

Structural Opinion Mining for Graph-based Sentiment Representation

Yuanbin Wu, Qi Zhang, Xuanjing Huang, Lide Wu
Fudan University
School of Computer Science
{ybwu,qz,xjhuang,ldwu}@fudan.edu.cn

Abstract

Based on analysis of on-line review corpus we observe that most sentences have complicated opinion structures and they cannot be well represented by existing methods, such as frame-based and feature-based ones. In this work, we propose a novel graph-based representation for sentence level sentiment. An integer linear programming-based structural learning method is then introduced to produce the graph representations of input sentences. Experimental evaluations on a manually labeled Chinese corpus demonstrate the effectiveness of the proposed approach.

1 Introduction

Sentiment analysis has received much attention in recent years. A number of automatic methods have been proposed to identify and extract opinions, emotions, and sentiments from text. Previous researches on sentiment analysis tackled the problem on various levels of granularity including document, sentence, phrase and word (Pang et al., 2002; Riloff et al., 2003; Dave et al., 2003; Takamura et al., 2005; Kim and Hovy, 2006; Somasundaran et al., 2008; Dasgupta and Ng, 2009; Hassan and Radev, 2010). They mainly focused on two directions: sentiment classification which detects the overall polarity of a text; sentiment related information extraction which tries to answer the questions like “*who* expresses *what* opinion on *which* target”.

Most of the current studies on the second direction assume that an opinion can be structured as a frame which is composed of a fixed number of slots. Typical slots include opinion holder, opinion expression, and evaluation target. Under this representa-

tion, they defined the task as a slots filling problem for each of the opinions. Named entity recognition and relation extraction techniques are usually applied in this task (Hu and Liu, 2004; Kobayashi et al., 2007; Wu et al., 2009).

However, through data analysis, we observe that 60.5% of sentences in our corpus do not follow the assumption used by them. A lot of important information about an opinion may be lost using those representation methods. Consider the following examples, which are extracted from real online reviews:

Example 1: *The interior is a bit noisy on the freeway*¹.

Example 2: *Takes good pictures during the daytime. Very poor picture quality at night*².

Based on the definition of opinion unit proposed by Hu and Liu (2004), from the first example, the information we can get is the author’s negative opinion about “interior” using an opinion expression “noisy”. However, the important restriction “on the freeway”, which narrows the scope of the opinion, is ignored. In fact, the tuple (“noisy”, “on the freeway”) cannot correctly express the original opinion: it is negative but under certain condition. The second example is similar. If the conditions “during the daytime” and “at night” are dropped, the extracted elements cannot correctly represent user’s opinions.

Example 3: *The camera is actually quite good for outdoors because of the software.*

Besides that, an opinion expression may induce other opinions which are not expressed directly. In example 3, the opinion expression is “good” whose

¹<http://reviews.carreview.com/blog/2010-ford-focus-review-the-compact-car-that-can/>

²<http://www.dooyoo.co.uk/digital-camera/sony-cyber-shot-dsc-s500/1151680/>

target is “camera”. But the “software” which triggers the opinion expression “good” is also endowed with a positive opinion. In practice, this induced opinion on “software” is actually more informative than its direct counterpart. Mining those opinions may help to form a complete sentiment analysis result.

Example 4: *The image quality is in the middle of its class, but it can still be a reasonable choice for students.*

Furthermore, the relations among individual opinions also provide additional information which is lost when they are considered separately. Example 4 is such a case that the whole positive comment of camera is expressed by a transition from a negative opinion to a positive one.

In order to address those issues, this paper describes a novel sentiment representation and analysis method. Our main contributions are as follows:

1. We investigate the use of graphs for representing sentence level sentiment. The vertices are evaluation target, opinion expression, modifiers of opinion. The Edges represent relations among them. The semantic relations among individual opinions are also included. Through the graph, various information on opinion expressions which is ignored by current representation methods can be well handled. And the proposed representation is language-independent.
2. We propose a supervised structural learning method which takes a sentence as input and the proposed sentiment representation for it as output. The inference algorithm is based on integer linear programming which helps to concisely and uniformly handle various properties of our sentiment representation. By setting appropriate prior substructure constraints of the graph, the whole algorithm achieves reasonable performances.

The remaining part of this paper is organized as follows: In Section 2 we discuss the proposed representation method. Section 3 describes the computational model used to construct it. Experimental results in test collections and analysis are shown in

Section 4. In Section 5, we present the related work and Section 6 concludes the paper.

2 Graph-based Sentiment Representation

In this work, we propose using directed graph to represent sentiments. In the graph, vertices are text spans in the sentences which are opinion expressions, evaluation targets, conditional clauses etc. Two types of edges are included in the graph: (1) relations among opinion expressions and their modifiers; (2) relations among opinion expressions. The edges of the first type exist within individual opinions. The second type of the edges captures the relations among individual opinions. The following sections detail the definition.

2.1 Individual Opinion Representation

Let r be an opinion expression in a sentence, the representation unit for r is a set of relations $\{(r, d_k)\}$. For each relation (r, d_k) , d_k is a *modifier* which is a span of text specifying the change of r 's meaning.

The relations between modifier and opinion expression can be the type of any kind. In this work, we mainly consider two basic types:

- *opinion restriction.* (r, d_k) is called an opinion restriction if d_k narrows r 's scope, adds a condition, or places limitations on r 's original meaning.
- *opinion expansion.* (r, d_k) is an opinion expansion if r 's scope expands to d_k , r induces another opinion on d_k , or the opinion on d_k is implicitly expressed by r .

Mining the opinion restrictions can help to get accurate meaning of an opinion, and the opinion expansions are useful to cover more indirect opinions. As with previous sentiment representations, we actually consider the third type of modifier which d_k is the evaluation target of r .

Figure 1 shows a concrete example. In this example, there are three opinion expressions: “good”, “sharp”, “slightly soft”. The modifiers of “good” are “indoors” and “Focus accuracy”, where relation (“good”, “indoors”) is an opinion restriction because “indoors” is the condition under which “Focus accuracy” is good. On the other hand, the relation

(“sharp”, “little 3x optical zooms”) is an opinion expansion because the “sharp” opinion on “shot” implies a positive opinion on “little 3x optical zooms”.

It is worth to remark that: 1) a modifier d_k can relate to more than one opinion expression. For example, multiple opinion expressions may share a same condition; 2) d_k itself can employ a set of relations, although the case appears occasionally. The following is an example:

Example 5: *The camera wisely get rid of many redundant buttons.*

In the example, “redundant buttons” is the evaluation target of opinion expression “wisely get rid of”, but itself is a relation between “redundant” and “buttons”. Such nested semantic structure is described by a path: “wisely get rid of”, \xrightarrow{target} [“redundant” \xrightarrow{target} “buttons”]_{nested target}.

2.2 Relations between Individual Opinion Representation

Assume $\langle r_i \rangle$ are opinion expressions ordered by their positions in sentence, and each of them has been represented by relations $\{(r_i, d_{ik})\}$ individually (the nested relations for d_{ik} have also been determined). Then we define two relations on adjacent pair r_i, r_{i+1} : *coordination* when the polarities of r_i and r_{i+1} are consistent, and *transition* when they are opposite. Those relations among r_i form a set B called *opinion thread*. In Figure 1, the opinion thread is: $\{(\text{“good”}, \text{“sharp”}), (\text{“sharp”}, \text{“slightly soft”})\}$.

The whole sentiment representation for a sentence can be organized by a direct graph $G = (V, E)$. Vertex set V includes all opinion expressions and modifiers. Edge set E collects both relations of each individual opinion and relations in opinion thread. The edges are labeled with relation types in label set $\mathcal{L} = \{\text{“restriction”}, \text{“expansion”}, \text{“target”}, \text{“coordination”}, \text{“transition”}\}$ ³.

Compared with previous works, the advantages of using G as sentiment representation are: 1) for individual opinions, the modifiers will collect more information than using opinion expression alone.

³We don’t define any “label” on vertices: if two span of text satisfy a relation in \mathcal{L} , they are chosen to be vertices and an edge with proper label will appear in E . In other words, vertices are identified by checking whether there exist relations among them.

Focus accuracy was good indoors, and although the little 3x optical zooms produced sharp shots, the edges were slightly soft on the Canon.

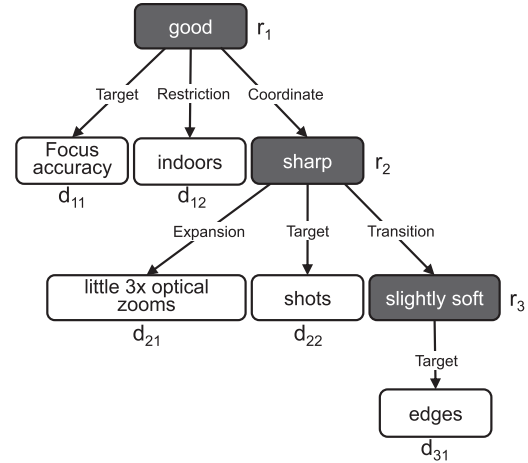


Figure 1: Sentiment representation for an example sentence

Thus G is a relatively complete and accurate representation; 2) the opinion thread can help to catch global sentiment information, for example the general polarity of a sentence, which is dropped when the opinions are separately represented.

3 System Description

To produce the representation graph G for a sentence, we need to extract candidate vertices and build the relations among them to get a graph structure. For the first task, the experimental results in Section 4 demonstrate that the standard sequential labeling method with simple features can achieve reasonable performance. In this section, we focus on the second task, and assume the vertices in the graph have already been correctly collected in the following formulation of algorithm.

3.1 Preliminaries

In order to construct graph G , we use a structural learning method. The framework is from the first order discriminative dependency parsing model (McDonald and Pereira, 2005). A sentence is denoted by s ; \mathbf{x} are text spans which will be vertices of graph; x_i is the i th vertex in \mathbf{x} ordered by their positions in s . For a set of vertices \mathbf{x} , \mathbf{y} is the graph of its sentiment representation, and $e = (x_i, x_j) \in \mathbf{y}$ is the direct edge from x_i to x_j in \mathbf{y} . In addition, x_0 is a

virtual root node without inedge. $\mathcal{G} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_n^N$ is training set.

Following the edge based factorization, the score of a graph is the sum of its edges' scores,

$$\begin{aligned} \text{score}(\mathbf{x}, \mathbf{y}) &= \sum_{(x_i, x_j) \in \mathbf{y}} \text{score}(x_i, x_j) \\ &= \sum_{(x_i, x_j) \in \mathbf{y}} \alpha^T \mathbf{f}(x_i, x_j), \end{aligned} \quad (1)$$

$\mathbf{f}(x_i, x_j)$ is a high dimensional feature vector of the edge (x_i, x_j) . The components of \mathbf{f} are either 0 or 1. For example the k-th component could be

$$\mathbf{f}_k(x_i, x_j) = \begin{cases} 1 & \text{if } x_i.\text{POS} = \text{JJ and } x_j.\text{POS} = \text{NN} \\ & \text{and label of } (x_i, x_j) \text{ is restriction.} \\ 0 & \text{otherwise} \end{cases}$$

Then the score of an edge is the linear combination of \mathbf{f} 's components, and the coefficients are in vector α .

Algorithm 1 shows the parameter learning process. It aims to get parameter α which will assign the correct graph \mathbf{y} with the highest score among all possible graphs of \mathbf{x} (denoted by \mathcal{Y}).

Algorithm 1 Online structural learning

Training Set: $\mathcal{G} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_n^N$

```

1:  $\alpha^0 = 0, r = 0, T = \text{maximum iteration}$ 
2: for  $t = 0$  to  $T$  do
3:   for  $n = 0$  to  $N$  do
4:      $\hat{\mathbf{y}} = \arg \max_{\mathbf{y} \in \mathcal{Y}} \text{score}(\mathbf{x}_n, \mathbf{y}) \triangleright$  Inference
5:     if  $\hat{\mathbf{y}} \neq \mathbf{y}_n$  then
6:       update  $\alpha^t$  to  $\alpha^{t+1} \triangleright$  PA
7:        $r = r + \alpha^{t+1}$ 
8:     end if
9:   end for
10: end for
11: return  $\alpha = r / (N * T)$ 

```

3.2 Inference

Like other structural learning tasks, the ‘‘arg max’’ operation in the algorithm (also called inference)

$$\begin{aligned} \hat{\mathbf{y}} &= \arg \max_{\mathbf{y} \in \mathcal{Y}} \text{score}(\mathbf{x}, \mathbf{y}) \\ &= \arg \max_{\mathbf{y} \in \mathcal{Y}} \sum_{(x_i, x_j) \in \mathbf{y}} \alpha^T \mathbf{f}(x_i, x_j) \end{aligned} \quad (2)$$

is hard because all possible values of \mathbf{y} form a huge search space. In our case, \mathcal{Y} is all possible directed acyclic graphs of the given vertex set, which number is exponential. Directly solving the problem of finding maximum weighted acyclic graph is equivalent to finding maximum feedback arc set, which is a NP-hard problem (Karp, 1972). We will use integer linear programming (ILP) as the framework for this inference problem.

3.2.1 Graph Properties

We first show some properties of graph G either from the definition of relations or corpus statistics.

Property 1. The graph is connected and without directed cycle. From individual opinion representation, each subgraph of G which takes an opinion expression as root is connected and acyclic. Thus the connectedness is guaranteed for opinion expressions are connected in opinion thread; the acyclic is guaranteed by the fact that if a modifier is shared by different opinion expressions, the inedges from them always keep (directed) acyclic.

Property 2. Each vertex can have one outedge labeled with coordination or transition at most. The opinion thread B is a directed path in graph.

Property 3. The graph is sparse. The average in-degree of a vertex is 1.03 in our corpus, thus the graph is almost a rooted tree. In other words, the cases that a modifier connects to more than one opinion expression rarely occur comparing with those vertices which have a single parent. An explanation for this sparseness is that opinions in online reviews always concentrate in local context and have local semantic connections.

3.2.2 ILP Formulation

Based on the property 3, we divide the inference algorithm into two steps: i) constructing G 's spanning tree (arborescence) with property 1 and 2; ii) finding additional non-tree edges as a post processing task. The first step is close to the works on ILP formulations of dependency parsing (Riedel and Clarke, 2006; Martins et al., 2009). In the second step, we use a heuristic method which greedily adds non-tree edges. A similar approximation method is also used in (McDonald and Pereira, 2006) for acyclic dependency graphs.

Step 1. Find MST. Following the multicommodity

flow formulation of maximum spanning tree (MST) problem in (Magnanti and Wolsey, 1994), the ILP for MST is:

$$\max. \sum_{i,j} y_{ij} \cdot \text{score}(x_i, x_j) \quad (3)$$

$$\text{s.t.} \quad \sum_{i,j} y_{ij} = |V| - 1 \quad (4)$$

$$\sum_i f_{ij}^u - \sum_k f_{jk}^u = \delta_j^u, 1 \leq u, j \leq |V| \quad (5)$$

$$\sum_k f_{0k}^u = 1, \quad 1 \leq u \leq |V| \quad (6)$$

$$f_{ij}^u \leq y_{ij}, \quad 1 \leq u, j \leq |V|, \\ 0 \leq i \leq |V| \quad (7)$$

$$f_{ij}^u \geq 0, \quad 1 \leq u, j \leq |V|, \\ 0 \leq i \leq |V| \quad (8)$$

$$y_{ij} \in \{0, 1\}, \quad 0 \leq i, j \leq |V|. \quad (9)$$

In this formulation, y_{ij} is an edge indicator variable that (x_i, x_j) is a spanning tree edge when $y_{ij} = 1$, (x_i, x_j) is a non-tree edge when $y_{ij} = 0$. Then output \mathbf{y} is represented by the set $\{y_{ij}, 0 \leq i, j \leq |V|\}$ ⁴. Eq(4) ensures that there will be exactly $|V| - 1$ edges are chosen. Thus if the edges corresponding to those non zero y_{ij} is a connected subgraph, \mathbf{y} is a well-formed spanning tree. Objective function just says the optimal solution of y_{ij} have the maximum weight.

The connectedness is guaranteed if for every vertex, there is exactly one path from root to it. It is formulated by using $|V| - 1$ flows $\{f^u, 1 \leq u \leq |V|\}$. f^u starts from virtual root x_0 towards vertex x_u . Each flow $f^u = \{f_{ij}^u, 0 \leq i, j \leq |V|\}$. f_{ij}^u indicates whether flow f^u is through edge (x_i, x_j) . so it should be 0 if edge (x_i, x_j) does not exist (by (7)). The Kronecker's delta δ_j^u in (5) guarantees f^u is only assumed by vertex x_u , so f^u is a well-formed path from root to x_u . (6) ensures there is only one flow (path) from root to x_u . Thus the subgraph is connected. The following are our constraints:

c1: Constraint on edges in opinion thread (10)-(11).

From the definition of opinion thread, we impose a constraint on every vertex's outedges in opinion thread, which are labeled with "coordination" or

⁴For simplicity, we overload symbol \mathbf{y} from the graph of the sentiment representation to the MST of it.

"transition". Let \mathbb{I}_{ob} be a characteristic function on edges: $\mathbb{I}_{ob}((j, k)) = 1$ when edge (x_j, x_k) is labeled with "coordination" or "transition", otherwise 0. We denote q variables for vertices:

$$q_j = \sum_k y_{jk} \cdot \mathbb{I}_{ob}((j, k)), \quad 0 \leq j \leq |V|. \quad (10)$$

Then following linear inequalities bound the number of outedges in opinion thread (≤ 1) on each vertex:

$$q_j \leq 1, \quad 0 \leq j \leq |V|. \quad (11)$$

c2: Constraint on target edge (12).

We also bound the number of evaluation targets for a vertex in a similar way. Let \mathbb{I}_t be characteristic function on edges identifying whether it is labeled with "target",

$$\sum_k y_{jk} \cdot \mathbb{I}_t((j, k)) \leq C_t, \quad 0 \leq j \leq |V|. \quad (12)$$

The parameter C_t can be adjusted according to the style of document. In online reviews, authors tend to use simple and short comments on individual targets, so C_t could be set small.

c3: Constraint on opinion thread (13)-(18).

From graph property 2, the opinion thread should be a directed path. It implies the number of connected components whose edges are "coordination" or "transition" should be less than 1. Two set of additional variables are needed: $\{c_j, 0 \leq j \leq |V|\}$ and $\{h_j, 0 \leq j \leq |V|\}$, where

$$c_j = \begin{cases} 1 & \text{if an opinion thread starts at } x_j \\ 0 & \text{otherwise} \end{cases},$$

and

$$h_j = \sum_i y_{ij} \cdot \mathbb{I}_{ob}((i, j)). \quad (13)$$

Then $c_j = \neg h_j \wedge q_j$, which can be linearized by

$$c_j \geq q_j - h_j, \quad (14)$$

$$c_j \leq 1 - h_j, \quad (15)$$

$$c_j \leq q_j, \quad (16)$$

$$c_j \geq 0. \quad (17)$$

If the sum of c_j is no more than 1, the opinion thread of graph is a directed path.

$$\sum_j c_j \leq 1. \quad (18)$$

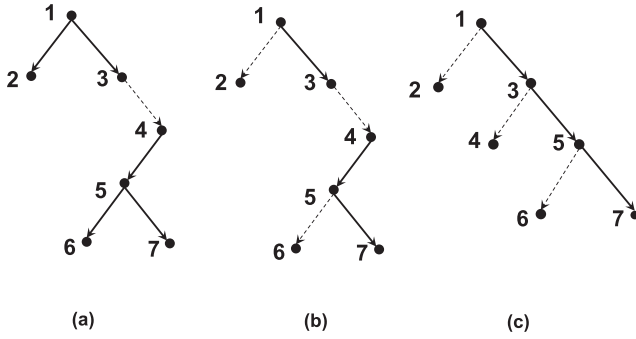


Figure 2: The effects of **c1** and **c3**. Assume solid lines are edges labeled with “coordination” and “transition”, dot lines are edges labeled with other types. (a) is an arbitrary tree. (b) is a tree with **c1** constraints. (c) is a tree with **c1** and **c3**. It shows **c1** are not sufficient for graph property 2: the edges in opinion thread may not be connected.

Figure 2 illustrates the effects of **c1** and **c3**.

Equations (10)-(18), together with basic multi-commodity flow model build up the inference algorithm. The entire ILP formulation involves $O(|V|^3)$ variables and $O(|V|^2)$ constraints. Generally, ILP falls into NPC, but as an important result, in the multicommodity flow formulation of maximum spanning tree problem, the integer constraints (9) on y_{ij} can be dropped. So the problem reduces to a linear programming which is polynomial solvable (Magnanti and Wolsey, 1994). Unfortunately, with our additional constraints the LP relaxation is not valid.

Step 2. Adding non-tree edges. We examine the case that a modifier attaches to different opinion expressions. That often occurs as the result of the sharing of modifiers among adjacent opinion expressions. We add those edges in the following heuristic way: If a vertex r_i in opinion thread does not have any modifier, we search the modifiers of its adjacent vertices r_{i+1} , r_{i-1} in the opinion thread, and add edge (r_i, d^*) where

$$d^* = \arg \max_{d \in S} \text{score}(r_i, d),$$

and S are the modifiers of r_{i-1} and r_{i+1} .

3.3 Training

We use online passive aggressive algorithm (PA) with Hamming cost of two graphs in training (Cramer et al., 2006).

Unigram Feature Template	
Inside Features	$x_i.\text{text}$ $w_0.\text{text}$ $w_1.\text{text}$
	$w_0.\text{POS}$ $w_1.\text{POS}$
	$w_{k-1}.\text{text}$ $w_k.\text{text}$
	$w_{k-1}.\text{POS}$ $w_k.\text{POS}$
	$x_i.\text{hasDigital}$
	$x_i.\text{isSingleWord}$
	$x_i.\text{hasSentimentWord}$
	$x_i.\text{hasParallelPhrase}$
Outside Features	$w_{-1}.\text{text}$ $w_{-2}.\text{text}$
	$w_{-1}.\text{POS}$ $w_{-2}.\text{POS}$
	$w_{k+1}.\text{text}$ $w_{k+2}.\text{text}$
	$w_{k+1}.\text{POS}$ $w_{k+2}.\text{POS}$
	$c_{-1}.\text{text}$ $c_{-2}.\text{text}$
	$c_{-1}.\text{POS}$ $c_{-2}.\text{POS}$
	$c_{l+1}.\text{text}$ $c_{l+2}.\text{text}$
	$c_{l+1}.\text{POS}$ $c_{l+2}.\text{POS}$
Other Features	
distance between parent and child	
dependency parsing relations	

Table 1: Feature set

3.4 Feature Construction

For each vertex x_i in graph, we use 2 sets of features: inside features which are extracted inside the text span of x_i ; outside features which are outside the text span of x_i . A vertex x_i is described both in word sequence (w_0, w_1, \dots, w_k) and character sequence (c_0, c_1, \dots, c_l) , for the sentences are in Chinese.

$$\begin{aligned} & \dots, w_{-1}, \underbrace{w_0, w_1, w_2, \dots, w_{k-1}, w_k, w_{k+1}}_{x_i} \dots \\ & \dots, c_{-1}, \underbrace{c_0, c_1, c_2, \dots, c_{l-1}, c_l, c_{l+1}}_{x_i} \dots \end{aligned}$$

For an edge (x_i, x_j) , the high dimensional feature vector $\mathbf{f}(x_i, x_j)$ is generated by using unigram features in Table 1 on x_i and x_j respectively. The distance between parent and child in sentence is also attached in features. In order to involve syntactic information, whether there is certain type of dependency relation between x_i and x_j is also used as a feature.

4 Experiments

4.1 Corpus

We constructed a Chinese online review corpus from Pcpop.com, Zol.com.cn, and It168.com, which have a large number of reviews about digital camera. The corpus contains 138 documents and 1735 sentences. Since some sentences do not contain any opinion, 1390 subjective sentences were finally chosen and manually labeled.

Two annotators labeled the corpus independently. The annotators started from locating opinion expressions, and for each of them, they annotated other modifiers related to it. In order to keep the reliability of annotations, another annotator was asked to check the corpus and determine the conflicts. Finally, we extracted 6103 elements, which are connected by 6284 relations.

Relation	Number
Target	2479
Coordinate	1173
Transition	154
Restriction	693
Expansion	386

Table 2: Statistics of relation types

Table 2 shows the number of various relation types appearing in the labeled corpus. We observe 60.5% of sentences and 32.1% of opinion expressions contain other modifiers besides “target”. Thus only mining the relations between opinion expressions and evaluation target is actually at risk of inaccurate and incomplete results.

4.2 Experiments Configurations

In all the experiments below, we take 90% of the corpus as training set, 10% as test set and run 10 folder cross validation. In feature construction, we use an external Chinese sentiment lexicon which contains 4566 positive opinion words and 4370 negative opinion words. For Chinese word segment, we use ctbparser⁵. Stanford parser (Klein and Manning, 2003) is used for dependency parsing. In the settings of PA, the maximum iteration number is

⁵<http://code.google.com/p/ctbparser/>

set to 2, which is chosen by maximizing the testing performances, aggressiveness parameter C is set to 0.00001. For parameters in inference algorithm, $C_t = 2$, the solver of ILP is Ipsolve⁶.

We evaluate the system from the following aspects: 1) whether the structural information helps to mining opinion relations. 2) How the proposed inference algorithm performs with different constraints. 3) How the various features affect the system. Except for the last one, the feature set used for different experiments are the same (“In+Out+Dep” in Table 5). The criteria for evaluation are similar to the unlabeled attachment score in parser evaluations, but due to the equation $|E| = |V| - 1$ is not valid if G is not a tree, we evaluate precision $P = \frac{\#true\ edges\ in\ result\ graph}{\#edges\ in\ result\ graph}$, recall $R = \frac{\#true\ edges\ in\ result\ graph}{\#edges\ in\ true\ graph}$, and F-score $F = \frac{2P \cdot R}{P + R}$.

4.3 Results

1. The effects of structural information. An alternative method to extract relations is directly using a classifier to judge whether there is a relation between any two elements. Those kinds of methods were used in previous opinion mining works (Wu et al., 2009; Kobayashi et al., 2007). To show the entire structural information is important for mining relations, we use SVM for binary classification on candidate pairs. The data point representing a pair (x_i, x_j) is the same as the high dimensional feature vectors $\mathbf{f}(x_i, x_j)$. The setting of our algorithm “MST+c1+c2+c3” is the basic MST with all the constraints. The results are shown in the Table 3.

	P	R	F
SVM	64.9	24.0	35.0
MST+c1+c2+c3-m	61.5	74.0	67.2
MST+c1+c2+c3	73.1	71.0	72.1

Table 3: Binary classifier and structural learning

From the results, the performance of SVM (especially recall) is relatively poor. A possible reason is that the huge imbalance of positive and negative training samples (only $\Theta(n)$ positive pairs among all n^2 pairs). And the absence of global structural

⁶<http://sourceforge.net/projects/lpsolve/>

knowledge makes binary classifier unable to use the information provided by classification results of other pairs.

In order to examine whether the complicated sentiment representation would disturb the classifier in finding relations between opinion expressions and its target, we evaluate the system by discarding the modifiers of opinion restriction and expansion from the corpus. The result is shown in the second row of Table 3. We observe that “MST+c1+c2+c3” is still better which means at least on overall performance the additional modifiers do not harm.

2. *The effect of constraints on inference algorithm.* In the inference algorithm, we utilized the properties of graph G and adapted the basic multicommodity flow ILP to our specific task. To evaluate how the constraints affect the system, we decompose the algorithm and combine them in different ways.

	P	R	F
MST	69.3	67.3	68.3
MST+c1	70.0	68.0	69.0
MST+c2	69.8	67.8	68.8
MST+c1+c2	70.6	68.6	69.6
MST+c1+c3	72.4	70.4	71.4
MST+c1+c2+c3	73.1	71.0	72.1
MST+c1+c2+c3+g	72.5	72.3	72.4

Table 4: Results on inference methods. “MST” is the basic multicommodity flow formulation of maximum spanning tree; c1, c2, c3 are groups of constraint from Section 3.2.2; “g” is our heuristic method for additional non spanning tree edges.

From Table 4, we observe that with any additional constraints the inference algorithm outperforms the basic maximum spanning tree method. It implies although we did not use high order model (e.g. involving grandparent and sibling features), prior structural constraints can also help to get a better output graph. By comparing with different constraint combinations, the constraints on opinion thread (c1, c3) are more effective than constraints on evaluation targets (c2). It is because opinion expressions are more important in the entire sentiment representation. The main structure of a graph is clear once the relations between opinion expressions are correctly determined.

3. *The effects of various features.* We evaluate the

performances of different feature configurations in Table 5. From the results, the outside feature set is more effective than inside feature set, even if it does not use any external resource. A possible reason is that the content of a vertex can be very complicated (a vertex even can be a clause), but the features surrounding the vertex are relatively simple and easy to identify (for example, a single preposition can identify a complex condition). The dependency feature has limited effect, due to that lots of online review sentences are ungrammatical and parsing results are unreliable. And the complexity of vertices also messes the dependency feature.

	P	R	F
In-s	66.3	66.3	66.3
In	66.7	66.4	66.6
Out	67.8	67.4	67.6
In+Out	72.0	70.5	71.0
In+Out+Dep	72.5	72.3	72.4

Table 5: Results with different features. “In” represents the result of inside feature set; “In-s” is “In” without the external opinion lexicon feature; “Out” uses the outside feature set; “In+Out” uses both “In” and “Out”, “In+Out+Dep” adds the dependency feature. The inference algorithm is “MST+c1+c2+c3+g” in Table 4.

We analyze the errors in test results. A main source of errors is the confusion of classifier between “target” relations and “coordination”, “transition” relations. The reason may be that for a modification on opinion expression (r, d_k) , we allow d_k recursively has its own modifiers (Example 5). Thus an opinion expression can be a modifier which brings difficulties to classifier.

4. *Extraction of vertices.* Finally we conduct an experiment on vertex extraction using standard sequential labeling method. The tag set is simply {B, I, O} which are signs of begin, inside, outside of a vertex. The underlying model is conditional random field⁷. Feature templates involved are in Table 6. We only use basic features in the experiment. 10 folder cross validation results are in table 7. We suspect that the performances (especially recall) could be improved if some external resources(i.e. ontology, domain related lexicon, etc.) are involved.

⁷We use CRF++ toolkit, <http://crfpp.sourceforge.net/>

Unigram Template	
$c_i.char$	character
$c_i.isDigit$	digit
$c_i.isAlpha$	english letter
$c_i.isPunc$	punctuation
$c_i.inDict$	in a sentiment word
$c_i.BWord$	start of a word
$c_i.EWord$	end of a word

Table 6: Features for vertex extraction. The sequential labeling is conducted on character level (c_i). The sentiment lexicon used in $c_i.inDict$ is the same as Table 1. We also use bigram feature templates on $c_i.char$, $c_i.isAlpha$, $c_i.inDict$ with respect to c_{i-1} and c_{i+1} .

	P	R	F
E+Unigram	56.8	45.1	50.3
E+Unigram+Bigram	57.3	47.9	52.1
O+Unigram	71.9	57.2	63.7
O+Unigram+Bigram	72.3	60.2	65.6

Table 7: Results on vertices extraction with 10 folder cross validation. We use two criterion: 1) the vertex is correct if it is exactly same as ground truth(“E”), 2) the vertex is correct if it overlaps with ground truth(“O”).

5 Related Work

Opinion mining has recently received considerable attentions. Large amount of work has been done on sentimental classification in different levels and sentiment related information extraction. Researches on different types of sentences such as comparative sentences (Jindal and Liu, 2006) and conditional sentences (Narayanan et al., 2009) have also been proposed.

Kobayashi et al. (2007) presented their work on extracting opinion units including: opinion holder, subject, aspect and evaluation. They used slots to represent evaluations, converted the task to two kinds of relation extraction tasks and proposed a machine learning-based method which used both contextual and statistical clues.

Jindal and Liu (2006) studied the problem of identifying comparative sentences. They analyzed different types of comparative sentences and proposed learning approaches to identify them.

Sentiment analysis of conditional sentences were studied by Narayanan et al. (2009). They aimed

to determine whether opinions expressed on different topics in a conditional sentence are positive, negative or neutral. They analyzed the conditional sentences in both linguistic and computational perspectives and used learning method to do it. They followed the *feature-based sentiment analysis* model (Hu and Liu, 2004), which also use flat frames to represent evaluations.

Integer linear programming was used in many NLP tasks (Denis and Baldrige, 2007), for its power in both expressing and approximating various inference problems, especially in parsing (Riedel and Clarke, 2006; Martins et al., 2009). Martins et al. (2009) also applied ILP with flow formulation for maximum spanning tree, besides, they also handled dependency parse trees involving high order features(sibling, grandparent), and with projective constraint.

6 Conclusions

This paper introduces a representation method for opinions in online reviews. Inspections on corpus show that the information ignored in previous sentiment representation can cause incorrect or incomplete mining results. We consider opinion restriction, opinion expansions, relations between opinion expressions, and represent them with a directed graph. Structural learning method is used to produce the graph for a sentence. An inference algorithm is proposed based on the properties of the graph. Experimental evaluations with a manually labeled corpus are given to show the importance of structural information and effectiveness of proposed inference algorithm.

7 Acknowledgement

The author wishes to thank the anonymous reviewers for their helpful comments. This work was partially funded by 973 Program (2010CB327906), National Natural Science Foundation of China (61003092, 61073069), 863 Program of China (2009AA01A346), Shanghai Science and Technology Development Funds(10dz1500104), Doctoral Fund of Ministry of Education of China (200802460066), Shanghai Leading Academic Discipline Project (B114), and Key Projects in the National Science & Technology Pillar Pro-

gram(2009BAH40B04).

References

- Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585.
- Sajib Dasgupta and Vincent Ng. 2009. Mine the easy, classify the hard: A semi-supervised approach to automatic sentiment classification. In *Proceedings of ACL-IJCNLP*.
- Kushal Dave, Steve Lawrence, and David M. Pennock. 2003. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *Proceedings of WWW*.
- Pascal Denis and Jason Baldridge. 2007. Joint determination of anaphoricity and coreference resolution using integer programming. In *Proceedings of NAACL-HLT*.
- Ahmed Hassan and Dragomir R. Radev. 2010. Identifying text polarity using random walks. In *Proceedings of ACL*, pages 395–403, Uppsala, Sweden, July. Association for Computational Linguistics.
- Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of SIGKDD*.
- Nitin Jindal and Bing Liu. 2006. Identifying comparative sentences in text documents. In *Proceedings of SIGIR*.
- R. Karp. 1972. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press.
- Soo-Min Kim and Eduard Hovy. 2006. Automatic identification of pro and con reasons in online reviews. In *Proceedings of the COLING-ACL*.
- Dan Klein and Christopher D. Manning. 2003. Fast exact inference with a factored model for natural language parsing. In *In Advances in Neural Information Processing Systems 15 (NIPS)*, pages 3–10. MIT Press.
- Nozomi Kobayashi, Kentaro Inui, and Yuji Matsumoto. 2007. Extracting aspect-evaluation and aspect-of relations in opinion mining. In *Proceedings of EMNLP-CoNLL*.
- Thomas L. Magnanti and Laurence A. Wolsey. 1994. Optimal trees.
- Andre Martins, Noah Smith, and Eric Xing. 2009. Concise integer linear programming formulations for dependency parsing. In *Proceedings of ACL-IJCNLP*.
- R. Mcdonald and F. Pereira. 2005. Identifying gene and protein mentions in text using conditional random fields. *BMC Bioinformatics*.
- Ryan Mcdonald and Fernando Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proc. of EACL*, pages 81–88.
- Ramanathan Narayanan, Bing Liu, and Alok Choudhary. 2009. Sentiment analysis of conditional sentences. In *Proceedings of EMNLP*.
- Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proc. of EMNLP 2002*.
- Sebastian Riedel and James Clarke. 2006. Incremental integer linear programming for non-projective dependency parsing. In *Proceedings of EMNLP*.
- Ellen Riloff, Janyce Wiebe, and Theresa Wilson. 2003. Learning subjective nouns using extraction pattern bootstrapping. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL*.
- Swapna Somasundaran, Janyce Wiebe, and Josef Ruppenhofer. 2008. Discourse level opinion interpretation. In *Proceedings of COLING*.
- Hiroya Takamura, Takashi Inui, and Manabu Okumura. 2005. Extracting semantic orientations of words using spin model. In *Proceedings of ACL*.
- Yuanbin Wu, Qi Zhang, Xuangjing Huang, and Lide Wu. 2009. Phrase dependency parsing for opinion mining. In *Proceedings of EMNLP*.