Mining Opinion Words and Opinion Targets in a Two-Stage Framework

Liheng Xu, Kang Liu, Siwei Lai, Yubo Chen and Jun Zhao
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China
{lhxu, kliu, swlai, ybchen, jzhao}@nlpr.ia.ac.cn

Abstract

This paper proposes a novel two-stage method for mining opinion words and opinion targets. In the first stage, we propose a *Sentiment Graph Walking* algorithm, which naturally incorporates syntactic patterns in a Sentiment Graph to extract opinion word/target candidates. Then random walking is employed to estimate confidence of candidates, which improves extraction accuracy by considering confidence of patterns. In the second stage, we adopt a self-learning strategy to refine the results from the first stage, especially for filtering out high-frequency noise terms and capturing the long-tail terms, which are not investigated by previous methods. The experimental results on three real world datasets demonstrate the effectiveness of our approach compared with state-of-the-art unsupervised methods.

1 Introduction

Opinion mining not only assists users to make informed purchase decisions, but also helps business organizations understand and act upon customer feedbacks on their products or services in real-time. Extracting opinion words and opinion targets are two key tasks in opinion mining. Opinion words refer to those terms indicating positive or negative sentiment. Opinion targets represent aspects or attributes of objects toward which opinions are expressed. Mining these terms from reviews of a specific domain allows a more thorough understanding of customers' opinions.

Opinion words and opinion targets often cooccur in reviews and there exist modified relations (called *opinion relation* in this paper) between them. For example, in the sentence "It has a clear screen", "clear" is an opinion word and "screen" is an opinion target, and there is an opinion relation between the two words. It is natural to identify such opinion relations through common syntactic patterns (also called *opinion patterns* in this paper) between opinion words and targets. For example, we can extract "clear" and "screen" by using a syntactic pattern "Adj-{mod}-Noun", which captures the opinion relation between them. Although previous works have shown the effectiveness of syntactic patterns for this task (Qiu et al., 2009; Zhang et al., 2010), they still have some limitations as follows.

False Opinion Relations: As an example, the phrase "everyday at school" can be matched by a pattern "Adj-{mod}-(Prep)-{pcomp-n}-Noun", but it doesn't bear any sentiment orientation. We call such relations that match opinion patterns but express no opinion *false opinion relations*. Previous pattern learning algorithms (Zhuang et al., 2006; Kessler and Nicolov, 2009; Jijkoun et al., 2010) often extract opinion patterns by frequency. However, some high-frequency syntactic patterns can have very poor precision (Kessler and Nicolov, 2009).

False Opinion Targets: In another case, the phrase "wonderful time" can be matched by an opinion pattern "Adj-{mod}-Noun", which is widely used in previous works (Popescu and Etzioni, 2005; Qiu et al., 2009). As can be seen, this phrase does express a positive opinion but unfortunately "time" is not a valid opinion target for most domains such as MP3. Thus, *false opinion targets* are extracted. Due to the lack of ground-truth knowledge for opinion targets, non-target terms introduced in this way can be hardly filtered out.

Long-tail Opinion Targets: We further notice that previous works prone to extract opinion targets with high frequency (Hu and Liu, 2004; Popescu and Etzioni, 2005; Qiu et al., 2009; Zhu et al., 2009), and they often have difficulty in identifying the infrequent or *long-tail opinion targets*.

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 1764–1773,
Sofia, Bulgaria, August 4-9 2013. ©2013 Association for Computational Linguistics

To address the problems stated above, this paper proposes a two-stage framework for mining opinion words and opinion targets. The underlying motivation is analogous to the novel idea "Mine the Easy, Classify the Hard" (Dasgupta and Ng, 2009). In our first stage, we propose a ***Sentiment Graph Walking*** algorithm to cope with the false opinion relation problem, which mines easy cases of opinion words/targets. We speculate that it may be helpful to introduce a confidence score for each pattern. Concretely, we create a *Sentiment Graph* to model opinion relations among opinion word/target/pattern candidates and apply random walking to estimate confidence of them. Thus, confidence of pattern is considered in a unified process. Patterns that often extract false opinion relations will have low confidence, and terms introduced by low-confidence patterns will also have low confidence accordingly. This could potentially improve the extraction accuracy.

In the second stage, we identify the hard cases, which aims to filter out false opinion targets and extract long-tail opinion targets. Previous supervised methods have been shown to achieve state-of-the-art results for this task (Wu et al., 2009; Jin and Ho, 2009; Li et al., 2010). However, the big challenge for fully supervised method is the lack of annotated training data. Therefore, we adopt a **self-learning strategy**. Specifically, we employ a semi-supervised classifier to refine the target results from the first stage, which uses some highly confident target candidates as the initial labeled examples. Then opinion words are also refined.

Our main contributions are as follows:

- We propose a *Sentiment Graph Walking* algorithm to mine opinion words and opinion targets from reviews, which naturally incorporates confidence of syntactic pattern in a graph to improve extraction performance. To our best knowledge, the incorporation of pattern confidence in such a Sentiment Graph has never been studied before for opinion words/targets mining task (Section 3).
- We adopt a self-learning method for refining opinion words/targets generated by *Sentiment Graph Walking*. Specifically, it can remove high-frequency noise terms and capture long-tail opinion targets in corpora (Section 4).
- We perform experiments on three real world datasets, which demonstrate the effectiveness of our method compared with state-of-the-art unsupervised methods (Section 5).

2 Related Work

In opinion words/targets mining task, most unsupervised methods rely on identifying opinion relations between opinion words and opinion targets. Hu and Liu (2004) proposed an association mining technique to extract opinion words/targets. The simple heuristic rules they used may potentially introduce many false opinion words/targets. To identify opinion relations more precisely, subsequent research work exploited syntax information. Popescu and Etzioni (2005) used manually complied syntactic patterns and Pointwise Mutual Information (PMI) to extract opinion words/targets. Qiu et al. (2009) proposed a bootstrapping framework called *Double Propagation* which introduced eight heuristic syntactic rules. While manually defining syntactic patterns could be time-consuming and error-prone, we learn syntactic patterns automatically from data.

There have been extensive works on mining opinion words and opinion targets by syntactic pattern learning. Riloff and Wiebe (2003) performed pattern learning through bootstrapping while extracting subjective expressions. Zhuang et al. (2006) obtained various dependency relationship templates from an annotated movie corpus and applied them to supervised opinion words/targets extraction. Kobayashi et al. (2007) adopted a supervised learning technique to search for useful syntactic patterns as contextual clues. Our approach is similar to (Wiebe and Riloff, 2005) and (Xu et al., 2013), all of which apply syntactic pattern learning and adopt self-learning strategy. However, the task of (Wiebe and Riloff, 2005) was to classify sentiment orientations in sentence level, while ours needs to extract more detailed information in term level. In addition, our method extends (Xu et al., 2013), and we give a more complete and in-depth analysis on the aforementioned problems in the first section. There were also many works employed graph-based method (Li et al., 2012; Zhang et al., 2010; Hassan and Radev, 2010; Liu et al., 2012), but none of previous works considered confidence of patterns in the graph.

In supervised approaches, various kinds of models were applied, such as HMM (Jin and Ho, 2009), SVM (Wu et al., 2009) and CRFs (Li et al., 2010). The downside of supervised methods was the difficulty of obtaining annotated training data in practical applications. Also, classifiers trained on one domain often fail to give satisfactory results when shifted to another domain. Our method does not rely on annotated training data.

3 The First Stage: Sentiment Graph Walking Algorithm

In the first stage, we propose a graph-based algorithm called *Sentiment Graph Walking* to mine opinion words and opinion targets from reviews.

3.1 Opinion Pattern Learning for Candidates Generation

For a given sentence, we first obtain its dependency tree. Following (Hu and Liu, 2004; Popescu and Etzioni, 2005; Qiu et al., 2009), we regard all adjectives as opinion word candidates (OC) and all nouns or noun phrases as opinion target candidates (TC). A statistic-based method in (Zhu et al., 2009) is used to detect noun phrases. Then candidates are replaced by wildcards "<OC>" or "<TC>". Figure 1 gives a dependency tree example generated by Minipar (Lin, 1998).

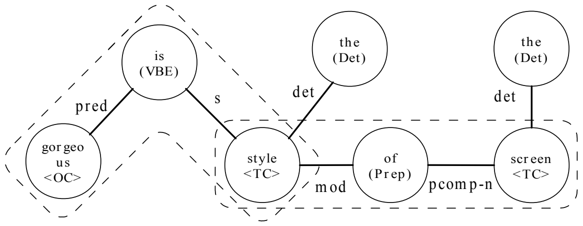

Figure 1: The dependency tree of the sentence "The style of the screen is gorgeous".

We extract two kinds of opinion patterns: "OC-TC" pattern and "TC-TC" pattern. The "OC-TC" pattern is the shortest path between an OC wildcard and a TC wildcard in dependency tree, which captures opinion relation between an opinion word candidate and an opinion target candidate. Similarly, the "TC-TC" pattern captures opinion relation between two opinion target candidates.[1] Words in opinion patterns are replaced by their POS tags, and we constrain that there are at most two words other than wildcards in each pattern. In Figure 1, there are two opinion patterns marked out by dash lines: "<OC>{pred}(VBE){s}<TC>" for the "OC-TC" type and "<TC>{mod}(Prep){pcomp-n}<TC>" for the "TC-TC" type. After all patterns are generated, we drop those patterns with frequency lower than a threshold F.

3.2 Sentiment Graph Construction

To model the opinion relations among opinion words/targets and opinion patterns, a graph named as *Sentiment Graph* is constructed, which is a weighted, directed graph $G = (V, E, W)$, where

- $V = \{V_{oc} \cup V_{tc} \cup V_p\}$ is the set of vertices in G, where V_{oc}, V_{tc} and V_p represent the set of opinion word candidates, opinion target candidates and opinion patterns, respectively.
- $E = \{E_{po} \cup E_{pt}\} \subseteq \{V_p \times V_{oc}\} \cup \{V_p \times V_{tc}\}$ is the weighted, bi-directional edge set in G, where E_{po} and E_{pt} are mutually exclusive sets of edges connecting opinion word/target vertices to opinion pattern vertices. Note that there are no edges between V_{oc} and V_{tc}.
- $W : E \rightarrow \mathbb{R}^+$ is the weight function which assigns non-negative weight to each edge. For each $(e : v_a \rightarrow v_b) \in E$, where $v_a, v_b \in V$, the weight function $w(v_a, v_b) = freq(v_a, v_b)/freq(v_a)$, where $freq(\cdot)$ is the frequency of a candidate extracted by opinion patterns or co-occurrence frequency between two candidates.

Figure 2 shows an example of Sentiment Graph.

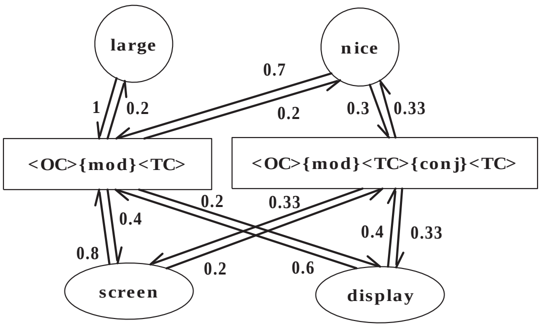

Figure 2: An example of Sentiment Graph.

3.3 Confidence Estimation by Random Walking with Restart

We believe that considering confidence of patterns can potentially improve the extraction accuracy. Our intuitive idea is: (i) If an opinion word/target is with higher confidence, the syntactic patterns containing this term are more likely to be used to express customers' opinion. (ii) If an opinion pattern has higher confidence, terms extracted by this pattern are more likely to be correct. It's a reinforcement process.

[1] We do not identify the opinion relation "OC-OC" because this relation is often unreliable.

We use Random Walking with Restart (RWR) algorithm to implement our idea described above. Let $\mathbf{M}_{oc\_p}$ denotes the transition matrix from V_{oc} to V_p, for $v_o \in V_{oc}, v_p \in V_p$, $\mathbf{M}_{oc\_p}(v_o, v_p) = w(v_o, v_p)$. Similarly, we have $\mathbf{M}_{tc\_p}$, $\mathbf{M}_{p\_oc}$, $\mathbf{M}_{p\_tc}$. Let $\mathbf{c}$ denotes confidence vector of candidates so $\mathbf{c}^t_{oc}$, $\mathbf{c}^t_{tc}$ and $\mathbf{c}^t_p$ are confidence vectors for opinion word/target/pattern candidates after walking t steps. Initially $\mathbf{c}^0_{oc}$ is uniformly distributed on a few domain-independent opinion word seeds, then the following formula are updated iteratively until $\mathbf{c}^t_{tc}$ and $\mathbf{c}^t_{oc}$ converge:

$$\mathbf{c}^{t+1}_p = \mathbf{M}^T_{oc\_p} \times \mathbf{c}^t_{oc} + \mathbf{M}^T_{tc\_p} \times \mathbf{c}^t_{tc} \quad (1)$$

$$\mathbf{c}^{t+1}_{oc} = (1-\lambda)\mathbf{M}^T_{p\_oc} \times \mathbf{c}^t_p + \lambda \mathbf{c}^0_{oc} \quad (2)$$

$$\mathbf{c}^{t+1}_{tc} = \mathbf{M}^T_{p\_tc} \times \mathbf{c}^t_p \quad (3)$$

where $\mathbf{M}^T$ is the transpose of matrix $\mathbf{M}$ and λ is a small probability of teleporting back to the seed vertices which prevents us from walking too far away from the seeds. In the experiments below, λ is set 0.1 empirically.

4 The Second Stage: Refining Extracted Results Using Self-Learning

At the end of the first stage, we obtain a ranked list of opinion words and opinion targets, in which higher ranked terms are more likely to be correct. Nevertheless, there are still some issues needed to be addressed:

1) In the target candidate list, some high-frequency frivolous general nouns such as “thing” and “people” are also highly ranked. This is because there exist many opinion expressions containing non-target terms such as “good thing”, “nice people”, etc. in reviews. Due to the lack of ground-truth knowledge for opinion targets, the false opinion target problem still remains unsolved.
2) In another aspect, long-tail opinion targets may have low degree in Sentiment Graph. Hence their confidence will be low although they may be extracted by some high quality patterns. Therefore, the first stage is incapable of dealing with the long-tail opinion target problem.
3) Furthermore, the first stage also extracts some high-frequency false opinion words such as “every”, “many”, etc. Many terms of this kind are introduced by high-frequency false opinion targets, for there are large amounts of phrases like “every time” and “many people”. So this issue is a side effect of the false opinion target problem.

To address these issues, we exploit a self-learning strategy. For opinion targets, we use a semi-supervised binary classifier called *target refining classifier* to refine target candidates. For opinion words, we use the classified list of opinion targets to further refine the extracted opinion word candidates.

4.1 Opinion Targets Refinement

There are two keys for opinion target refinement: (i) How to generate the initial labeled data for target refining classifier. (ii) How to properly represent a long-tail opinion target candidate other than comparing frequency between different targets.

For the first key, it is clearly improper to select high-confidence targets as positive examples and choose low-confidence targets as negative examples[2], for there are noise with high confidence and long-tail targets with low confidence. Fortunately, a large proportion of general noun noises are the most frequent words in common texts. Therefore, we can generate a small domain-independent general noun (GN) corpus from large web corpora to cover some most frequently used general noun examples. Then labeled examples can be drawn from the target candidate list and the GN corpus.

For the second key, we utilize opinion words and opinion patterns with their confidence scores to represent an opinion target. By this means, a long-tail opinion target can be determined by its own contexts, whose weights are learnt from contexts of frequent opinion targets. Thus, if a long-tail opinion target candidate has high contextual support, it will have higher probability to be found out in despite of its low frequency.

Creation of General Noun Corpora. 1000 most frequent nouns in Google-1-gram[3] were selected as general noun candidates. On the other hand, we added all nouns in the top three levels of hyponyms in four WordNet (Miller, 1995) synsets “object”, “person”, “group” and “measure” into the GN corpus. Our idea was based on the fact that a term is more general when it sits in higher level in the WordNet hierarchy. Then inapplicable candidates were discarded and a 3071-word English

[2] Note that the “positive” and “negative” here denote opinion targets and non-target terms respectively and they do not indicate sentiment polarities.

[3] http://books.google.com/ngrams.

GN corpus was created. Another Chinese GN corpus with 3493 words was generated in the similar way from HowNet (Gan and Wong, 2000).

Generation of Labeled Examples. Let $\mathscr{T} = \{\mathscr{Y}_{+1}, \mathscr{Y}_{-1}\}$ denotes the initial labeled set, where N most highly confident target candidates but not in our GN corpora are regarded as the positive example set $\mathscr{Y}_{+1}$, other N terms from GN corpora which are also top ranked in the target list are selected as the negative example set $\mathscr{Y}_{-1}$. The reminder unlabeled candidates are denoted by $\mathscr{T}^*$.

Feature Representation for Classifier. Given $\mathscr{T}$ and $\mathscr{T}^*$ in the form of $\{(\mathbf{x_i}, y_i)\}$. For a target candidate t_i, $\mathbf{x_i} = (o_1, \ldots, o_n, p_1, \ldots, p_m)^T$ represents its feature vector, where o_j is the opinion word feature and p_k is the opinion pattern feature. The value of feature is defined as follows,

$$x(o_j) = conf(o_j) \times \frac{\sum_{p_k} freq(t_i, o_j, p_k)}{freq(o_j)} \quad (4)$$

$$x(p_k) = conf(p_k) \times \frac{\sum_{o_j} freq(t_i, o_j, p_k)}{freq(p_k)} \quad (5)$$

where $conf(\cdot)$ denotes confidence score estimated by RWR, $freq(\cdot)$ has the same meaning as in Section 3.2. Particularly, $freq(t_i, o_j, p_k)$ represents the frequency of pattern p_k extracting opinion target t_i and opinion word o_j.

Target Refinement Classifier: We use support vector machine as the binary classifier. Hence, the classification problem can be formulated as to find a hyperplane $< \mathbf{w}, b >$ that separates both labeled set $\mathscr{T}$ and unlabeled set $\mathscr{T}^*$ with maximum margin. The optimization goal is to minimize over $(\mathscr{T}, \mathscr{T}^*, \mathbf{w}, b, \xi_1, ..., \xi_n, \xi_1^*, ..., \xi_k^*)$:

$$\begin{aligned} &\frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=0}^{n}\xi_i + C^*\sum_{j=0}^{k}\xi_j^* \\ subject\ to: &\ \forall_{i=1}^{n}: y_i[\mathbf{w}\cdot\mathbf{x_i} + b] \geq 1 - \xi_i \\ &\ \forall_{j=1}^{k}: y_j^*[\mathbf{w}\cdot\mathbf{x_j^*} + b] \geq 1 - \xi_j^* \\ &\ \forall_{i=1}^{n}: \xi_i > 0 \\ &\ \forall_{j=1}^{k}: \xi_j^* > 0 \end{aligned}$$

where $y_i, y_j^* \in \{+1, -1\}$, $\mathbf{x_i}$ and $\mathbf{x_j^*}$ represent feature vectors, C and C^* are parameters set by user. This optimization problem can be implemented by a typical Transductive Support Vector Machine (TSVM) (Joachims, 1999).

4.2 Opinion Words Refinement

We use the classified opinion target results to refine opinion words by the following equation,

$$s(o_j) = \sum_{t_i \in T}\sum_{p_k} \frac{s(t_i)conf(p_k)freq(t_i, o_j, p_k)}{freq(t_i)}$$

where T is the opinion target set in which each element is classified as positive during opinion target refinement, $s(t_i)$ denotes confidence score exported by the target refining classifier. Particularly, $freq(t_i) = \sum_{o_j}\sum_{p_k} freq(t_i, o_j, p_k)$. A higher score of $s(o_j)$ means that candidate o_j is more likely to be an opinion word.

5 Experiments

5.1 Datasets and Evaluation Metrics

Datasets: We select three real world datasets to evaluate our approach. The first one is called *Customer Review Dataset* (*CRD*) (Hu and Liu, 2004) which contains reviews on five different products (represented by D1 to D5) in English. The second dataset is pre-annotated and published in *COAE08*[4], where two domains of Chinese reviews are selected. At last, we employ a benchmark dataset in (Wang et al., 2011) and named it as *Large*. We manually annotated opinion words and opinion targets as the gold standard. Three annotators were involved. Firstly, two annotators were required to annotate out opinion words and opinion targets in sentences. When conflicts happened, the third annotator would make the final judgment. The average Kappa-values of the two domains were 0.71 for opinion words and 0.66 for opinion targets. Detailed information of our datasets is shown in Table 1.

Dataset	Domain	#Sentences	#OW	#OT
Large (English)	Hotel	10,000	434	1,015
	MP3	10,000	559	1,158
COAE08 (Chinese)	Camera	2,075	351	892
	Car	4,783	622	1,179

Table 1: The detailed information of datasets. OW stands for opinion words and OT stands for targets.

Pre-processing: Firstly, HTML tags are removed from texts. Then Minipar (Lin, 1998) is used to parse English corpora, and Standford Parser (Chang et al., 2009) is used for Chinese

[4] http://ir-china.org.cn/coae2008.html

Methods	**D1**			**D2**			**D3**			**D4**			**D5**			**Avg.**
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	F
Hu	0.75	0.82	0.78	0.71	0.79	0.75	0.72	0.76	0.74	0.69	0.82	0.75	0.74	0.80	0.77	0.76
DP	**0.87**	0.81	**0.84**	**0.90**	0.81	**0.85**	**0.90**	0.86	**0.88**	0.81	0.84	0.82	**0.92**	0.86	**0.89**	**0.86**
Zhang	0.83	0.84	0.83	0.86	0.85	**0.85**	0.86	**0.88**	0.87	0.80	0.85	0.82	0.86	0.86	0.86	0.85
Ours-Stage1	0.79	**0.85**	0.82	0.82	**0.87**	0.84	0.83	0.87	0.85	0.78	**0.88**	0.83	0.82	**0.88**	0.85	0.84
Ours-Full	0.86	0.82	**0.84**	0.88	0.83	**0.85**	0.89	0.86	0.87	**0.83**	0.86	**0.84**	0.89	0.85	0.87	**0.86**

Table 2: Results of opinion target extraction on the *Customer Review Dataset*.

Methods	**D1**			**D2**			**D3**			**D4**			**D5**			**Avg.**
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	F
Hu	0.57	**0.75**	0.65	0.51	0.76	0.61	0.57	0.73	0.64	0.54	0.62	0.58	0.62	0.67	0.64	0.62
DP	**0.64**	0.73	0.68	0.57	0.79	0.66	0.65	0.70	0.67	0.61	0.65	0.63	0.70	0.68	**0.69**	0.67
Ours-Stage1	0.61	**0.75**	0.67	0.55	**0.80**	0.65	0.63	**0.75**	**0.68**	0.60	**0.69**	0.64	0.68	**0.70**	**0.69**	0.67
Ours-Full	**0.64**	0.74	**0.69**	**0.59**	0.79	**0.68**	**0.66**	0.71	**0.68**	**0.65**	0.67	**0.66**	**0.72**	0.67	**0.69**	**0.68**

Table 3: Results of opinion word extraction on the *Customer Review Dataset*.

corpora. Stemming and fuzzy matching are also performed following previous work (Hu and Liu, 2004).

Evaluation Metrics: We evaluate our method by precision(P), recall(R) and F-measure(F).

5.2 Our Method vs. the State-of-the-art

Three state-of-the-art unsupervised methods are used as competitors to compare with our method.

Hu extracts opinion words/targets by using adjacency rules (Hu and Liu, 2004).

DP uses a bootstrapping algorithm named as *Double Propagation* (Qiu et al., 2009).

Zhang is an enhanced version of *DP* and employs HITS algorithm (Kleinberg, 1999) to rank opinion targets (Zhang et al., 2010).

Ours-Full is the full implementation of our method. We employ SVM$^{\text{light}}$ (Joachims, 1999) as the target refining classifier. Default parameters are used except the bias item is set 0.

Ours-Stage1 only uses *Sentiment Graph Walking* algorithm which does't have opinion word and opinion target refinement.

All of the above approaches use same five common opinion word seeds. The choice of opinion seeds seems reasonable, as most people can easily come up with 5 opinion words such as "good", "bad", etc. The performance on five products of *CRD* dataset is shown in Table 2 and Table 3. *Zhang* does not extract opinion words so their results for opinion words are not taken into account. We can see that *Ours-Stage1* achieves superior recall but has some loss in precision compared with *DP* and *Zhang*. This may be because the *CRD* dataset is too small and our statistic-based method may suffer from data sparseness. In spite of this, *Ours-Full* achieves comparable F-measure with *DP*, which is a well-designed rule-based method.

The results on two larger datasets are shown in Table 4 and Table 5, from which we can have the following observation: (i) All syntax-based-methods outperform *Hu*, showing the importance of syntactic information in opinion relation identification. (ii) *Ours-Full* outperforms the three competitors on all domains provided. (iii) *Ours-Stage1* outperforms *Zhang*, especially in terms of recall. We believe it benefits from our automatical pattern learning algorithm. Moreover, *Ours-Stage1* do not loss much in precision compared with *Zhang*, which indicates the applicability to estimate pattern confidence in Sentiment Graph. (iv) *Ours-Full* achieves 4-9% improvement in precision over the most accurate method, which shows the effectiveness of our second stage.

5.3 Detailed Discussions

This section gives several variants of our method to have a more detailed analysis.

Ours-Bigraph constructs a bi-graph between opinion words and targets, so opinion patterns are not included in the graph. Then RWR algorithm is used to only assign confidence to opinion word/target candidates.

Ours-Stage2 only contains the second stage, which doesn't apply *Sentiment Graph Walking* algorithm. Hence the confidence score $conf(\cdot)$ in Equations (4) and (5) have no values and they are set to 1. The initial labeled examples are exactly the same as *Ours-Full*. Due to the limitation of space, we only give analysis on opinion target extraction results in Figure 3.

Methods	MP3			Hotel			Camera			Car			Avg.
	P	R	F	P	R	F	P	R	F	P	R	F	F
Hu	0.53	0.55	0.54	0.55	0.57	0.56	0.63	0.65	0.64	0.62	0.58	0.60	0.58
DP	0.66	0.57	0.61	0.66	0.60	0.63	0.71	0.70	0.70	0.72	0.65	0.68	0.66
Zhang	0.65	0.62	0.63	0.64	0.66	0.65	0.71	0.78	0.74	0.69	0.68	0.68	0.68
Ours-Stage1	0.62	0.68	0.65	0.63	0.71	0.67	0.69	0.80	0.74	0.66	0.71	0.68	0.69
Ours-Full	**0.73**	**0.71**	**0.72**	**0.75**	**0.73**	**0.74**	**0.78**	**0.81**	**0.79**	**0.76**	**0.73**	**0.74**	**0.75**

Table 4: Results of opinion targets extraction on *Large* and *COAE08*.

Methods	MP3			Hotel			Camera			Car			Avg.
	P	R	F	P	R	F	P	R	F	P	R	F	F
Hu	0.48	0.65	0.55	0.51	0.68	0.58	0.72	0.74	0.73	0.70	0.71	0.70	0.64
DP	0.58	0.62	0.60	0.60	0.66	0.63	0.80	0.73	0.76	0.79	0.71	0.75	0.68
Ours-Stage1	0.59	**0.69**	0.64	0.61	**0.71**	0.66	0.79	**0.78**	0.78	0.77	**0.77**	0.77	0.71
Ours-Full	**0.64**	0.67	**0.65**	**0.67**	0.69	**0.68**	**0.82**	**0.78**	**0.80**	**0.80**	0.76	**0.78**	**0.73**

Table 5: Results of opinion words extraction on *Large* and *COAE08*.

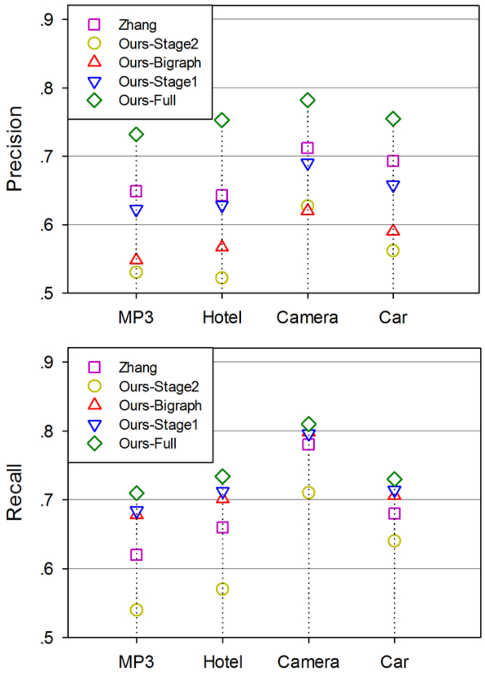

Figure 3: Opinion target extraction results.

5.3.1 The Effect of Sentiment Graph Walking

We can see that our graph-based methods (*Ours-Bigraph* and *Ours-Stage1*) achieve higher recall than *Zhang*. By learning patterns automatically, our method captures opinion relations more efficiently. Also, *Ours-Stage1* outperforms *Ours-Bigraph*, especially in precision. We believe it is because *Ours-Stage1* estimated confidence of patterns so false opinion relations are reduced. Therefore, the consideration of pattern confidence is beneficial as expected, which alleviates the false opinion relation problem. On another hand, we find that *Ours-Stage2* has much worse performance than *Ours-Full*. This shows the effectiveness of *Sentiment Graph Walking* algorithm since the confidence scores estimated in the first stage are indispensable and indeed key to the learning of the second stage.

5.3.2 The Effect of Self-Learning

Figure 4 shows the average Precision@N curve of four domains on opinion target extraction. *Ours-GN-Only* is implemented by only removing 50 initial negative examples found by our GN corpora. We can see that the GN corpora work quite well, which find out most top-ranked false opinion targets. At the same time, *Ours-Full* has much better performance than *Ours-GN-Only* which indicates that *Ours-Full* can filter out more noises other than the initial negative examples. Therefore, our self-learning strategy alleviates the shortcoming of false opinion target problem. Moreover, Table 5 shows that the performance of opinion word extraction is also improved based on the classified results of opinion targets.

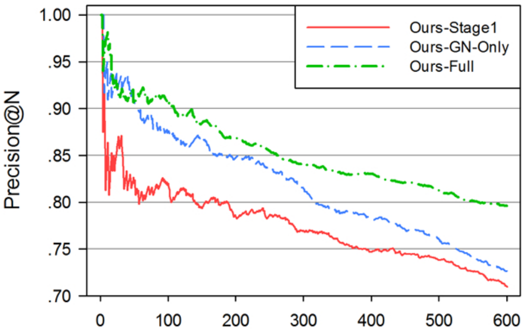

Figure 4: The average precision@N curve of the four domains on opinion target extraction.

ID	**Pattern**	**Example**	**#Ext.**	**Conf.**	**PrO**	**PrT**
#1	<OC>{mod}<TC>	it has a **clear** *screen*	7344	0.3938	0.59	0.66
#2	<TC>{subj}<OC>	the *sound quality* is **excellent**	2791	0.0689	0.62	0.70
#3	<TC>{conj}<TC>	the *size* and *weight* make it **convenient**	3620	0.0208	N/A	0.67
#4	<TC>{subj}<TC>	the *button layout* is a **simplistic** *plus*	1615	0.0096	N/A	0.67
#5	<OC>{pnmod}<TC>	the *buttons* **easier** to use	128	0.0014	0.61	0.34
#6	<TC>{subj}(V){s}(VBE){subj}<OC>	*software* provided is **simple**	189	0.0015	0.54	0.33
#7	<OC>{mod}(Prep){pcomp-c}(V){obj}<TC>	**great** for playing *audible books*	211	0.0013	0.43	0.48

Table 6: Examples of English patterns. #Ext. represent number of terms extracted, Conf. denotes confidence score estimated by RWR and PrO/PrT stand for precisions of extraction on opinion words/targets of a pattern respectively. Opinion words in examples are in bold and opinion targets are in italic.

Figure 5 gives the recall of long-tail opinion targets[5] extracted, where *Ours-Full* is shown to have much better performance than *Ours-Stage1* and the three competitors. This observation proves that our method can improve the limitation of long-tail opinion target problem.

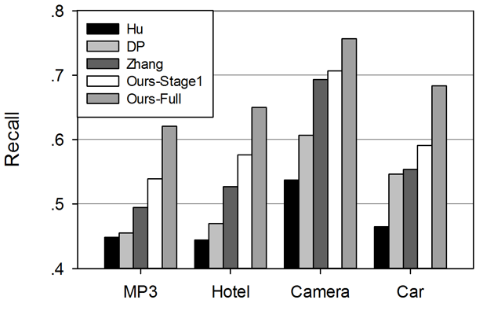

Figure 5: The recall of long-tail opinion targets.

5.3.3 Analysis on Opinion Patterns

Table 6 shows some examples of opinion pattern and their extraction accuracy on MP3 reviews in the first stage. Pattern #1 and #2 are the two most high-confidence opinion patterns of "OC-TC" type, and Pattern #3 and #4 demonstrate two typical "TC-TC" patterns. As these patterns extract too many terms, the overall precision is very low. We give Precision@400 of them, which is more meaningful because only top listed terms in the extracted results are regarded as opinion targets. Pattern #5 and #6 have high precision on opinion words but low precision on opinion targets. This observation demonstrates the false opinion target problem. Pattern #7 is a pattern example that extracts many false opinion relations and it has low precision for both opinion words and opinion targets. We can see that Pattern #7 has a lower confidence compared with Pattern #5 and #6 although it extracts more words. It's because it has a low probability of walking from opinion seeds to this pattern. This further proves that our method can reduce the confidence of low-quality patterns.

5.3.4 Sensitivity of Parameters

Finally, we study the sensitivity of parameters when recall is fixed at 0.70. Figure 6 shows the precision curves at different N initial training examples and F filtering frequency. We can see that the performance saturates when N is set to 50 and it does not vary much under different F, showing the robustness of our method. We thus set N to 50, and F to 3 for *CRD*, 5 for *COAE08* and 10 for *Large* accordingly.

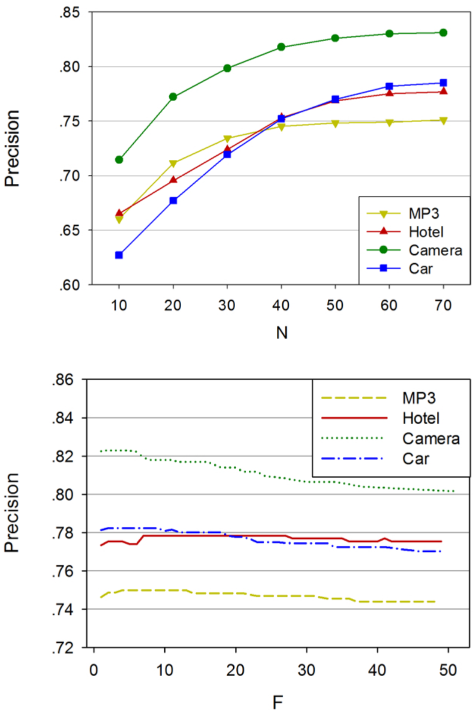

Figure 6: Influence of parameters.

[5] Since there is no explicit definition for the notion "long-tail", we conservatively regard 60% opinion targets with the lowest frequency as the "long-tail" terms.

6 Conclusion and Future Work

This paper proposes a novel two-stage framework for mining opinion words and opinion targets. In the first stage, we propose a *Sentiment Graph Walking* algorithm, which incorporates syntactic patterns in a Sentiment Graph to improve the extraction performance. In the second stage, we propose a self-learning method to refine the result of first stage. The experimental results show that our method achieves superior performance over state-of-the-art unsupervised methods.

We further notice that opinion words are not limited to adjectives but can also be other type of word such as verbs or nouns. Identifying all kinds of opinion words is a more challenging task. We plan to study this problem in our future work.

Acknowledgement

Thanks to Prof. Yulan He for her insightful advices. This work was supported by the National Natural Science Foundation of China (No. 61070106, No. 61272332 and No. 61202329), the National High Technology Development 863 Program of China (No. 2012AA011102), the National Basic Research Program of China (No. 2012CB316300), Tsinghua National Laboratory for Information Science and Technology (TNList) Cross-discipline Foundation and the Opening Project of Beijing Key Laboratory of Internet Culture and Digital Dissemination Research (ICDD201201).

References

Pi-Chuan Chang, Huihsin Tseng, Dan Jurafsky, and Christopher D. Manning. 2009. Discriminative reordering with chinese grammatical relations features. In *Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation*, SSST '09, pages 51–59.

Sajib Dasgupta and Vincent Ng. 2009. Mine the easy, classify the hard: a semi-supervised approach to automatic sentiment classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 701–709.

Kok Wee Gan and Ping Wai Wong. 2000. Annotating information structures in chinese texts using hownet. In *Proceedings of the second workshop on Chinese language processing: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics - Volume 12*, CLPW '00, pages 85–92, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ahmed Hassan and Dragomir Radev. 2010. Identifying text polarity using random walks. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 395–403, Stroudsburg, PA, USA. Association for Computational Linguistics.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 168–177, New York, NY, USA. ACM.

Valentin Jijkoun, Maarten de Rijke, and Wouter Weerkamp. 2010. Generating focused topic-specific sentiment lexicons. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 585–594, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wei Jin and Hung Hay Ho. 2009. A novel lexicalized hmm-based learning framework for web opinion mining. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 465–472.

Thorsten Joachims. 1999. Transductive inference for text classification using support vector machines. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pages 200–209.

Jason Kessler and Nicolas Nicolov. 2009. Targeting sentiment expressions through supervised ranking of linguistic configurations. In *Proceedings of the Third International AAAI Conference on Weblogs and Social Media*.

Jon M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, September.

Nozomi Kobayashi, Kentaro Inui, and Yuji Matsumoto. 2007. Extracting aspect-evaluation and aspect-of relations in opinion mining. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1065–1074, June.

Fangtao Li, Chao Han, Minlie Huang, Xiaoyan Zhu, Ying-Ju Xia, Shu Zhang, and Hao Yu. 2010. Structure-aware review mining and summarization. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 653–661, Stroudsburg, PA, USA. Association for Computational Linguistics.

Fangtao Li, Sinno Jialin Pan, Ou Jin, Qiang Yang, and Xiaoyan Zhu. 2012. Cross-domain co-extraction of sentiment and topic lexicons. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 410–419, July.

Dekang Lin. 1998. Dependency-based evaluation of minipar. In *Workshop on Evaluation of Parsing Systems at ICLRE*.

Kang Liu, Liheng Xu, and Jun Zhao. 2012. Opinion target extraction using word-based translation model. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1346–1356, Stroudsburg, PA, USA. Association for Computational Linguistics.

George A. Miller. 1995. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41.

Ana-Maria Popescu and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 339–346.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2009. Expanding domain sentiment lexicon through double propagation. In *Proceedings of the 21st international jont conference on Artifical intelligence*, IJCAI'09, pages 1199–1204.

Ellen Riloff and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, EMNLP '03, pages 105–112, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hongning Wang, Yue Lu, and ChengXiang Zhai. 2011. Latent aspect rating analysis without aspect keyword supervision. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 618–626, New York, NY, USA. ACM.

Janyce Wiebe and Ellen Riloff. 2005. Creating subjective and objective sentence classifiers from unannotated texts. In *Proceedings of the 6th international conference on Computational Linguistics and Intelligent Text Processing*, CICLing'05, pages 486–497.

Yuanbin Wu, Qi Zhang, Xuanjing Huang, and Lide Wu. 2009. Phrase dependency parsing for opinion mining. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, pages 1533–1541.

Liheng Xu, Kang Liu, Siwei Lai, Yubo Chen, and Jun Zhao. 2013. Walk and learn: A two-stage approach for opinion words and opinion targets co-extraction. In *Proceedings of the 22nd International World Wide Web Conference*, WWW '13.

Lei Zhang, Bing Liu, Suk Hwan Lim, and Eamonn O'Brien-Strain. 2010. Extracting and ranking product features in opinion documents. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1462–1470.

Jingbo Zhu, Huizhen Wang, Benjamin K. Tsou, and Muhua Zhu. 2009. Multi-aspect opinion polling from textual reviews. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 1799–1802.

Li Zhuang, Feng Jing, and Xiao-Yan Zhu. 2006. Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, CIKM '06, pages 43–50.