presented

by a bag-of-words feature vector $\vec{f}_i = \begin{bmatrix} f_{i1} \\ \vdots \\ f_{iV} \end{bmatrix}$,

where f_{ij} describes the importance of c_j to d_i . A high value of f_{ij} indicates word c_j contributes a lot to document d_i in semantic view, and vice versa. Note that $f_{ij} > 0$ if c_j appears in d_i , while

$f_{ij} = 0$ if not. For simplicity, every \vec{f}_i is normalized to a unit vector, such that features of different documents are relatively comparable.

There are several ways to define the word importance, and we choose normalized *TF-IDF* (Jones, 1972). Therefore, we have $f_{ij} \propto TF-IDF(d_i, c_j)$, and $\|\vec{f}_i\| = 1$.

2.3 Polarity Value

In the above description, the sentiment polarity has only two states, positive or negative. We extend both word and document polarities to polarity values in this section.

Definition 1 *Word Polarity Value:* For each word $c_j \in C$, we denote its **word polarity value** as $w(c_j)$. $w(c_j) > 0$ indicates c_j is a positive word, while $w(c_j) < 0$ indicates c_j is a negative word. $|w(c_j)|$ indicates the strength of the belief of c_j ’s polarity. Denote $w(c_j)$ as w_j , and the word polar-

ity value vector $\vec{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_V \end{bmatrix}$.

For example, if $w(\text{“bad”}) < w(\text{“greedy”}) < 0$, we can say “bad” is more likely to be a negative word than “greedy”.

Definition 2 *Document Polarity Value:* For each document d_i , **document polarity value** is

$$y(d_i) = \text{cosine}(\vec{f}_i, \vec{w}) = \frac{\vec{f}_i^T \cdot \vec{w}}{\|\vec{w}\|}. \quad (1)$$

We denote $y(d_i)$ as y_i for short.

Here, we can regard y_i as a polarity estimate for d_i based on \vec{w} . To explain this, Table 1 shows an example. “MR1”, “MR2” and “MR3” are three movie review documents, and “compelling” and “boring” are polarity words in the vocabulary. we simply use TF to construct the document feature vectors without normalization. In the table, these three vectors, \vec{f}_1 , \vec{f}_2 and \vec{f}_3 , are (3, 1), (2, 1) and (1, 3) respectively. Similarly, we can get $\vec{w} = (1, -1)$, indicating “compelling” is a positive word while “boring” is negative. After normalizing \vec{f}_1 , \vec{f}_2 and \vec{f}_3 , and calculating their cosine similarities with \vec{w} , we obtain $y_1 > y_2 > 0 > y_3$. These inequalities tell us the first two reviews are positive, while the last review is negative. Furthermore, we believe that “MR1” is more positive than “MR2”.

	“compelling”	“boring”
MR1	3	1
MR2	2	1
MR3	1	3
w	1	-1

Table 1: Three rows in the middle shows the feature vectors of three movie reviews, and the last row shows the word polarity value vector \vec{w} . For simplicity, we use TF value to represent the word importance feature.

2.4 Optimization Model

As mentioned above, we can regard y_i as a polarity estimate for document d_i . A precise prediction makes the positive document’s estimator close to 1, and the negative’s close to -1. We define the polarity estimate error for document d_i as:

$$e_i = |y_i - l_i| = \left| \frac{\vec{f}_i^T \cdot \vec{w}}{\|\vec{w}\|} - l_i \right|. \quad (2)$$

Our learning procedure tries to decrease e_i . We obtain \vec{w} by minimizing the overall estimation error of all document samples $\sum_{i=1}^n e_i^2$. Thus, the optimization problem can be described as

$$\min_{\vec{w}} \sum_{i=1}^n \left(\frac{\vec{f}_i^T \cdot \vec{w}}{\|\vec{w}\|} - l_i \right)^2. \quad (3)$$

After solving this problem, we not only obtain the polarity of each word c_j according to the sign of w_j , but also its polarity belief based on $|w_j|$.

2.5 Model Solution

We use normalized vector \vec{x} to substitute $\frac{\vec{w}}{\|\vec{w}\|}$, and derive an equivalent optimization problem:

$$\begin{aligned} \min_{\vec{x}} \quad & E(\vec{x}) = \sum_{i=1}^n (\vec{f}_i^T \cdot \vec{x} - l_i)^2 \\ \text{s.t.} \quad & \|\vec{x}\| = 1. \end{aligned} \quad (4)$$

The equality constraint of above model makes the problem non-convex. We relax the equality constraint to $\|\vec{x}\| \leq 1$, then the problem becomes convex. We can rewrite the objective function as the form of least square regression: $E(\vec{x}) = \|\mathbf{F} \cdot \vec{x} - \vec{l}\|^2$, where \mathbf{F} is the feature matrix, and

$$\text{equals to } \begin{bmatrix} \vec{f}_1^T \\ \vdots \\ \vec{f}_n^T \end{bmatrix}.$$

Now we can solve the problem by convex optimization algorithms (Boyd and Vandenberghe, 2004), such as gradient descend method. In each iteration step, we update \vec{x} by $\Delta \vec{x} = \eta \cdot (-\nabla E) = 2\eta \cdot (\mathbf{F}^T \vec{l} - \mathbf{F}^T \mathbf{F} \vec{x})$, where $\eta > 0$ is the learning rate.

3 Experiment

3.1 Experimental Setup

We leverage two widely used document datasets. The first dataset is the Cornell Movie Review Data¹, containing 1,000 positive and 1,000 negative processed reviews. The other is the Stanford Large Dataset² (Maas et al., 2011), a collection of 50,000 comments from IMDB, evenly divided into training and test sets.

The ground-truth is generated with the help of a sentiment lexicon, MPQA subjective lexicon³. We randomly select 20% polarity words as the seed words, and the remaining are candidate ones. Here, the seed words are provided for the baseline methods but not for ours. In order to increase the difficulty of our task, several non-polarity words are added to the candidate word set. Table 2 shows the word distribution of two datasets.

Dataset	Word Set	pos	neg	non	total
Cornell	seed	135	201	-	336
	candidate	541	806	1232	2579
Stanford	seed	202	343	-	545
	candidate	808	1370	2566	4744

Table 2: Word Distribution

In order to demonstrate the effectiveness of our model, we select two baselines, *SO-PMI* (Turney and Littman, 2003) and *COM* (Chen et al., 2012). Both of them need seed words.

3.2 Top-K Test

In face of the long lists of recommended polarity words, people are only concerned about the top-ranked words with the highest sentiment value. In this experiment we consider the accuracy of the top K polarity words. The quality of a polarity word list is measured by $p@K = \frac{N_{right,K}}{K}$, where $N_{right,K}$ is the number of top- K words which are correctly recommended.

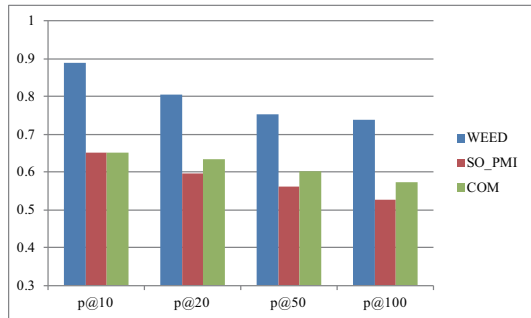
¹<http://www.cs.cornell.edu/people/pabo/movie-review-data/>

²<http://ai.stanford.edu/amaas/data/sentiment/>

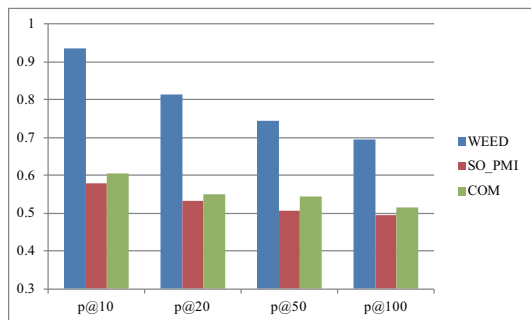
³<http://www.cs.pitt.edu/mpqa/>

WEED		SO-PMI		COM	
positive words	negative words	positive words	negative words	positive words	negative words
great excellent	bad stupid	destiny lush	cheap worst	best great	ridiculous bad
perfect perfectly	worst mess	brilliant skillfully	ridiculous annoying	will star	plot evil
terrific best	boring ridiculous	courtesy courtesy	damn pathetic	bad fun	star garish
true wonderfully	awful plot	gorgeous magnificent	inconsistencies fool	better plot	dreadfully stupid
brilliant outstanding	worse terrible	temptation marvelously	desperate giddy	love horror	pretty fun

Table 3: Case Study



(a) Cornell Dataset



(b) Stanford Dataset

Figure 1: Top-K Test

Figure 1 shows the final result of $p@K$, which is the average score of the positive and negative list. We can see that in both datasets, our approach highly outperforms two baselines, and the precision is 14.4%-33.0% higher than the best baseline. $p@10$ s of WEED for Cornell and Stanford datasets reach to 93.5% and 89.0%, and it shows the top 10 words in our recommended list is exceptionally reliable. As the size of K increases, the accuracy of all methods falls accordingly. This shows three approaches rank the most probable polarity words in the front of the word list. Compared with the small dataset, we obtain a better result with large K on the Stanford dataset.

3.3 Case Study

We conduct an experiment to illustrate the characteristics of three methods. Table 3 shows top-10 positive and negative words for each method,

where the bold words are the ones with correct polarities. From the first two columns, we can see the accuracy of WEED is very high, where positive words are absolutely correct and negative word list makes only one mistake, “plot”. The other columns of this table shows the baseline methods both achieve reasonable results but do not perform as well as WEED.

Our approach is able to identify frequently used sentiment words, which are vital for the applications without prior sentiment lexicons. The sentiment words identified by SO-PMI are not so representative as WEED and COM. For example, “skillfully” and “giddy” are correctly classified but they are not very frequently used. COM tends to assign wrong polarities to the sentiment words although these words are often used. In the 5th and 6th columns of Table 3, “bad” and “horror” are recognized as positive words, while “pretty” and “fun” are recognized as negative ones. These concrete results show that WEED captures the generality of the sentiment words, and achieves a higher accuracy than the baselines.

4 Conclusion and Future Work

We propose an effective optimization-based model, WEED, to identify sentiment words from the corpus without seed words. The algorithm exploits the sentiment information provided by the documents. To the best of our knowledge, this paper is the first work that identifies sentiment words without any seed words. Several experiments on real datasets show that WEED outperforms the state-of-the-art methods with seed words.

Our work can be considered as the first step of building a domain-specific sentiment lexicon. Once some sentiment words are obtained in a certain domain, our future work is to improve WEED by utilizing these words.

Acknowledgments

This work is partially supported by National Natural Science Foundation of China (Grant No. 61170091).

References

- S. Boyd and L. Vandenberghe. 2004. *Convex optimization*. Cambridge university press.
- L. Chen, W. Wang, M. Nagarajan, S. Wang, and A.P. Sheth. 2012. Extracting diverse sentiment expressions with target-dependent polarity from twitter. In *Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media (ICWSM)*, pages 50–57.
- Wen Fan, Shutao Sun, and Guohui Song. 2011. Probability adjustment naïve bayes algorithm based on nondomain-specific sentiment and evaluation word for domain-transfer sentiment analysis. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2011 Eighth International Conference on*, volume 2, pages 1043–1046. IEEE.
- A. Hassan and D. Radev. 2010. Identifying text polarity using random walks. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 395–403. Association for Computational Linguistics.
- A. Hassan, A. Abu-Jbara, R. Jha, and D. Radev. 2011. Identifying the semantic orientation of foreign words. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 592–597.
- K.S. Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.
- N. Kaji and M. Kitsuregawa. 2007. Building lexicon for sentiment analysis from massive collection of html documents. In *Proceedings of the joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 1075–1083.
- H. Kanayama and T. Nasukawa. 2006. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 355–363. Association for Computational Linguistics.
- Lun-Wei Ku, Yu-Ting Liang, and Hsin-Hsi Chen. 2006. Opinion extraction, summarization and tracking in news and blog corpora. In *Proceedings of AAAI-2006 spring symposium on computational approaches to analyzing weblogs*, volume 2001.
- A.L. Maas, R.E. Daly, P.T. Pham, D. Huang, A.Y. Ng, and C. Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational Linguistics (acL-2011)*.
- Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.
- B. Pang, L. Lee, and S. Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.
- P. Turney and M.L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association.