Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Identifying Sentiment Words Using an Optimization Model with L_1 Regularization

Zhi-Hong Deng † **and Hongliang Yu** † ‡ **and Yunlun Yang** †
†Key Laboratory of Machine Perception (Ministry of Education),
School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China
‡Language Technologies Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA
zhdeng@cis.pku.edu.cn, yuhongliang@cs.cmu.edu, incomparable-lun@pku.edu.cn

Abstract

Sentiment word identification is a fundamental work in numerous applications of sentiment analysis and opinion mining, such as review mining, opinion holder finding, and twitter classification. In this paper, we propose an optimization model with L_1 regularization, called ISOMER, for identifying the sentiment words from the corpus. Our model can employ both seed words and documents with sentiment labels, different from most existing researches adopting seed words only. The L_1 penalty in the objective function yields a sparse solution since most candidate words have no sentiment. The experiments on the real datasets show that ISOMER outperforms the classic approaches, and that the lexicon learned by ISOMER can be effectively adapted to document-level sentiment analysis.

Introduction

With the rapid growth of Web 2.0, loads of user-generated messages expressing sentiment spread throughout the internet. Some messages imply a user's predilection, such as his preference for a certain product, or mood after watching a film. Discovering such hidden information is a demanding task. That is why sentiment analysis (Pang and Lee 2008; Liu 2012) has become a hotspot in recent years.

Sentiment word identification is a fundamental work in numerous applications of sentiment analysis and opinion mining. (Dave, Lawrence, and Pennock 2003) develop a method for automatically distinguishing positive and negative product reviews. In (Kim and Hovy 2004), the algorithm can find the opinion holder and sentiment given a topic by determining the word sentiment first. In the information extracting system for review mining, (Popescu and Etzioni 2007) present a component to evaluate the sentiment polarity of words in the context of given product features and sentences. Also, according to (Chen et al. 2012), sentiment word identification can be applied to twitter classification.

But how do we recognize sentiment words? Several researchers have addressed this problem by supervised learning. Most work needs to use seed words (the known sentiment words), which are usually manually selected. It is well known that a word sense is often ambiguous without the help of the context. However, the content of a document is unambiguous, thus the sentiment of a document is more explicit than that of a word. Therefore, we think labeled documents (documents with sentiment labels) should be a useful resource when recognizing sentiment words. This observation inspires us to explore how to identify sentiment words by using labeled documents and seed words.

In this paper, we study the problem of automatically identifying sentiment words from the corpus, by an optimization model with L_1 regularization, called ISOMER (abbreviation for "**I**dentifying **S**entiment words using an **O**ptimization **M**odel with L_1 regulariz**er**"). The distinctive aspect of our approach is that ISOMER adopts both labeled documents and seed words, or either one if the other is hard to obtain.

In our model, the sentiment polarities of words are treated as parameters to be determined. The probabilistic estimation is composed of two parts, the generative probabilities of labeled documents and seed words. We use maximum likelihood estimation (MLE) to form objective function which hybridizes these two. Since most words in the corpus have no sentiment, we expect a sparse solution only providing nonzero values to the sentiment words. Therefore, the ℓ_1 regularizer, also called lasso penalty (Tibshirani 1996) is added to the objective function. After solving the problem, we obtain the sentiment polarity of each word and the corresponding strength, which is vital for determining the significant sentiment words. To the best of our knowledge, this paper is the first work to identify sentiment words by employing both labeled documents and sentiment words.

Our main contributions are summarized below:

(1) We study the problem of sentiment word identification and formulate it as an optimization problem, which employs both the document labels and seed words. Our approach can not only assign sentiment to each word, but also learn the strength.

(2) Since most candidate words have no sentiment, we introduce L_1 penalty to our model, yielding a sparse solution.

(3) The experiments on English and Chinese datasets demonstrate that our model outperforms the classic approaches for sentiment word identification. Further experiments show that the lexicon learned by our model can be effectively implemented to document-level sen-

timent analysis.

Related Work

Sentiment word identification is an important technique in sentiment analysis. According to the type of training resource, we categorize the approaches into *document-based approaches* and *graph-based approaches*. Document-based approaches extract the word relations from the documents, and learn the polarities with the help of seed words. Graph-based approaches construct a word network, using a structured dictionary such as WordNet (Miller 1995), and analyze the graph.

We first introduce some document-based approaches which are more relevant to our work. The work (Hatzivassiloglou and McKeown 1997) is an early research of sentiment word identification, aiming at adjectives only. The basic assumption is: conjunctions such as "and" connect two adjectives with the same sentiment orientation, while "but" usually connects two words with opposite orientations. After the estimation of conjunctions, a clustering algorithm separates the adjectives into groups of different orientations. In (Turney and Littman 2003) and (Kaji and Kitsuregawa 2007), the semantic polarity of a given word is calculated from the strength of its association with the positive word set, minus the strength of its association with the negative word set. The authors use statistical measures, such as point wise mutual information (PMI), to compute similarities in words or phrases. (Qiu et al. 2009) provide a semi-supervised framework which constantly exploits the newly discovered sentiment words to extract more sentiment words, until no more words can be added. (Chen et al. 2012) present an optimization-based approach to automatically extract sentiment expressions for a given target from unlabeled tweets. They construct two networks in which the candidate expressions are connected by their consistency and inconsistency relations. The objective function is defined as the sum of the squared errors for all the relations in two networks. Similarly, the work of (Lu et al. 2011) and (Amiri and Chua 2012) also employ optimization-based approaches to automatically identify the sentiment polarity of words. Unlike the previous approaches, the work of (Yu, Deng, and Li 2013) exploits the sentiment labels of documents rather than seed words. The authors 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.

Graph-based approaches are also important. (Takamura, Inui, and Okumura 2005) use spin model to automatically create a sentiment word list from glosses in a dictionary, a thesaurus or a corpus, by regarding word sentiment polarities as spins of electrons. The lexical network is constructed by linking two words if one word appears in the gloss of the other word. (Breck, Choi, and Cardie 2007) introduce some word features in their model, including lexical features to capture specific phrases, local syntactic features to learn syntactic context, and graph-based features to capture both more general patterns and expressions already known to be opinion-related. In (Hassan and Radev 2010) and (Hassan et al. 2011), the authors apply a random walk model to a large word relatedness graph, producing a polarity estimate for any given word. Several sources could be used to link words in the graph, and the synonyms and hypernyms in WordNet is their choice in the experiment.

In summary, the previous methods employ either seed words or labeled documents. Comparing to these studies, our model is able to employ both labeled documents and seed words. As far as we know, no similar approach has been proposed so far.

Sentiment Word Identification

In this section, we first formulate the problem of sentiment word identification. Before discussing our model, we introduce the concepts of surrogate polarity and sentiment strength. Then, we build a probabilistic framework. Next, the probability distributions in the framework are introduced. Finally, we give the corresponding solution and present the entire process of our algorithm.

Problem Formalization

We formulate the sentiment word identification problem as follows. Assume we have a sentiment corpus $\mathcal{D} = \{(d_1, y_1), \ldots, (d_n, y_n)\}$, where d_i is a document and y_i is the corresponding sentiment label. We suppose $y_i = 1$ if d_i is a positive document, and $y_i = -1$ if d_i is negative. Similarly, there are Q seed words in the sentiment lexicon $\mathcal{V} = \{(v_1, l_1), \ldots, (v_Q, l_Q)\}$, where v_i is the seed word and $l_i \in \{-1, 1\}$ is the sentiment polarity. The candidate word vocabulary is represented as $\mathcal{W} = \{w_1, \ldots, w_m\}$. Our goal is to find the sentiment words from $\mathcal{W}$, and also give their confident values.

The Proposed Method

Surrogate polarity Before discussing our model, we first introduce the concepts of **surrogate polarity** and **sentiment strength**. The sentiment polarity of a word is always limited to some discrete values, e.g. {"positive", "negative"} or {1,-1}. In our model, the general strategy is to infer a *surrogate polarity*, a real number, for every word in the candidate set $\mathcal{W}$. Let s_i denote the surrogate polarity of word w_i. w_i is classified as having a positive sense if $s_i > 0$, and a negative sense if $s_i < 0$. Therefore, the surrogate polarity can be regarded as the extension of the discrete polarity. The magnitude of the surrogate polarity is the strength of the sentiment. A high value of $|s_i|$ implies w_i is likely to be a sentiment word.

As we are provided with labeled corpus $\mathcal{D}$ and seed words $\mathcal{V}$, we can establish an optimization model to obtain surrogate polarities of candidate words. Therefore, sentiment word identification is to infer $\vec{s}$ by minimizing the loss function: $\vec{s^*} = \arg\min_{\vec{s}} L(\vec{s}; \mathcal{D}, \mathcal{V})$. The selection of $L(\cdot)$ is to determine how the labels of training corpus and words can be reconstructed/predicted by $\vec{s}$. To avoid overfitting, 2-norm penalty is often used to the objective function: $\vec{s^*} = \arg\min_{\vec{s}} L(\vec{s}; \mathcal{D}, \mathcal{V}) + \beta \cdot \|\vec{s}\|_2$. However, most elements s_i of the corresponding solution are non-zero, so that it is difficult to distinguish polarity words and neutral words. Because most candidate words have no sentiment, we expect their surrogate polarities to be 0. In another word, a sparse $\vec{s^*}$, only providing a non-zero value $s_i \neq 0$ when w_i is a sentiment word, is needed. Therefore, we impose 1-norm (lasso) penalty on $\vec{s}$ instead of 2-norm to yield sparse solution:

$$\vec{s^*} = \arg\min_{\vec{s}} L(\vec{s}; \mathcal{D}, \mathcal{V}) + \beta \cdot \|\vec{s}\|_1. \quad (1)$$

Framework Since we have a sentiment corpus $\mathcal{D}$ and a sentiment lexicon $\mathcal{V}$, we build a probabilistic model for them. Assume the documents in $\mathcal{D}$ are conditionally independent given the surrogate polarity of each word, then the probability of generating the labels of $\mathcal{D}$ is:

$$p(y_{1:n}|d_{1:n}, \vec{s}) = \prod_{i=1}^{n} p(y_i|d_i, \vec{s}), \quad (2)$$

where $\vec{s} = \begin{bmatrix} s_1 \\ \vdots \\ s_m \end{bmatrix}$.

We propose a similar strategy for the sentiment lexicon. Suppose a graph whose nodes denote words and edges denote the relations between words, extracted from the corpus. There are two types of nodes, seed nodes denoting seed words and candidate nodes denoting candidate words. Then the graph is simplified by ignoring the homogeneous relations, i.e. only the relations between seed nodes and candidate nodes are considered. The word graph then becomes a bipartite graph. Accordingly, we obtain the following conditional probability for the seed words similar to Formula (2):

$$p(l_{1:Q}|v_{1:Q}, \vec{s}) = \prod_{i=1}^{Q} p(l_i|v_i, \vec{s}) \triangleq \prod_{i=1}^{Q} p(l_i|\vec{r_i}, \vec{s}), \quad (3)$$

where $\vec{r_i} = \begin{bmatrix} r_{i1} \\ \vdots \\ r_{im} \end{bmatrix}$ and each element r_{ij} represents the relation between seed word v_i and candidate word w_j.

We denote the log value of Formula (2) and (3) as $\ell_{\text{doc}}(\vec{s}) = \log p(y_{1:n}|d_{1:n}, \vec{s})$ and $\ell_{\text{word}}(\vec{s}) = \log p(l_{1:Q}|v_{1:Q}, \vec{s})$ respectively. The values of $\vec{s}$ will be learned by maximum likelihood estimation which combines the two objectives:

$$\begin{aligned} \vec{s^*} &= \arg\max_{\vec{s}} \ \ell(\vec{s}) \\ &= \arg\max_{\vec{s}} \lambda \frac{\ell_{\text{doc}}(\vec{s})}{N} + (1-\lambda) \frac{\ell_{\text{word}}(\vec{s})}{Q}, \end{aligned} \quad (4)$$

where $0 \leq \lambda \leq 1$ is the linear combination coefficient. Observe that $\vec{s^*}$ is computed by considering only the corpus when $\lambda = 1$, or by considering only seed words when $\lambda = 0$. $\frac{\ell_{\text{doc}}(\vec{s})}{N}$ and $\frac{\ell_{\text{word}}(\vec{s})}{Q}$ are the "average log likelihood" values of the documents and seed words, making the two terms comparable.

As shown by Formula (1), we impose an ℓ_1 regularizer on $\vec{s}$. In order to rewrite the problem as the minimization form as Formula (1), we denote the negative log likelihood $Nll_{\text{doc}}(\vec{s}) = -\ell_{\text{doc}}(\vec{s})$ and $Nll_{\text{word}} = -\ell_{\text{word}}(\vec{s})$. The problem becomes:

$$\min_{\vec{s}} \ \lambda \frac{Nll_{\text{doc}}(\vec{s})}{N} + (1-\lambda) \frac{Nll_{\text{word}}(\vec{s})}{Q} + \beta \cdot \|\vec{s}\|_1, \quad (5)$$

where $\beta \geq 0$ is a tuning parameter. The term $\beta \cdot \|\vec{s}\|_1$ is also called a lasso penalty. The first two terms constitute the loss function $L(\vec{s}; \mathcal{D}, \mathcal{V})$. After solving the problem, we obtain the positive words having positive surrogate polarities, and negative words having negative surrogate polarities.

Model Specification In this section, we introduce the ℓ_1 regularized logistic regression (Ng 2004) to the problem of sentiment word identification for the first time. This model specifies the sentiment distributions and yields a sparse solution.

Document probability

To specify the conditional probability of a document's polarity $p(y_i|d_i, \vec{s})$, the document d_i is represented by vector space model (VSM). An $n \times m$ document-word matrix $\mathbf{F} = \begin{bmatrix} f_{11} & \cdots & f_{1m} \\ \vdots & \ddots & \vdots \\ f_{n1} & \cdots & f_{nm} \end{bmatrix}$ is constructed, where f_{ij} describes the importance of candidate word w_j to d_i. *TF* and *TF-IDF*(Jones 1972) are two widely used functions to compute f_{ij}.

The i^{th} row of $\mathbf{F}$ is the bag-of-words features of document d_i. Accordingly, d_i's feature vector can be denoted as $\vec{f}_i^{\text{doc}} = [f_{i1}, \ldots, f_{im}]^T$.

We use $p(y_i|\vec{f}_i^{\text{doc}}, \vec{s})$ to substitute $p(y_i|d_i, \vec{s})$. In the logistic regression, the conditional probability has a Bernoulli distribution, i.e. $y_i|\vec{f}_i^{\text{doc}}, \vec{s} \sim \text{Ber}\left(y_i|\sigma(\vec{s}^T \cdot \vec{f}_i^{\text{doc}})\right)$, where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function. Hence the negative log likelihood Nll_{doc} is:

$$\begin{aligned} Nll_{\text{doc}}(\vec{s}) &= -\sum_{i=1}^{n} \log p(y_i|d_i, \vec{s}) \\ &= -\sum_{i=1}^{n} \log p(y_i|\vec{f}_i^{\text{doc}}, \vec{s}) \\ &= \sum_{i=1}^{N} \log\left(1 + \exp(-y_i \vec{s}^T \cdot \vec{f}_i^{\text{doc}})\right). \end{aligned} \quad (6)$$

Seed word probability

We construct an analogous document-word matrix $\mathbf{G} \in \mathbb{R}^{n \times Q}$ for the seed words. Similar to the document feature vectors, the columns of document-word matrix $\mathbf{G}$ or $\mathbf{F}$ can be regarded as feature vectors of the seed words and candidate words. Denote $\vec{g}_i^{\text{word}}$ and $\vec{f}_j^{\text{word}}$ as the the feature vectors of v_i and w_j. In this paper, the relation value between v_i and w_j is proportional to the cosine similarity of the their feature vectors, i.e. $r_{ij} \propto cosine(\vec{g}_i^{\text{word}}, \vec{f}_j^{\text{word}})$. By such method, we obtain the relation vector $\vec{r_i}$ for every seed word.

Similar to the conditional probability of the document, $p(l_i|\vec{r_i}, \vec{s})$ also obeys the Bernoulli distribution as $l_i|\vec{r_i}, \vec{s} \sim \text{Ber}\left(l_i|\sigma(\vec{s}^T \cdot \vec{r_i})\right)$. Then the negative log likelihood Nll_{word} can be written as:

$$\begin{aligned} Nll_{\text{word}}(\vec{s}) &= -\sum_{i=1}^{n} \log p(l_i|\vec{r_i}, \vec{s}) \\ &= \sum_{i=1}^{Q} \log\left(1 + \exp(-l_i \vec{s}^T \cdot \vec{r_i})\right). \end{aligned} \tag{7}$$

We note that $\vec{f}_i^{\text{doc}}$ and $\vec{r_j}$ are normalized in our model to avoid the bias of vector length.

Solution and Algorithm

Since $Nll_{\text{doc}}(\vec{s})$, $Nll_{\text{word}}(\vec{s})$ and $\|\vec{s}\|_1$ are convex functions, their nonnegative weighted sum, i.e. the loss function in Problem (5), preserves convexity (Boyd and Vandenberghe 2004). However, the objective function is not differentiable because of the ℓ_1 regularizer. We follow the *sub-gradient* proposed in (Schmidt, Fung, and Rosales 2007) to solve Problem (5). Here the sub-gradient for the objective function is defined as:

$$\nabla_i f(\vec{s}) = \begin{cases} x_i + \beta \cdot sign(s_i) & |s_i| > 0 \\ x_i - \beta \cdot sign(x_i) & s_i = 0, |x_i| > \beta \\ 0 & s_i = 0, |x_i| \leq \beta, \end{cases} \tag{8}$$

where $x_i = \nabla_i Nll(\vec{s})$. Now we can solve Problem (5) by general convex optimization algorithms. In the iteration steps, we update $\vec{s}$ by $\Delta\vec{s} = -\eta \cdot \nabla_{\vec{s}} f$, where $\eta > 0$ is the learning rate. Note that the sparsity of $\vec{s}^*$ is guaranteed, for the component elements with insignificant gradients remain unchanged according to the third line of Formula (8).

Experiment

In this section, we present the experimental results of our model compared with other baseline approaches.

Experimental Setup

Data Set To evaluate our method we leverage two English datasets and one Chinese dataset as the source of training corpus. The Cornell Movie Review Data [1], first used in (Pang, Lee, and Vaithyanathan 2002), is a widely used benchmark. This corpus contains 1,000 positive and 1,000 negative processed reviews of movies, extracted from the Internet Movie Database. The other corpus is the Stanford Large Movie Review Dataset[2] (Maas et al. 2011). (Maas et al. 2011) constructed a collection of 50,000 reviews from IMDB, half of which are positive reviews and half negative. We use MPQA subjective lexicon [3] to generate the gold standard. Only the strongly subjective clues are considered as sentiment words, consisting of 1717 positive and 3621 negative words.

[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/

Dataset	Word Set	#pos	#neg	#non	#total
Cornell	seed	143	190	-	333
	candidate	534	817	1262	2613
Stanford	seed	219	344	-	563
	candidate	791	1369	2618	4778
Chinese	seed	86	171	-	257
	candidate	362	648	1277	2287

Table 1: Word Distribution

As for the Chinese dataset, 5 unbiased assessors download 1,000 news reports from Sina News[4], containing 500 positive and 500 negative articles. To construct the golden standard, the assessors are asked to manually select the sentiment words from these articles by thoroughly reading them. A word with at least 2 votes is regarded as a sentiment word. 788 positive words and 2319 negative words are chosen to form the ground truth at last.

Word Selection The ground truth of English sentiment word set is shaped by the intersection of words in the sentiment lexicon and vocabulary of each corpus. We randomly select 20% words in sentiment word set as the seed words, and the remaining are candidate words. In order to simulate the real situation where we cannot distinguish sentiment words before running the algorithms, several *neutral words* are added to the candidate word set. To form the Chinese candidate word set, all documents are segmented using the word segmentation tool ICTCLAS [5]. Table 1 shows the average word counts for each dataset. About half of the candidate words are neutral. In the following experiments, we randomly generate the seed words and candidate words 10 times for each dataset.

Baseline Methods In order to demonstrate the effectiveness of our model, we select two classic methods, namely SO-PMI and COM, as baselines.

- **SO-PMI**: SO-PMI (Turney and Littman 2003) is a classic approach of sentiment word identification. This method calculates the value of SO-PMI of each word, i.e. the strength difference between its association with the positive word set and the negative word set. The sentiment orientation of word c is assigned according to the sign of $SO\text{-}PMI(c)$.
- **COM**: Our second baseline, COM (Chen et al. 2012), is an optimization-based approach. In the model, two word networks of consistency and inconsistency relations are constructed. The objective function is built to minimize the sum of squared errors for these two networks, with the seed words polarities as its prior information. The solution of the model indicates the polarity of each words.

Overall Evaluation

In this section, we choose *precision*, *recall* and *F-score* to evaluate the performance of each method. In the following

[4]http://news.sina.com.cn/

[5]http://ictclas.org/

	Cornell			Stanford			Chinese		
	P	R	F	P	R	F	P	R	F
ISOMER	**0.4454**	**0.6103**	**0.5150**	**0.4441**	**0.6399**	**0.5243**	**0.5068**	**0.5684**	**0.5358**
SO-PMI	0.4096	0.5724	0.4775	0.3938	0.6011	0.4758	0.4488	0.5108	0.4777
COM	0.3881	0.5346	0.4497	0.3806	0.5674	0.4556	0.4636	0.5523	0.5041

Table 2: Overall Evaluation. Since many neutral words are added to the candidate set, the task is much more challenging than simply binary classification.

experiments, TF-IDF is used as the word weighting scheme to compute f_{ij} in our model.

Result The overall results of three approaches are shown in Table 2. It is noteworthy that *this task is much more challenging than simply binary classification*, since a large number of neutral words are added to the candidate set.

We find that ISOMER achieves the best *precision*, *recall* and *F-score* on all datasets. It shows our method dominates other approaches. SO-PMI performs better than COM on two English datasets but COM performs better on the Chinese dataset. The performance of COM may be limited by the construction of its word networks. The consistency relations, requiring no negation applied or no contrasting conjunction connecting two words, may many cover redundant relations. As for SO-PMI, it performs poorly on Chinese dataset. One possible explanation is that the PMI score reflects the co-occurrence frequency of two words even if their positions are with a great distance and no semantic relatedness. Chinese news articles are always longer than the English movie reviews so that PMI will record numerous such long-distant word pairs. In addition, PMI's bias towards infrequent words may lead to bad results.

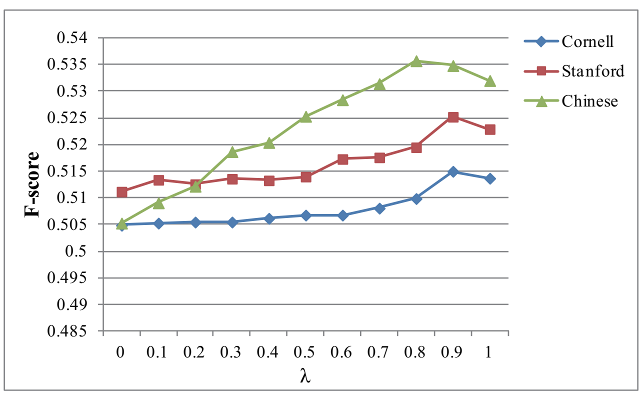


Figure 1: $F - score$ by varying λ.

Document Sentiment vs. Word Sentiment To compare the contribution of the labeled sentiment of documents and words in identifying the unknown sentiment words, we conduct extensive experiments for various λ values, linear combination coefficient that determines the importance of two likelihood functions. Figure 1 shows the results of ISOMER in terms of *F-score* when λ varies. The model only considers the labeled documents when $\lambda = 1$ and only considers seed words when $\lambda = 0$. The figure shows when λ is close to 1

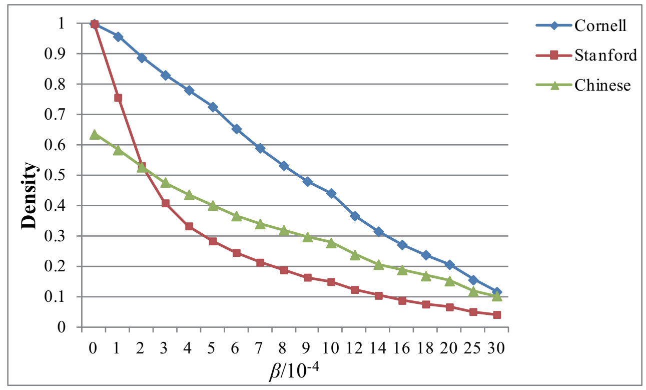


Figure 2: Density by varying regularizer β.

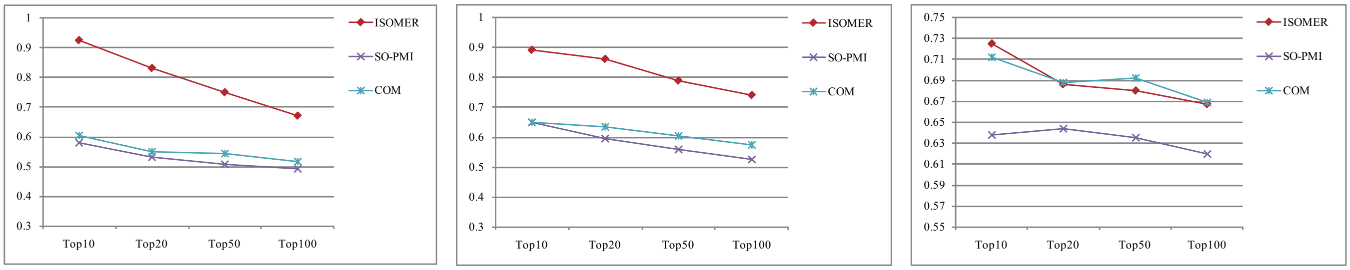
the model achieves better result in all three datasets. Such phenomenon indicates documents can more unequivocally express sentiment than words do due to the context information. More specifically, the best *F-score*s are achieved when $\lambda = 0.8$ and 0.9 for the Chinese corpus and two English corpuses (Cornell and Stanford). We adopt the above settings in our experiments.

Effect of Regularizer In our model, the tuning parameter β determines the proportion of selected sentiment words in the candidate set, called *"density"*. As β increases, the regularizer tends to select fewer and more significant words. One can choose different values of β according to the requirement of sentiment dictionary for different applications: a small value of β gives more recommendations and a greater value of β makes more accurate results. For convenience of comparing with other methods, we choose β for each dataset which enables the density to approximately equal to the real value, i.e. $\beta = 2 \times 10^{-4}$ for Stanford and Chinese datasets and $\beta = 9 \times 10^{-4}$ for Cornell dataset.

Figure 2 shows how the density varies with β for three datasets. We can see that the density decreases quickly on the large dataset (Stanford). The reason is high dimensional $\vec{s}$ always has a larger value of 1-norm than the low when the density is fixed, thus it is very sensitive to the penalty parameter. The decreasing trend on the other two datasets are nearly linear.

Top-K Test

In the applications like active learning classification, we only select several most informative samples for manual annotation. In this experiment, we identify the positive and negative



(a) Cornell Dataset (b) Stanford Dataset (c) Chinese Dataset

Figure 3: Top-K Test. We test the precision of the most confident sentiment words of each method.

Method \ Weighting	Cornell			Stanford			Chinese		
	Boolean	BM25	TF-IDF	Boolean	BM25	TF-IDF	Boolean	BM25	TF-IDF
ISOMER	**0.833**	**0.837**	**0.816**	**0.778**	**0.785**	**0.757**	**0.918**	**0.916**	**0.908**
SO-PMI	0.701	0.707	0.685	0.688	0.700	0.612	0.834	0.848	0.796
COM	0.735	0.738	0.722	0.696	0.695	0.631	0.882	0.838	0.834
CHI	0.827	0.817	0.814	0.774	0.777	0.730	0.906	0.912	0.902
IG	0.820	0.818	0.812	0.754	0.778	0.731	0.912	0.900	0.906

Table 3: Document-level sentiment classification by 200 sentiment words.

words from the candidate word set by three methods and obtain their sentiment strengths as well. After that we evaluate the accuracy of the *Top-K* sentiment words, i.e. K positive and K negative words with highest strengths.

Figure 3 shows the result. ISOMER outperforms other approaches on all three datasets. As results on Cornell and Stanford datasets suggest, ISOMER has remarkable advantage over the other two baselines on English corpuses. For the Chinese dataset, on the other hand, ISOMER and COM achieve significantly higher precisions compared with SO-PMI. As an additional insight from Figure 3(c), we point out that COM, while performing poorly on the former two English datasets, has an amazingly high precision on Chinese dataset. It suggests that the concept of consistency and inconsistency might be more relevant for Chinese corpus.

Document-level Sentiment Classification

To evaluate the usefulness of the learned sentiment lexicons, we apply them to document-level sentiment classification. In this experiment, the top-ranked sentiment words, 100 positive and 100 negative words, extracted by each method are used as the features of the classifier. In addition, two widely used methods, CHI (χ^2 statistic), and IG (information gain), are introduced as baselines. CHI and IG are two statistical functions to learn the sentiment weight of the word by using the *labeled documents*. Since these two methods cannot give the polarities of words, we choose 200 words with the highest values as the features. After feature selection, we use *boolean* (Pang, Lee, and Vaithyanathan 2002), *TF-IDF* and *BM25* (Robertson, Zaragoza, and Taylor 2004) as the term weighting strategy. We calculate the *precision* on 10-fold cross-validation by SVM classifier [6].

As shown in Table 3, ISOMER-based classifiers achieve best results in all datasets, indicating that our method can recognize the representative and frequently used sentiment words with high accuracy, and document-level sentiment analysis can indeed benefit from such lexicon. Among the baseline methods, CHI and IG-based classifiers give more reasonable results than SO-PMI and COM-based classifiers, because the former can take the labeled documents into account. We also note that although COM performs well on Chinese dataset in "Top-K Test", COM-based classifier does not achieve high precision. It may be because the words provided by COM are not frequently used in the news articles even if they are correct.

Conclusion and Future Work

In this paper, we propose an optimization model with L_1 penalty, called ISOMER, to identify sentiment words. L_1 penalty induces a sparse solution since most candidate words have no sentiment. The experiments on the real datasets show that ISOMER outperforms the classic approaches. Good performance on English and Chinese datasets indicates ISOMER has high generalization ability and robustness for sentiment word identifying of different languages. Furthermore, the lexicon learned by ISOMER can be effectively adapted to document-level sentiment analysis.

Sentiment word identification plays a fundamental work in multiple applications of sentiment analysis and opinion mining. Our future work extends into some of these fields after constructing the sentiment lexicon using our model.

[6] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Acknowledgments

This work is partially supported by the National High Technology Research and Development Program of China (Grant No. 2015AA015403) and the National Natural Science Foundation of China (Grant No. 61170091). We would also like to thank the anonymous reviewers for their helpful comments.

References

Amiri, H., and Chua, T.-S. 2012. Mining slang and urban opinion words and phrases from cqa services: an optimization approach. In *Proceedings of the Fifth International Conference on Web Search and Web Data Mining*, 193–202.

Boyd, S., and Vandenberghe, L. 2004. *Convex optimization*. Cambridge university press.

Breck, E.; Choi, Y.; and Cardie, C. 2007. Identifying expressions of opinion in context. In *Proceedings of the 20th international joint conference on Artifical intelligence*, 2683–2688.

Chen, L.; Wang, W.; Nagarajan, M.; Wang, S.; and Sheth, A. 2012. Extracting diverse sentiment expressions with target-dependent polarity from twitter. In *Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media*, 50–57.

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

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

Hassan, A.; Abu-Jbara, A.; Jha, R.; and Radev, D. 2011. Identifying the semantic orientation of foreign words. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 592–597.

Hatzivassiloglou, V., and McKeown, K. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, 174–181.

Jones, K. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation* 28(1):11–21.

Kaji, N., and Kitsuregawa, M. 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*, 1075–1083.

Kim, S.-M., and Hovy, E. 2004. Determining the sentiment of opinions. In *Proceedings of the 20th international conference on Computational Linguistics*, 1367.

Liu, B. 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies* 5(1):1–167.

Lu, Y.; Castellanos, M.; Dayal, U.; and Zhai, C. 2011. Automatic construction of a context-aware sentiment lexicon: an optimization approach. In *Proceedings of the 20th International Conference on World Wide Web*, 347–356.

Maas, A.; Daly, R.; Pham, P.; Huang, D.; Ng, A.; and Potts, C. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational Linguistics*.

Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM* 38(11):39–41.

Ng, A. Y. 2004. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, 78.

Pang, B., and Lee, L. 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval* 2(1-2):1–135.

Pang, B.; Lee, L.; and Vaithyanathan, S. 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*, 79–86.

Popescu, A.-M., and Etzioni, O. 2007. Extracting product features and opinions from reviews. In *Natural language processing and text mining*. Springer. 9–28.

Qiu, G.; Liu, B.; Bu, J.; and Chen, C. 2009. Expanding domain sentiment lexicon through double propagation. In *Proceedings of the 21st international jont conference on Artifical intelligence*, 1199–1204.

Robertson, S.; Zaragoza, H.; and Taylor, M. 2004. Simple bm25 extension to multiple weighted fields. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, 42–49.

Schmidt, M.; Fung, G.; and Rosales, R. 2007. Fast optimization methods for l1 regularization: A comparative study and two new approaches. In *Machine Learning: ECML 2007*. Springer. 286–297.

Takamura, H.; Inui, T.; and Okumura, M. 2005. Extracting semantic orientations of words using spin model. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 133–140.

Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 267–288.

Turney, P., and Littman, M. 2003. Measuring praise and criticism: Inference of semantic orientation from association.

Yu, H.; Deng, Z.-H.; and Li, S. 2013. Identifying sentiment words using an optimization-based model without seed words. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2)*, 855–859.