

Adapting Information Bottleneck Method for Automatic Construction of Domain-oriented Sentiment Lexicon

Weifu Du¹, Songbo Tan², Xueqi Cheng² and Xiaochun Yun²

¹Haerbin Institute of Technology, Haerbin, China

²Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

{duweifu,tansongbo}@software.ict.ac.cn

ABSTRACT

Domain-oriented sentiment lexicons are widely used for fine-grained sentiment analysis on reviews; therefore, the automatic construction of domain-oriented sentiment lexicon is a fundamental and important task for sentiment analysis research. Most of existing construction approaches take only the kind of relationships between words into account, which makes them have a lot of room for improvement. This paper proposes an adapted information bottleneck method for the construction of domain-oriented sentiment lexicon. This approach can naturally make full use of the mutual reinforcement between documents and words by fusing three kinds of relationships either from words to documents or from words to words; either homogeneous or heterogeneous; either within-domain or cross-domain. The experimental results demonstrate that proposed method could dramatically improve the accuracy of the baseline approach on the construction of out-of-domain sentiment lexicon.¹

Categories and Subject Descriptors

I.2.7 [Artificial Intelligence]: Natural Language Processing; H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval; I.5 [Pattern Recognition]: Applications

General Terms

Algorithms, Performance, Experimentation

Keywords

Sentiment Analysis; Opinion Mining; Information Retrieval

1. INTRODUCTION

In the Web2.0 era, the Internet turns from a static information media into a platform for dynamic information exchanging, on which people can express their views and show their individualities. More and more people are willing to record their feelings (blog), give voice to public affairs (news review), express their likes or dislikes on products (product review), and so on. In

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WSDM '10, February 4–6, 2010, New York City, New York, USA.
Copyright 2010 ACM 978-1-60558-889-6/10/02...\$10.00.

the face of the increasing volume of sentimental information available on the Internet, there is a growing interest in helping people to better find, filter, and manage these resources.

Automatic sentiment analysis [1][9][13][17][19-26] could play an important role in a wide variety of flexible and dynamic information management tasks. For example, with the help of sentiment analysis system, in the field of public administration, the administrators could receive the feedback on one policy in a timelier manner; in the field of business, manufacturers could perform more targeted updates on products to improve the consumer's experience.

In order to infer sentiment orientation of reviews in different domains, one of the commonly-used methods is to build a general sentiment lexicon. However, it is an impossible task to build a general sentiment lexicon that could perform well in every domain, because sentiment expression often behaves with strong domain-specific nature [2]. In other words, in each different domain, the sentiments are apt to be expressed by their own domain-specific features. For example, "rise" and "rebound" are often used to express positive sentiment for stock review; while "luxury" and "classical" are often employed to convey positive sentiment for house review. This so-called domain-specific nature makes it an important job to design an automated approach that could build a sentiment lexicon for each new domain.

In this paper, we assume a typical application scenario: We have a labeled document set D_i and a sentiment lexicon W_i (a word list with sentiment polarity label) from one domain which is called in-domain, and another document set D_o from a related but different domain which is called out-of-domain. The latter is unlabeled and we want to build a specified sentiment lexicon W_o for it can make full use of in-domain knowledge.

So far, two kinds of approaches have been proposed to deal with this problem. One is based on a thesaurus. This method utilizes synonyms or glosses of a thesaurus to determine polarity of words [5][9][10][12]. The second approach exploits raw corpus. Polarity is decided by using co-occurrence in a corpus. This approach is based on a hypothesis that polar terms conveying the same polarity co-occur with each other. Typically, a small set of paradigm polar terms are prepared, and new polar terms are detected based on the strength of co-occurrence with the seeds [8][11][17].

Most of existing approaches take the homogeneous relationship between words (i.e., relationship between out-of-domain words and in-domain words (WW_{inter} -Relationship)) into account, while ignore the other two kinds of heterogeneous

¹ This work is done at Institute of Computing Technology.

relationships (i.e., relationship between out-of-domain words and out-of-domain documents (WD_{intra} -Relationship), relationship between out-of-domain words and in-domain documents (WD_{inter} -Relationship)). Consequently, there is a room for improvement and it is still a challenge to find more beneficial guidance from in-domain data for the construction of out-of-domain sentiment lexicon.

To address this issue, we aim to take into account all of the three kinds of relationships: WD_{intra} -Relationship, WW_{inter} -Relationship, and WD_{inter} -Relationship. In this work, we propose an iterative reinforcement approach to implement the above inspiration. The main idea is to adapt information bottleneck method [15] by incorporating the three kinds of relationships. As a result, our approach could be considered as a sentiment-lexicon-construction version of information bottleneck method.

To investigate the effectiveness and robustness of this approach, we conduct an extensive experiment on three domain-specific sentiment corpora, including electronic product reviews, hotel reviews, and stock reviews. The experimental results indicate that proposed approach can dramatically improve the performance of the baseline approach on the construction of out-of-domain sentiment lexicon.

2. RELATED WORK

In this section, we review several prior works mostly related to our work, including sentiment lexicon construction, and information bottleneck method.

Most of previous methods about lexicon construction use term similarity and some paradigm terms to construct sentiment lexicon. The basic observations underlying these methods are quite different from each other. However, these methods could roughly be classified into two categories in terms of the manner of obtaining term similarity, the first kind of approaches based on the thesaurus, and the second kind of approaches based on corpus.

2.1 Thesaurus Based Approach

Kamps et al. [10] built lexical network by linking synonyms provided by a thesaurus, and term polarity was defined by the distance from seed words (“good” and “bad”) in the network. This method relies on a hypothesis that synonyms have the same polarity. Hu and Liu [9] used similar lexical network, but they considered not only synonyms but also antonyms. Kim and Hovy [12] proposed two probabilistic models to estimate the strength of polarity. In their models, synonyms are used as features. Esuli and Sebastiani [5][6] utilized glosses of words to determine polarity.

Compared with our approach, the drawback of using a thesaurus is the lack of scalability. It is difficult to handle such words that are not contained in a thesaurus (e.g. newly-coined words or colloquial words).

2.2 Corpus Based Approach

Another approach is based on an idea that polar terms conveying the same polarity co-occur with each other in corpus. Turney’s work [17] is one of the most famous works that discussed learning polarity from corpus. Turney determined polarity value based on co-occurrence with seed words (“excellent” and “poor”). The co-occurrence is measured by the number of hits returned by a search engine. The polarity value proposed by [17] is as follows.

$$\log_2 \frac{hits(c\ NEAR\ excellent)hits(poor)}{hits(c\ NEAR\ poor)hits(excellent)}$$

where $hits(q)$ means the number of hits returned by a search engine when query q is issued. *NEAR* means NEAR operator, which enables to retrieve only such documents that contain two queries within ten words.

The basic assumption of Turney’s method is that sentiment terms of similar orientation tend to co-occur at the document level. Gamon and Aue [7] extended Turney’s method by adding one assumption that sentiment terms of opposite orientation tend not to co-occur at the sentence level.

Hatzivassiloglou and McKeown [8] constructed lexical network and determine polarity of adjectives. Although this is similar to the thesaurus-based approaches, they built the network from intra-sentential co-occurrence. Takamura et al. built lexical network from not only such co-occurrence but also other resources including a thesaurus [16]. They used spin model to predict polarity of words.

Popescu and Etzioni [14] applied relaxation labeling to polarity identification. This method iteratively assigns polarity to words by using various features including intra-sentential co-occurrence and synonyms of a thesaurus.

Kanayama and Nasukawa [11] used both intra- and inter-sentential co-occurrence to learn polarity of words and phrases. Their method covers wider range of co-occurrence than other work such as [8].

2.3 Information Bottleneck Method

The information bottleneck method [15] provides an information theoretic framework, for extracting features of one variable that are relevant for the values of another variable. For instance, in the process of word clustering, the relationship between documents and words is taken into account, which motives us to take it as the kernel of our approach. Because the in-domain knowledge is not utilized in the traditional IB method, so we adapt the method to integrate these in-domain knowledge, which will be illustrated in detail in section 3, and the experimental result shows the effectiveness of the extension.

3. PROPOSED ALGORITHM

3.1 The Problem

Let D_i be the set of in-domain documents with sentiment polarity labels, W_i be the sentiment lexicon labeled from D_i , and terms in W_i are all labeled as positive or negative. D_o be the set of out-of-domain data without polarity labels. In our task, we attempt to design a feasible approach to construct a domain-specific sentiment lexicon W_o for the out-of-domain data. More precisely, we want to group the out-of-domain sentiment words W_o into 2 clusters (i.e., positive or negative). Let \hat{D}_o denote the out-of-domain documents clustering, and \hat{W}_o denote the word clusters. Then the word cluster function C_{W_o} and document cluster function C_{D_o} can be defined as

$$C_{W_o}(w) = \hat{w}, \text{ where } w \in \hat{w} \wedge \hat{w} \in \hat{W}_o \quad (1)$$

$$C_{D_o}(d) = \hat{d}, \text{ where } d \in \hat{d} \wedge \hat{d} \in \hat{D}_o \quad (2)$$

where \hat{w} represents the word cluster that w belongs to and \hat{d} represents the document cluster that d belongs to.

Since the out-of-domain data are all unlabelled, the key point of our work is to investigate how to utilize the knowledge, i.e. D_i and W_i , available from the in-domain data in an efficient way.

Proposed approach is intuitively based on the following assumptions:

Assumption 1: A document should be positive (or negative) if it contains many positive (or negative) words, and a word should be positive (or negative) if it appears in many positive (or negative) documents.

Assumption 2: Even though the two domains may be under different distributions, we are able to identify a common part between them (e.g. the same word behaves the same orientation).

The first assumption makes use of mutual ‘recommendations’ between documents and words. Under the second assumption, we can extract and propagate knowledge and clues from in-domain data to out-of-domain data to guide the clustering. In more detail, the following three kinds of relationships are fused in the proposed approach:

WD_{intra}-Relationship: it reflects the heterogeneous relationship between out-of-domain words W_o and out-of-domain documents D_o .

WW_{inter}-Relationship: it reflects the homogeneous relationship between out-of-domain words W_o and in-domain words W_i .

WD_{inter}-Relationship: it reflects the heterogeneous relationship between out-of-domain words W_o and in-domain documents D_i .

In this study, the three kinds of relationships are all measured in a unique information theoretic framework, which will be clearly mathematically defined in section 3.3. Figure 1 gives an illustration of the relationships.

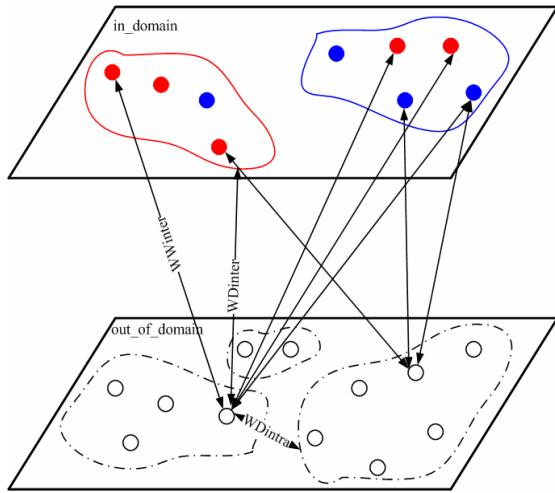


Figure 1: Illustration of the Relationships

There are two planes in this figure, the upper one denotes the in-domain data and the lower one denotes the out-of-domain data. In the in-domain plane, the blobs circled by a solid line denote documents and the solid nodes in each blob denote the terms within the document. Since we can obtain the polarity of the in-domain data, we use red and blue solid lines to denote the positive and negative sentiment respectively (for documents and terms). For the out-of-domain data, we use dotted lines and hollow nodes to denote the polarity of documents and terms, respectively.

From our observation, we can utilize three kinds of information to guide identification of the polarity of out-of-domain terms, which are figured by gray bidirectional arrow lines, including one homogeneous relationship (WW_{inter}-Relationship) and two heterogeneous relationships (WD_{inter}-Relationship and WD_{intra}-Relationship).

In the process of the construction of out-of-domain sentiment lexicon, we fuse the three relationships to group the out-of-domain words into two disjoint subgroups, and utilize the sentiment labels obtained from in-domain data to infer the semantic orientation of out-of-domain words, simultaneously.

3.2 Information Bottleneck Method

The information bottleneck method (IB) was proposed by Slonim and Tishby [15]. According to Shannon’s information theory [4], for two random variables X, Y , the mutual information $I(X; Y)$ between the random variables X, Y is given by a symmetric function:

$$I(X; Y) = \sum_{x \in X, y \in Y} p(x)p(y|x) \log \frac{p(y|x)}{p(y)} \quad (3)$$

which is the only consistent statistical measure of the information that variable X contains about variable Y (and vice versa).

The IB method is based on the following simple idea. Given the empirical joint distribution of two variables, one variable is compressed so that the mutual information about the other variable is preserved as much as possible. The method can be considered as finding a minimal sufficient partition or efficient relevant coding of one variable with respect to the other one. Roughly speaking, some of the mutual information will be lost in the process of compression, e.g. $I(C; Y) \leq I(X; Y)$ (C is a compressed representation of X). This problem can be solved by introducing a Lagrange multiplier β , and then minimizing the function:

$$L[p(c|x)] = I(C; X) - \beta I(C; Y) \quad (4)$$

The information bottleneck principle determines the distortion measure between the points x and c to be the Kullback-Leibler divergence [4] between the conditional distributions $p(y|x)$ and $p(y|c)$,

$$D_{KL}[p(y|x) || p(y|c)] = \sum_y p(y|x) \log \frac{p(y|x)}{p(y|c)} \quad (5)$$

The single positive (Lagrange) parameter β determines the ‘softness’ of the classification. When $\beta \rightarrow \infty$, there is a simple implementation of the information bottleneck method, restricted to the case of ‘hard’ clusters. In this case, every $x \in X$ belongs to precisely one cluster $c \in C$.

The algorithm starts with a trivial partitioning into $|X|$ singleton clusters, where each cluster contains exactly one element of X . At each step we merge two components of the current partition into a single new component in a way that locally minimizes the loss of mutual information about the categories, given in $I(C;Y)$. The decrease in the mutual information $I(C;Y)$ due to one merger is defined by

$$\delta I(c_i, c_j) \equiv I(C_{\text{before}}, Y) - I(C_{\text{after}}, Y) \quad (6)$$

where $I(C_{\text{before}}, Y)$ and $I(C_{\text{after}}, Y)$ are the information values before and after the merger, respectively.

By introducing the information optimization criterion, the resulting similarity measure directly emerges from the analysis. The algorithm is now very simple. At each step the IB algorithm perform ‘‘the best possible merger’’, i.e., merge the clusters $\{c_i, c_j\}$ which minimize $\delta I(c_i, c_j)$. For more details, please refer to [15].

3.3 Adapted Information Bottleneck Method for In-domain Knowledge

In traditional information bottleneck method, when clustering, only the relationship between out-of-domain documents and words (i.e., $I(C;Y)$, where C denotes C_{w_o} and Y denotes D_o) is taken into account. In order to integrate more in-domain knowledge to implement the task of out-of-domain sentiment lexicon construction, in the rest of this section, we will adapt the traditional IB algorithm to fit our task.

In order to fuse the three kinds of relationship mentioned in section 3.1, we use $I(W_o; D_o)$ to measure WD_{intra} -Relationship, use $I(W_o; W_i)$ to measure WW_{inter} -Relationship, and use $I(W_o; D_i)$ measure WD_{inter} -Relationship. Consequently, we adapt the loss function of the traditional IB algorithm by the following form:

$$I(D_o; W_o) - I(\hat{D}_o; \hat{W}_o) + \alpha \cdot \left[\left(I(D_i; W_o) - I(D_i; \hat{W}_o) \right) + \left(I(W_i; W_o) - I(W_i; \hat{W}_o) \right) \right] \quad (7)$$

where the trade-off parameter α is non-negative that represents the impact of in-domain knowledge on clustering.

The traditional IB algorithm is a clustering approach, since it can group out-of-domain terms together in an elegant way. However, it is insufficient because it only uses WD_{intra} -Relationship to identify the polarity of terms. By our extension, i.e., through usage of in-domain knowledge, we can utilize the polarity of in-domain data to identify the polarity of out-of-domain terms.

For the sake of being easy to implement, we need to transform the loss function in Equation (7) into another form that is represented by KL-divergence. Before transforming the loss function, let us first define some probability mass functions.

Definition 1: Let $f(D_o, W_o)$ denote the joint probability distribution of D_o and W_o . That is

$$f(d_o, w_o) = p(d_o, w_o) \quad (8)$$

$\hat{f}(D_o, W_o)$ denotes the joint probability distribution of D_o and W_o under co-clustering (\hat{D}_o, \hat{W}_o) that

$$\begin{aligned} \hat{f}(d_o, w_o) &= p(\hat{d}_o, \hat{w}_o) p(d_o | \hat{d}_o) p(w_o | \hat{w}_o) \\ &= p(\hat{d}_o, \hat{w}_o) \frac{p(d_o)}{p(\hat{d}_o)} \frac{p(w_o)}{p(\hat{w}_o)} \end{aligned} \quad (9)$$

where $d_o \in \hat{d}_o$ and $w_o \in \hat{w}_o$, where \hat{d}_o denotes an out-of-domain document cluster, and \hat{w}_o denotes an out-of-domain word cluster.

Definition 2: Let $g(D_i, W_o)$ denote the joint probability distribution of D_i and W_o . That is

$$g(d_i, w_o) = p(d_i, w_o) \quad (10)$$

$\hat{g}(D_i, W_o)$ denotes the joint probability distribution of D_i and W_o under the word clustering \hat{W}_o that

$$\begin{aligned} \hat{g}(d_i, w_o) &= p(d_i, \hat{w}_o) p(w_o | \hat{w}_o) \\ &= p(d_i, \hat{w}_o) \frac{p(w_o)}{p(\hat{w}_o)} \end{aligned} \quad (11)$$

We can also define $g(W_i, W_o)$ and $\hat{g}(W_i, W_o)$ in a similar way.

Theorem 1: for a fixed clustering, we can rewrite the loss function with the Kullback-Leibler divergence,

$$\begin{aligned} &I(D_o; W_o) - I(\hat{D}_o; \hat{W}_o) \\ &+ \alpha \cdot \left[\left(I(D_i; W_o) - I(D_i; \hat{W}_o) \right) + \left(I(W_i; W_o) - I(W_i; \hat{W}_o) \right) \right] \\ &= D_{\text{KL}} \left(f(D_o, W_o) \| \hat{f}(D_o, W_o) \right) \\ &+ \alpha \cdot \left[D_{\text{KL}} \left(g(D_i, W_o) \| \hat{g}(D_i, W_o) \right) + D_{\text{KL}} \left(g(W_i, W_o) \| \hat{g}(W_i, W_o) \right) \right] \end{aligned} \quad (12)$$

The detailed proof of Theorem 1 is given in the Appendix.

As a result, our iterative reinforcement approach is derived. This algorithm iteratively searches a clustering for the out-of-domain data (documents and words), and assigns sentiment polarity labels to the word clusters to complete the sentiment-lexicon building task.

As shown in Figure 2, in each update, the algorithm chooses the best cluster to minimize the loss function. After the iteration, we can get the out-of-domain word clusters with sentiment polarity label.

Input: A labeled in-domain document set D_i ; an unlabeled out-of-domain document set D_o ; a labeled in-domain word set W_i ; an unlabeled out-of-domain word set W_o ; initial clustering $(C_{D_o}^{(0)}, C_{W_o}^{(0)})$

Initialize the joint probability distribution f, \hat{f}, g and \hat{g} based on Equation (8), (9), (10), (11), respectively.

1. $t \leftarrow 1$

2. Repeat

a. compute the document cluster:

$$C_{D_o}^{(t)}(d) = \arg \min_{d_o} D_{KL}(f^{(t-1)}(d_o, W_o) \| f(\hat{d}_o, W_o))$$

b. update the probability distribution \hat{f} based on

$$C_{D_o}^{(t)}, C_{W_o}^{(t-1)} \text{ and Equation (9). } C_{W_o}^{(t)} = C_{W_o}^{(t-1)} \text{ and } \hat{g} = \hat{g}^{(t-1)}$$

c. compute the word cluster:

$$\begin{aligned} C_{W_o}^{(t+1)}(w_o) = \arg \min_{w_o} D_{KL} & \left(f^{(t)}(D_o, W_o) \| f(D_o, \hat{W}_o) \right) \\ & + \alpha \cdot D_{KL} \left(g^{(t)}(D_i, W_o) \| g(D_i, \hat{W}_o) \right) \\ & + \alpha \cdot D_{KL} \left(g^{(t)}(W_i, W_o) \| g(D_i, \hat{W}_o) \right) \end{aligned}$$

d. update the probability distribution \hat{g} based on $C_{W_o}^{(t+1)}$ and

$$\text{Equation (11). } \hat{f}^{(t+1)} = \hat{f}^{(t)} \text{ and } C_{D_o}^{(t+1)} = C_{D_o}^{(t)}$$

e. $t \leftarrow t + 2$

3. until $(C_{D_o}^{(t)} = C_{D_o}^{(t-1)})$ and $(C_{W_o}^{(t)} = C_{W_o}^{(t-1)})$

Output the partition functions $C_{D_o}^{(t)}$ and $C_{W_o}^{(t)}$

Figure 2: Pseudo-code of the adapted information bottleneck method

For more preciseness, in the following theorem, we will prove the convergence of proposed algorithm.

Theorem 2: Iterating over the equations given in Figure 2 converges to a stationary fixed point of the loss function (Equation 12).

Proof: The general idea of the proof is to show that updates defined by proposed algorithm can only reduce the loss function, and since the loss function is shown to be convex, we are guaranteed to converge to a (locally) optimized solution.

Lemma 2.1: $D(p\|q)$ is convex in the pair (p, q) ; that is, if (p_1, q_1) and (p_2, q_2) are two pairs of probability mass function, then

$$\begin{aligned} D(\lambda p_1 + (1-\lambda)p_2 \| \lambda q_1 + (1-\lambda)q_2) \\ \leq \lambda D(p_1 \| q_1) + (1-\lambda)D(p_2 \| q_2) \end{aligned} \quad (13)$$

Proof:

$$\begin{aligned} & D(\lambda p_1 + (1-\lambda)p_2 \| \lambda q_1 + (1-\lambda)q_2) \\ & = (\lambda p_1(x) + (1-\lambda)p_2(x)) \log \frac{\lambda p_1(x) + (1-\lambda)p_2(x)}{\lambda q_1(x) + (1-\lambda)q_2(x)} \\ & \leq \lambda p_1(x) \log \frac{\lambda p_1(x)}{\lambda q_1(x)} + (1-\lambda)p_2(x) \log \frac{(1-\lambda)p_2(x)}{(1-\lambda)q_2(x)} \quad (14) \\ & = \lambda D(p_1 \| q_1) + (1-\lambda)D(p_2 \| q_2) \end{aligned}$$

The loss function is a sum of Kullback-Leibler divergences, and in particular is non-negative. Moreover, from lemma 2.1, we verify that the Kullback-Leibler divergence is (strictly) convex with respect to each of its arguments (for more details, please refer to [4]). Since a sum of convex functions is also convex, and $\alpha \geq 0$, the function defined in Equation 12 is non-negative and convex.

Moreover, it is easy to see that in the process of iterating, the changes before and after all clustering are all non-negative. Hence, the iterating is equivalent to reducing the loss function.

Now we can conclude that through the iterative update, proposed algorithm converges to a (local) minimum. Note that, although the algorithm is able to minimize the loss function value in Equation 12, it is only able to find a locally minimal one. Finding the global optimal value is NP-hard.

4. EXPERIMENTAL SETUP

In order to evaluate the properties of the proposed algorithm, in this section, we describe our experiments and the data used in these experiments. Some researchers conducted sentiment classifier transferring research on English corpus, which are obtained from one web site, and are all product reviews. In order to highlight the domain-specific nature of sentiment expression, we collect reviews not only from different web sites, but also from domains with less similarity. Aimed at Chinese applications, we conduct the experiments based on the specialty of Chinese language, and verify the performance on Chinese web reviews. However, the main proposed approach in this paper is language independent in essence.

4.1 Data

We use three domain-specific datasets, i.e., Htl², Elec³, and Sto⁴. All of them are downloaded from the Internet, which including comments on hotel (from www.ctrip.com), electronics (from detail.zol.com.cn) and stock (from blog.sohu.com/stock), respectively. The detailed information is illustrated in Table 1.

Table 1: the detailed information of corpus

Domain	Positive	Negative	Total
Hotel	2000	2000	4000
Electronics	1054	554	1608
Stock	364	683	1047

We use ICTCLAS (<http://ictclas.org/>), a Chinese word segmentation software, to extract sentiment words from these

² <http://www.searchforum.org.cn/tansongbo/corpus/Htl-IV.rar>.

³ <http://www.searchforum.org.cn/tansongbo/corpus/Elec-IV.rar>.

⁴ <http://www.searchforum.org.cn/tansongbo/corpus/Sto-IV.rar>.

texts. In the usage of the part-of-speech tagging function provided by this software, we take all adjectives, adverbs and adjective-noun phrases as candidate sentiment words.

After removing the repeated words and words with ambiguity, we get a list of words in each domain. Then, we manually label the semantic orientation of every word, and use these labeled word lists as the sentiment lexicons in the following experiments.

In order to highlight the nature of domain-oriented sentiment lexicon, we distinguish the domain-dependent sentiment words from the domain-independent sentiment words in the process of labeling. We take the words only occur in one domain or the ones show reverse orientation among different domains as domain-dependent sentiment words; we take the words occur in more than one domain and behave with the same orientation as domain-independent sentiment words.

To justify the reliability of this labeling process, we ask three annotators to label one domain data, respectively. Three annotators had pair-wise agreement scores (Cohen’s Kappa score [3]) of 80.10%, 83.87% and 85.96%, which is high enough to be considered consistent. Table 2 presents the detailed information of labeled sentiment lexicon of each domain.

Table 2: the detailed information of labeled sentiment lexicon of each domain

Extracted Sentiment Words	Total (before pruning)	Non-Repeated			
		Pos		Neg	
		Independ	Depend	Independ	Depend
Hotel	93616	253	93	199	59
Electronics	58967	298	124	242	90
Stock	79560	343	89	567	112

4.2 Comparison Method

Since proposed method aims to construct domain-oriented sentiment lexicon, we should compare it with existing word semantic orientation inferring methods. Most of these approaches infer word semantic orientation by measuring the relationship between words, which can be either corpus-based [17][18] or knowledge-based [6][10]. Since the proposed approach is also corpus-based, for justness, we take the PMI method [18], improved PMI (SM+SO) method [7] and lexicon extension (LE) method [11] as the baseline methods, and compare the performance between these methods and our method.

The PMI method takes some labeled sentiment words as paradigm words to infer the semantic orientation of unlabelled words. In the implementation, we use the common part (sentiment words) of in-domain data and out-of-domain data as the paradigm words of the PMI method.

For SM+SO method, we set up the experimental environment as the default configurations as [7].

Since the LE method is an unsupervised method, we take the common part (sentiment words) between in-domain data and out-of-domain data as the origin lexicon, and set up the experimental environment as the default configurations as [11].

4.3 Evaluation Metrics

We use accuracy to evaluate the performance of proposed method. Let C be the clustering function which maps from word (or document) to its true sentiment label, and F be the function

which maps from word to its prediction sentiment label that given by the sentiment inferring methods. The accuracy is defined as:

$$Accuracy(w) = \frac{|\{w | w \in W_o \wedge C(w) = F(w)\}|}{|W_o|}$$

5. EXPERIMENTAL RESULTS AND DISCUSSION

5.1 Performance Comparison

Table 3 and

Table 4 report the performance comparison between proposed method and the three baselines on six tasks for domain-independent words and domain-dependent words.

Table 3: Accuracy of domain-independent sentiment word classification

	Baselines			Proposed Method
	PMI	SM+SO	LE	
Elec→Htl	76.6	77.5	80.7	88.1
Elec→Sto	69.7	68.3	71.3	73.6
Htl→Elec	74.1	76.7	83.4	79.7
Htl→Sto	85.4	88.0	86.7	84.8
Sto→Elec	70.5	73.3	81.3	76.7
Sto→Htl	67.9	71.2	81.8	84.8
Average	74.4	75.4	80.8	81.2

Table 4: Accuracy of domain-dependent sentiment word classification

	Baselines			Proposed Method
	PMI	SM+SO	LE	
Elec→Htl	68.4	73.5	73.2	87.5
Elec→Sto	57.8	60.6	63.1	73.2
Htl→Elec	72.1	75.4	76.3	75.9
Htl→Sto	73.7	76.4	78.1	82.2
Sto→Elec	70.6	73.3	73.4	74.1
Sto→Htl	68.8	71.2	73.6	82.8
Average	68.5	71.7	72.9	79.2

By the comparison between the two tables, we can find that nearly all approaches show better performance on domain-independent tasks than on domain-dependent tasks, which indicates the difficulty of domain-oriented sentiment lexicon construction.

From

Table 4, we can find that proposed method shows better performance on nearly all of the data sets. In consideration of that the baseline methods take only the relationship between out-of-domain words and in-domain words (WW_{inter} -Relationship) into account, while neglect the other two kinds of relationship (WD_{inter} -Relationship and WD_{intra} -Relationship), the full use of the three kinds of relationship may contribute to the performance of proposed method.

The experimental results show that the classifications on electronics and stock achieve worse performance than that of

hotel, because the two domains have fewer domain-independent words, which weakens the effect of WD_{intra} -Relationship.

Seen from these experimental results, a question may arise: why does the PMI method perform so poorly that it seems to disaccord the conclusion drawn by [18].

A reasonable explanation is that the PMI method is corpus-based, and the corpus size influences its performance very much. The experimental results provided by [18] is obtained by making use of search engine, and taking the whole Internet as the corpus. While the corpus in our experiment is relatively small, which may brings much noise and makes the co-occurrence information sparse. These factors may lead to the poor performance of the PMI method. From another perspective, this also shows the robustness of proposed method on relatively small-scaled corpus.

5.2 Convergence

Since proposed method is an iterative algorithm, it is an important issue to show the convergence property of proposed method. Theorem 2 has proven the convergence of our method theoretically. In this section, we will show the convergence of our method empirically. Figure 3 depicts the accuracy curves as functions for each iteration on six tasks for inferring sentiment for domain-dependent words, electronic-to-hotel, hotel-to-electronic and stock-to-hotel.

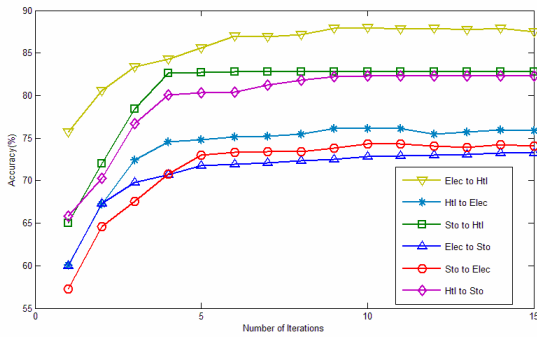


Figure 3: Word semantic orientation inferring accuracy curve after each iteration

From this figure, we can see that proposed method convergences very fast, when the iteration is over 5, all of the three curves tend to be stable. This shows the convergence of our method empirically. On this observation, we think it is sensible to consider that “10” is enough for our method to achieve a satisfactory solution.

5.3 Varying In-Domain Data Size

To investigate the robustness of our method, we conduct experimental tests on labeled in-domain data with different size. These labeled data are randomly chosen from in-domain data set (in this experiment, we take “stock→hotel” data as in-domain data) by different proportion. For comparison, we test the performance of the PMI method in the same experimental setup. Figure 4 presents this experimental result.

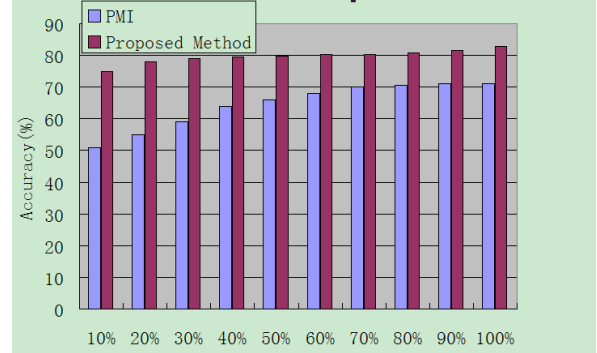


Figure 4: Accuracy of word semantic orientation inferring on different size of “stock-to-hotel” data set

From this figure, we can find that our method shows comparable performance even when there is only 10% of the in-domain data, while PMI gets worse quickly when the proportion of in-domain data decreases, especially when the proportion is less than 30%.

We think that in this experiment, the PMI method refers the semantic orientation of out-of-domain words relying on only the WW_{inter} -Relationship, which is provided by in-domain corpus. Therefore, the size-varying in in-domain corpus shows considerable influence on the performance of the PMI method; while proposed method also takes the WD_{inter} -Relationship and WD_{intra} -Relationship into account, which may counteract this adverse affect of the size-varying in in-domain corpus.

5.4 Varying the Parameter

There is only one parameter in proposed method, which is the trade-off parameter α in Equation 12. We conduct experimental tests by varying the parameter on the three data sets: electronics-to-hotel, hotel-to-electronics and stock-to-hotel. Figure 5 presents this experimental result.

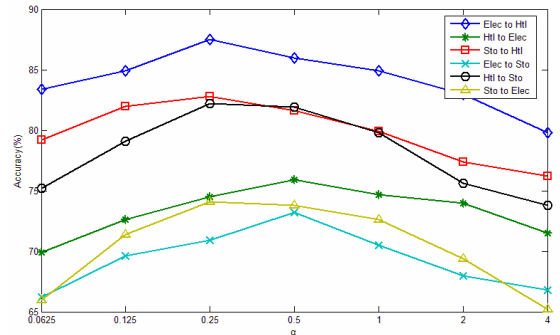


Figure 5: Accuracy curves of word semantic orientation inferring on different α

The parameter α reflects the influence of in-domain knowledge on the guide of clustering on out-of-domain data. From this figure, we can find that when α is small, by introducing in-domain knowledge, the accuracy increase; while when α is larger than a threshold, the algorithm gradually degenerates into the clustering based fully on in-domain knowledge, which excludes the contextual knowledge about out-of-domain data (i.e., WD_{intra} -Relationship). This will result in the

decline in accuracy. According to this figure, we set α to 0.25 in our experiments.

6. CONCLUSIONS AND FURTHER WORK

In this paper, we propose an adapted information bottleneck method for the automatic construction of domain-oriented sentiment lexicon by fusing the cross-domain knowledge (including word-to-document and word-to-word relationships) and within-domain knowledge (word-to-document relationship) in a unified information-theoretic framework, and solve this problem using an iterative reinforcement approach. Our theory verifies the convergence property of proposed method. The empirical results also support our theoretical analysis. In our experiment, it is shown that proposed method greatly outperforms the baseline methods in the task of building out-of-domain sentiment lexicon.

In this study, only the mutual information measure is employed to measure the three kinds of relationship. In order to show the robustness of the framework, our future effort is to investigate how to integrate more measures into this framework.

7. ACKNOWLEDGMENTS

This work was mainly supported by two funds, i.e., 60933005 and 60803085.

8. REFERENCES

- [1] A. Andreevskaia and S. Bergler. 2008. When Specialists and Generalists Work Together: Overcoming Domain Dependence in Sentiment Tagging. In Proceedings of ACL-08: HLT.
- [2] A. Aue and M. Gamon. 2005. Customizing Sentiment Classifiers to New Domains: a Case Study. In Proceedings of RANLP
- [3] J. Cohen. 1960. A coefficient of agreement for nominal scales. In: Educational and Psychological measurements 20, pp. 37-46
- [4] T. Cover and J. Thomas. 1991. Elements of Information Theory. John Wiley & Sons, New York.
- [5] A. Esuli and F. Sebastiani. 2005. Determining the semantic orientation of terms through gloss classification. In Proceedings of CIKM.
- [6] A. Esuli and F. Sebastiani. 2006. SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining. In Proceedings of LREC.
- [7] M. Gamon and A. Aue. 2005. Automatic identification of sentiment vocabulary exploiting low association with known sentiment terms. In Proceedings of ACL.
- [8] V. Hatzivassiloglou and K. McKeown. 1997. Predicting the semantic orientation of adjectives. In Proceedings of ACL.
- [9] M. Hu and B. Liu. 2004. Mining and summarizing customer reviews. In Proceedings of KDD.
- [10] J. Kamps, M. Marx, R. Mokken, and M. Rijke. 2004. Using WordNet to measure semantic orientation of adjectives. In Proceedings of LREC.
- [11] H. Kanayama, T. Nasukawa. 2006. Fully Automatic Lexicon Expansion for Domain-oriented Sentiment Analysis. In Proceedings of EMNLP.
- [12] S. Kim and E. Hovy. 2004. Determining the sentiment of opinions. In Proceedings of COLING.
- [13] B. Pang, L. Lee and S. Vaithyanathan. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. In Proceedings of EMNLP.
- [14] A. Popescu and O. Etzioni. 2005. Extracting product features and opinions from reviews. In Proceedings of HLT/EMNLP.
- [15] N. Slonim, N. Tishby. 1999. Agglomerative information bottleneck. In Proceedings of NIPS.
- [16] H. Takamura, T. Inui, M. Okumura. 2005. Extracting Semantic Orientations of Words using Spin Model. In Proceedings of ACL.
- [17] P. Turney. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In Proceedings of ACL.
- [18] P. Turney and M. Littman. 2003. Measuring Praise and Criticism: Inference of Semantic Orientation from Association. In: ACM Transactions on Information Systems, 21(4): 315-346.
- [19] J. Wiebe, T. Wilson and M. Bell. 2001. Identifying Collocations for Recognizing Opinions. In Proceedings of the ACL/EACL Workshop on Collocation.
- [20] H. Yu and V. Hatzivassiloglou. 2003. Towards Answering opinion Questions: Separating Facts from Opinions and Identifying the Polarity of Opinion Sentences. In Proceedings of EMNLP.
- [21] V. Stoyanov and C. Cardie. 2008. Topic Identification for Fine-Grained Opinion Analysis. In Proceedings of Coling.
- [22] H. Tang, S. Tan and X. Cheng. 2009. A Survey on Sentiment Detection of Reviews. Expert Systems with Applications.
- [23] S. Tan, G. Wu, H. Tang and X. Cheng. 2007. A novel scheme for domain-transfer problem in the context of sentiment analysis. In Proceedings of CIKM.
- [24] S. Tan, X. Cheng, Y. Wang, H. Xu. 2009. Adapting Naive Bayes to Domain Adaptation for Sentiment Analysis. In Proceedings of ECIR.
- [25] Q. Wu, S. Tan, H. Zhai, G. Zhang, M. Duan and X. Cheng. 2009. SentiRank: Cross-Domain Graph Ranking for Sentiment Classification. In Proceedings of WI.
- [26] W. Du, S. Tan. 2009. Building Domain-oriented Sentiment Lexicon by Improved Information Bottleneck. In Proceedings of CIKM.

9. APPENDIX

A. Proof of Theorem 1

Proof:

Lemma 1.1:

$$I(D_o; W_o) - I(\hat{D}_o; \hat{W}_o) = D_{KL} \left(f(D_o, W_o) \parallel \hat{f}(\hat{D}_o, \hat{W}_o) \right) \quad (15)$$

Proof:

$$\begin{aligned} & I(D_o; W_o) - I(\hat{D}_o; \hat{W}_o) \\ &= \sum_{\hat{d}_o \in \hat{D}_o} \sum_{\hat{w}_o \in \hat{W}_o} \sum_{d_o \in D_o} \sum_{w_o \in W_o} p(d_o, w_o) \log \frac{p(d_o, w_o)}{p(\hat{d}_o) p(\hat{w}_o)} \\ &\quad - \sum_{\hat{d}_o \in \hat{D}_o} \sum_{\hat{w}_o \in \hat{W}_o} \left(\sum_{d_o \in D_o} \sum_{w_o \in W_o} p(d_o, w_o) \right) \log \frac{p(\hat{d}_o, \hat{w}_o)}{p(\hat{d}_o) p(\hat{w}_o)} \\ &= \sum_{\hat{d}_o \in \hat{D}_o} \sum_{\hat{w}_o \in \hat{W}_o} \sum_{d_o \in D_o} \sum_{w_o \in W_o} p(d_o, w_o) \log \frac{p(d_o, w_o)}{p(\hat{d}_o, \hat{w}_o) \frac{p(d_o)}{p(\hat{d}_o)} \frac{p(w_o)}{p(\hat{w}_o)}} \\ &= \sum_{\hat{d}_o \in \hat{D}_o} \sum_{\hat{w}_o \in \hat{W}_o} \sum_{d_o \in D_o} \sum_{w_o \in W_o} f(d_o, w_o) \log \frac{f(d_o, w_o)}{\hat{f}(\hat{d}_o, \hat{w}_o)} \\ &= D_{KL} \left(f(D_o, W_o) \parallel \hat{f}(\hat{D}_o, \hat{W}_o) \right) \end{aligned} \quad (16)$$

Lemma 1.2:

$$I(D_i; W_o) - I(D_i; \hat{W}_o) = D_{KL} \left(g(D_i, W_o) \parallel \hat{g}(\hat{D}_i, \hat{W}_o) \right) \quad (17)$$

Proof:

$$\begin{aligned} & I(D_i; W_o) - I(D_i; \hat{W}_o) \\ &= \sum_{\hat{d}_i \in \hat{D}_i} \sum_{\hat{w}_o \in \hat{W}_o} \sum_{d_i \in D_i} \sum_{w_o \in W_o} p(d_i, w_o) \log \frac{p(d_i, w_o)}{p(\hat{d}_i) p(\hat{w}_o)} \\ &\quad - \alpha \sum_{\hat{d}_i \in \hat{D}_i} \sum_{\hat{w}_o \in \hat{W}_o} \left(\sum_{d_i \in D_i} \sum_{w_o \in W_o} p(d_i, w_o) \right) \frac{p(\hat{d}_i, \hat{w}_o)}{p(\hat{d}_i) p(\hat{w}_o)} \\ &= \sum_{\hat{d}_i \in \hat{D}_i} \sum_{\hat{w}_o \in \hat{W}_o} \sum_{d_i \in D_i} \sum_{w_o \in W_o} p(d_i, w_o) \log \frac{p(d_i, w_o)}{p(\hat{d}_i, \hat{w}_o) \frac{p(w_o)}{p(\hat{w}_o)}} \\ &= \sum_{\hat{d}_i \in \hat{D}_i} \sum_{\hat{w}_o \in \hat{W}_o} \sum_{d_i \in D_i} \sum_{w_o \in W_o} g(d_i, w_o) \log \frac{g(d_i, w_o)}{g(\hat{d}_i, \hat{w}_o)} \\ &= D_{KL} \left(g(D_i, W_o) \parallel \hat{g}(\hat{D}_i, \hat{W}_o) \right) \end{aligned} \quad (18)$$

Lemma 1.3:

$$I(W_i; W_o) - I(W_i; \hat{W}_o) = D_{KL} \left(g(W_i, W_o) \parallel \hat{g}(\hat{W}_i, \hat{W}_o) \right) \quad (19)$$

The proof of Lemma 1.3 is omitted, and it can be derived using the similar argument to Lemma 1.2.

Combining the above three lemmas, we can achieve the conclusion of Theorem 1.