# Co-Extracting Opinion Targets and Opinion Words from Online Reviews Based on the Word Alignment Model

Kang Liu, Liheng Xu, and Jun Zhao

**Abstract**—Mining opinion targets and opinion words from online reviews are important tasks for fine-grained opinion mining, the key component of which involves detecting opinion relations among words. To this end, this paper proposes a novel approach based on the partially-supervised alignment model, which regards identifying opinion relations as an alignment process. Then, a graph-based co-ranking algorithm is exploited to estimate the confidence of each candidate. Finally, candidates with higher confidence are extracted as opinion targets or opinion words. Compared to previous methods based on the nearest-neighbor rules, our model captures opinion relations more precisely, especially for long-span relations. Compared to syntax-based methods, our word alignment model effectively alleviates the negative effects of parsing errors when dealing with informal online texts. In particular, compared to the traditional unsupervised alignment model, the proposed model obtains better precision because of the usage of partial supervision. In addition, when estimating candidate confidence, we penalize higher-degree vertices in our graph-based co-ranking algorithm to decrease the probability of error generation. Our experimental results on three corpora with different sizes and languages show that our approach effectively outperforms state-of-the-art methods.

**Index Terms**—Opinion mining, opinion targets extraction, opinion words extraction

## 1 Introduction

With the rapid development of Web 2.0, a huge number of product reviews are springing up on the Web. From these reviews, customers can obtain first-hand assessments of product information and direct supervision of their purchase actions. Meanwhile, manufacturers can obtain immediate feedback and opportunities to improve the quality of their products in a timely fashion. Thus, mining opinions from online reviews has become an increasingly urgent activity and has attracted a great deal of attention from researchers [1], [2], [3], [4].

To extract and analyze opinions from online reviews, it is unsatisfactory to merely obtain the overall sentiment about a product. In most cases, customers expect to find fine-grained sentiments about an aspect or feature of a product that is reviewed. For example:

> *"This phone has a colorful and big screen, but its LCD resolution is very disappointing."*

Readers expect to know that the reviewer expresses a positive opinion of the phone's screen and a negative opinion of the screen's resolution, not just the reviewer's overall sentiment. To fulfill this aim, both opinion targets and opinion words must be detected. First, however, it is necessary to extract and construct an opinion target list and an opinion word lexicon, both of which can provide prior knowledge that is useful for fine-grained opinion mining and both of which are the focus of this paper.

An opinion target is defined as the object about which users express their opinions, typically as nouns or noun phrases. In the above example, "*screen*" and "*LCD resolution*" are two opinion targets. Previous methods have usually generated an opinion target list from online product reviews. As a result, opinion targets usually are product features or attributes. Accordingly this subtask is also called as product feature extraction [5], [6]. In addition, opinion words are the words that are used to express users' opinions. In the above example, "*colorful*", "*big*" and "*disappointing*" are three opinion words. Constructing an opinion words lexicon is also important because the lexicon is beneficial for identifying opinion expressions.

For these two subtasks, previous work generally adopted a collective extraction strategy. The intuition represented by this strategy was that in sentences, opinion words usually co-occur with opinion targets, and there are strong modification relations and associations among them (which in this paper are called opinion relations or opinion associations). Therefore, many methods jointly extracted opinion targets and opinion words in a bootstrapping manner [1], [4], [6], [7]. For example, "*colorful*" and "*big*" are usually used to modify "*screen*" in the cell-phone domain, and there are remarkable opinion relations among them. If we know "*big*" to be an opinion word, then "*screen*" is very likely to be an opinion target in this domain. Next, the extracted opinion target "*screen*" can be used to deduce that "*colorful*" is most likely an opinion word. Thus, the extraction is alternatively performed between opinion targets and opinion words until there is no item left to extract.

• *The authors are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. E-mail: {kliu, lhxu, jzhao}@nlpr.ia.ac.cn.*

*Manuscript received 3 Sept. 2013; revised 29 June 2014; accepted 6 July 2014. Date of publication 16 July 2014; date of current version 28 Jan. 2015. Recommended for acceptance by M. Sanderson. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TKDE.2014.2339850*

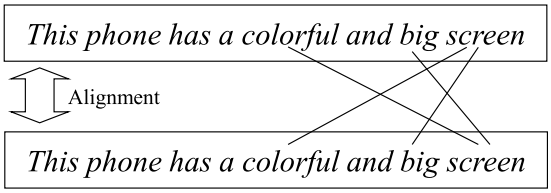



Fig. 1. Mining opinion relations between words using the word alignment model.

Although there are many variants of bootstrapping-based approaches [1], [7], [8], we notice that these methods still have some limitations as follows:

1) In previous methods, mining the opinion relations between opinion targets and opinion words was the key to collective extraction. To this end, the most-adopted techniques have been *nearest-neighbor rules* [5], [8], [9] and *syntactic patterns* [6], [10]. Nearest-neighbor rules regard the nearest adjective/verb to a noun/noun phrase in a limited window as its modifier. Clearly, this strategy cannot obtain precise results because there exist long-span modified relations and diverse opinion expressions. To address this problem, several methods exploited syntactic information, in which the opinion relations among words are decided according to their dependency relations in the parsing tree. Accordingly several heuristic syntactic patterns were designed [6], [7], [10]. However, online reviews usually have informal writing styles, including grammatical errors, typographical errors, and punctuation errors. This makes the existing parsing tools, which are usually trained on formal texts such as news reports, prone to generating errors. Accordingly, these syntax-based methods, which heavily depend on parsing performance, suffer from parsing errors and often do not work well [3]. To improve the performance of these methods, we can specially design exquisite, high-precision patterns. However, with an increase in corpus size, this strategy is likely to miss more items and has lower recall. Therefore, how to precisely detect the opinion relations among words is a considerable challenge in this task.
2) The collective extraction adopted by most previous methods was usually based on a bootstrapping framework, which has the problem of error propagation. If some errors are extracted by an iteration, they would not be filtered out in subsequent iterations. As a result, more errors are accumulated iteratively. Therefore, how to alleviate, or even avoid, error propagation is another challenge in this task.

To resolve these two challenges, this paper presents an alignment-based approach with graph co-ranking to collectively extract opinion targets and opinion words. Our main contributions can be summarized as follows:

1) To precisely mine the opinion relations among words, we propose a method based on a monolingual word alignment model (WAM). An opinion target can find its corresponding modifier through word alignment. For example in Fig. 1, the opinion words "*colorful*" and "*big*" are aligned with the target word "*screen*". Compared to previous nearest-neighbor rules, the WAM does not constrain identifying modified relations to a limited window; therefore, it can capture more complex relations, such as long-span modified relations. Compared to syntactic patterns, the WAM is more robust because it does not need to parse informal texts. In addition, the WAM can integrate several intuitive factors, such as word co-occurrence frequencies and word positions, into a unified model for indicating the opinion relations among words. Thus, we expect to obtain more precise results on opinion relation identification. The alignment model used in [4] has proved to be effective for opinion target extraction. However, for opinion word extraction, there is still no straightforward evidence to demonstrate the WAM's effectiveness.
2) We further notice that standard word alignment models are often trained in a completely unsupervised manner, which results in alignment quality that may be unsatisfactory. We certainly can improve alignment quality by using supervision [11]. However, it is both time consuming and impractical to manually label full alignments in sentences. Thus, we further employ a partially-supervised word alignment model (PSWAM). We believe that we can easily obtain a portion of the links of the full alignment in a sentence. These can be used to constrain the alignment model and obtain better alignment results. To obtain partial alignments, we resort to syntactic parsing. Although existing syntactic parsing algorithms cannot precisely obtain the whole syntactic tree of informal sentences, some opinion relations can still be obtained precisely by using high-precision syntactic patterns. A constrained EM algorithm based on hill-climbing is then performed to determine all of the alignments in sentences, where the model will be consistent with these links as much as possible. In this way, some errors induced by completely unsupervised WAMs will be corrected. For example, in Fig. 2, "*kindly*" and "*courteous*" are incorrectly identified as modifiers for "*foods*" if the WAM is performed in a wholly unsupervised manner. However, by using some syntactic patterns, we can assert that "*courteous*" should be aligned to "*services*". Through the PSWAM, "*kindly*" and "*courteous*" are correctly linked to "*services*". This model not only inherits the advantages of the word alignment model for opinion relation identification, but it also has a more precise performance because of the use of partial supervision. Thus, it is reasonable to expect that the PSWAM is likely to yield better results compared to traditional methods for extracting opinion targets and opinion words.
3) To alleviate the problem of error propagation, we resort to graph co-ranking. Extracting opinion targets/words is regarded as a co-ranking process. Specifically, a graph, named as *Opinion Relation Graph*, is constructed to model all opinion target/word candidates and the opinion relations among them. A random walk based co-ranking algorithm is then

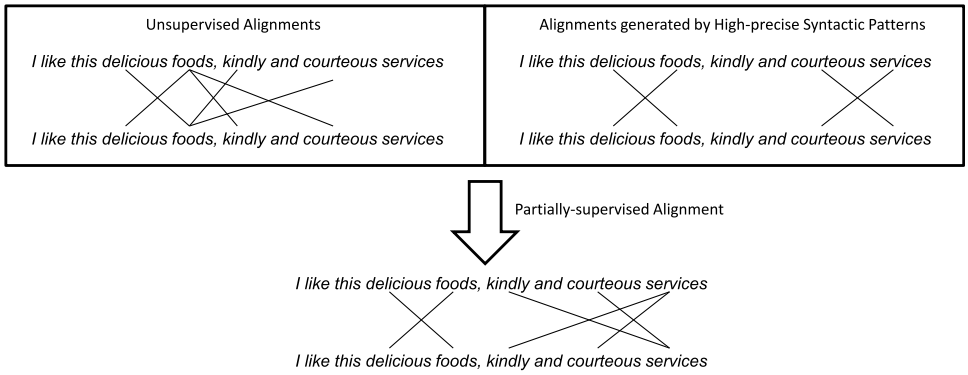



Fig. 2. Mining opinion relations between words using partially supervised alignment model.

proposed to estimate each candidate's confidence on the graph. In this process, we penalize high-degree vertices to weaken their impacts and decrease the probability of a random walk running into unrelated regions on the graph. Meanwhile, we calculate the prior knowledge of candidates for indicating some noises and incorporating them into our ranking algorithm to make collaborated operations on candidate confidence estimations. Finally, candidates with higher confidence than a threshold are extracted. Compared to the previous methods based on the bootstrapping strategy, opinion targets/words are no longer extracted step by step. Instead, the confidence of each candidate is estimated in a global process with graph co-ranking. Intuitively, the error propagation is effectively alleviated.

To illustrate the effectiveness of the proposed method, we select real online reviews from different domains and languages as the evaluation datasets. We compare our method to several state-of-the-art methods on opinion target/word extraction. The experimental results show that our approach improves performance over the traditional methods.

## 2 Related Work

Opinion target and opinion word extraction are not new tasks in opinion mining. There is significant effort focused on these tasks [1], [6], [12], [13], [14]. They can be divided into two categories: sentence-level extraction and corpus-level extraction according to their extraction aims.

In sentence-level extraction, the task of opinion target/word extraction is to identify the opinion target mentions or opinion expressions in sentences. Thus, these tasks are usually regarded as sequence-labeling problems [13], [14], [15], [16]. Intuitively, contextual words are selected as the features to indicate opinion targets/words in sentences. Additionally, classical sequence labeling models are used to build the extractor, such as CRFs [13] and HMM [17]. Jin and Huang [17] proposed a lexicalized HMM model to perform opinion mining. Both [13] and [15] used CRFs to extract opinion targets from reviews. However, these methods always need the labeled data to train the model. If the labeled training data are insufficient or come from the different domains than the current texts, they would have unsatisfied extraction performance. Although [2] proposed a method based on transfer learning to facilitate cross-domain extraction of opinion targets/words, their method still needed the labeled data from out-domains and the extraction performance heavily depended on the relevance between in-domain and out-domain.

In addition, much research focused on corpus-level extraction. They did not identify the opinion target/word mentions in sentences, but aimed to extract a list of opinion targets or generate a sentiment word lexicon from texts. Most previous approaches adopted a collective unsupervised extraction framework. As mentioned in our first section, detecting opinion relations and calculating opinion associations among words are the key component of this type of method. Wang and Wang [8] adopted the co-occurrence frequency of opinion targets and opinion words to indicate their opinion associations. Hu and Liu [5] exploited nearest-neighbor rules to identify opinion relations among words. Next, frequent and explicit product features were extracted using a bootstrapping process. Only the use of co-occurrence information or nearest-neighbor rules to detect opinion relations among words could not obtain precise results. Thus, [6] exploited syntax information to extract opinion targets, and designed some syntactic patterns to capture the opinion relations among words. The experimental results showed that their method performed better than that of [5]. Moreover, [10] and [7] proposed a method, named as Double Propagation, that exploited syntactic relations among words to expand sentiment words and opinion targets iteratively. Their main limitation is that the patterns based on the dependency parsing tree could not cover all opinion relations. Therefore, Zhang et al. [3] extended the work by [7]. Besides the patterns used in [7], Zhang et al. further designed specific patterns to increase recall. Moreover, they used an HITS [18] algorithm to compute opinion target confidences to improve precision. Liu et al. [4] focused on opinion target extraction based on the WAM. They used a completely unsupervised WAM to capture opinion relations in sentences. Next, opinion targets were extracted in a standard random walk framework. Liu's experimental results showed that the WAM was effective for extracting opinion targets. Nonetheless, they present no evidence to demonstrate the effectiveness of the WAM on opinion word extraction.

Furthermore, a study employed topic modeling to identify implicit topics and sentiment words [19], [20], [21], [22]. The aims of these methods usually were not to extract an opinion target list or opinion word lexicon from reviews. Instead, they were to cluster for all words into corresponding

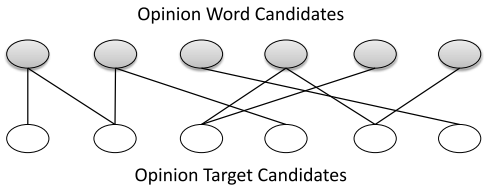



Fig. 3. Opinion relation graph.

aspects in reviews, which was different from the task in this paper. These methods usually adopted coarser techniques, such as frequency statistics and phrase detection, to detect the proper opinion targets/words. They put more emphasis on how to cluster these words into their corresponding topics or aspects.

## 3 The Overview of Our Method

In this section, we present the main framework of our method. As mentioned in Section 1, we regard extracting opinion targets/words as a co-ranking process. We assume that all nouns/noun phrases in sentences are opinion target candidates, and all adjectives/verbs are regarded as potential opinion words, which are widely adopted by previous methods [4], [5], [7], [8]. Each candidate will be assigned a confidence, and candidates with higher confidence than a threshold are extracted as the opinion targets or opinion words. To assign a confidence to each candidate, our basic motivation is as follows.

> *If a word is likely to be an opinion word, the nouns/noun phrases with which that word has a modified relation will have higher confidence as opinion targets. If a noun/noun phrase is an opinion target, the word that modifies it will be highly likely to be an opinion word.*

We can see that the confidence of a candidate (opinion target or opinion word) is collectively determined by its neighbors according to the opinion associations among them. Simultaneously, each candidate may influence its neighbors. This is an iterative reinforcement process. To model this process, we construct a bipartite undirected graph $G = (V, E, W)$, named as *Opinion Relation Graph*. In $G$, $V = V^t \cup V^o$ denotes the set of vertices, of which there are two types: $v^t \in V^t$ denote opinion target candidates (the white nodes in Fig. 3) and $v^o \in V^o$ denote opinion word candidates (the gray nodes in Fig. 3). $E$ is the edge set of the graph, where $e_{ij} \in E$ means that there is an opinion relation between two vertices. It is worth noting that the edges $e_{ij}$ only exist between $v^t$ and $v^o$ and there is no edge between the two of the same types of vertices. $w_{ij} \in W$ means the weight of the edge $e_{ij}$, which reflects the opinion association between these two vertices.

Based on our *Opinion Relation Graph*, we propose a graph-based co-ranking algorithm to estimate the confidence of each candidate. Briefly, there are two important problems: 1) how to capture the opinion relations ($e_{ij} \in E$) and calculate the opinion associations between opinion targets and opinion words ($w_{ij} \in W$); 2) how to estimate the confidence of each candidate with graph co-ranking.

For the first problem, we adopt a monolingual word alignment model to capture opinion relations in sentences. A noun/noun phrase can find its modifier through word alignment. We additionally employ a partially-supervised word alignment model, which performs word alignment in a partially supervised framework. After that, we obtain a large number of word pairs, each of which is composed of a noun/noun phrase and its modifier. We then calculate associations between opinion target candidates and opinion word candidates as the weights on the edges.

For the second problem, we exploit a random walking with restart algorithm to propagate confidence among candidates and estimate the confidence of each candidate on *Opinion Relation Graph*. More specifically, we penalize the high-degree vertices according to the vertices' entropies and incorporate the candidates' prior knowledge. In this way, extraction precision can be improved.

## 4 Capturing Opinion Relations between Opinion Targets and Opinion Words Using the Word Alignment Model

### 4.1 Word Alignment Model

As mentioned in the above section, we formulate opinion relation identification as a word alignment process. We employ the word-based alignment model [23] to perform monolingual word alignment, which has been widely used in many tasks such as collocation extraction [24] and tag suggestion [25]. In practice, every sentence is replicated to generate a parallel corpus. A bilingual word alignment algorithm is applied to the monolingual scenario to align a noun/noun phase (potential opinion targets) with its modifiers (potential opinion words) in sentences.

Formally, given a sentence with $n$ words $S = \{w_1, w_2, \ldots, w_n\}$, the word alignment $A = \{(i, a_i) \mid i \in [1, n], a_i \in [1, n]\}$ can be obtained as

$$A^* = \underset{A}{argmax}\, P(A \mid S), \tag{1}$$

where $(i, a_i)$ means that a noun/noun phrase at position $i$ is aligned with its modifier at position $a_i$. There are several word alignment models for usage, such as IBM-1, IBM-2 and IBM-3 [23]. We select IBM-3 model in our task, which has been proven to perform better than other models for our task [4]. Thus, we have

$$P_{ibm3}(A \mid S) \propto \prod_{i=1}^{n} n(\phi_i \mid w_i) \prod_{j=1}^{n} t(w_j \mid w_{a_j}) d(j \mid a_j, n), \tag{2}$$

where there are three main factors $t(w_j \mid w_{a_j})$, $d(j \mid a_j, n)$ and $n(\phi_i \mid w_i)$ that model different information to indicate the opinion relations among words.

$t(w_j \mid w_{a_j})$ models the co-occurrence information of two words in corpora. If a word frequently modifies a noun (noun phrase), they will have a higher value of $t(w_j \mid w_{a_j})$. For example, in reviews of cell phone, "*big*" often co-occurs with "*phone's size*"; therefore, "*big*" has high association with "*phone's size*".

$d(j \mid a_j, n)$ models word position information, which describes the probability that a word in position $a_j$ is aligned with a word in position $j$.

$n(\phi_i \mid w_i)$ describes the ability of a word for "one-to-many" relation, which means that a word can modify (or be

modified by) several words. $\phi_i$ denotes the number of words that are aligned with $w_i$. For example,

"*Iphone4 has an amazing screen and software*".

In this sentence, "*amazing*" is used to modify two words: "*screen*" and "*software*". Thus, $\phi$ equals to 2 for "*amazing*".

**Algorithm 1.** Constrained Hill-Climbing Algorithm.

**Input:** Review sentences $S_i = \{w_1, w_2, \ldots, w_n\}$
**Output:** The calculated alignment $\hat{a}$ for sentences

**Initialization:** Calculate the seed alignment $a_0$ orderly using simple model (IBM-1, IBM-2, HMM)
**Step 1**: Optimize toward the constraints
**while** $N_{ill}(\hat{a}) > 0$ **do**
  **if** $\{a: N_{ill}(a) < N_{ill}(\hat{a})\}$=∅ then
  break
  $\hat{a} = argmax_{a \in nb(\hat{a})} Pro(f|e, a)$
**end**
**Step 2**: Toward the optimal alignment under the constraint
**for** $i < N$ *and* $j < N$ **do**
  $M_{i,j} = -1$, **if** $(i, j) \notin \hat{A}$;
**end**
**while** $M_{i_1,j_1} > 1$ *or* $S_{j_1,j_2} > 1$ **do**
  **If** $(j_1, a_{j_2}) \notin \hat{A}$ *or* $(j_2, a_{j_1}) \notin \hat{A}$ **then**
    $S_{j_1,j_2} = -1$
  **end**
  $M_{i_1,j_1} = \arg\max M_{i,j}$
  $S_{j_1,j_2} = \arg\max S_{i,j}$
  **If** $M_{i_1,j_1} > S_{j_1,j_2}$ **then**
    Update $M_{i_1,*}$,$M_{j_1,*}$,$M_{*,i_1}$,$M_{*,j_1}$
    Update $S_{i_1,*}$,$S_{j_1,*}$,$S_{*,i_1}$,$S_{*,j_1}$
    set $\hat{a} := M_{i_1,j_1}(a)$
  **end**
  **else**
    Update $M_{i_1,*}$,$M_{j_2,*}$,$M_{*,i_1}$,$M_{*,j_2}$
    Update $S_{j_2,*}$,$S_{j_1,*}$,$S_{*,j_2}$,$S_{*,j_1}$
    set $\hat{a} := S_{j_1,j_2}(a)$
  **end**
**end**
**return** $\hat{a}$;

Notably, if we are to directly apply the standard alignment model to our task, an opinion target candidate (noun/noun phrase) may align with the irrelevant words rather than potential opinion words (adjectives/verbs), such as prepositions and conjunctions. Thus, we introduce some constraints in the alignment model as follows:

1) Nouns/noun phrases (adjectives/verbs) must be aligned with adjectives/verbs (nouns/noun phrases) or a null word. Aligning to a null word means that this word either has no modifier or modifies nothing;
2) Other unrelated words, such as prepositions, conjunctions and adverbs, can only align with themselves.

According to these constraints, for the sentence in Fig. 1, we obtain the following alignment results shown in Fig. 4, where "*NULL*" means the null word. From this example, we can see that unrelated words, such as "*This*", "*a*" and "*and*", are aligned with themselves. There are no opinion words to modify "*Phone*" and "*has*" modifies nothing; therefore, these two words may align with "*NULL*".

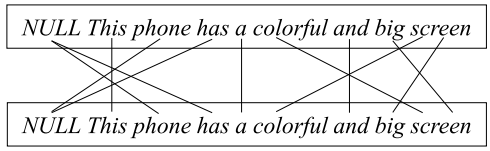



Fig. 4. Mining opinion relations between words using the word alignment model under constrains.

To obtain the optimal alignments in sentences, we adopt an EM-based algorithm [23] to train the model. Specifically, for training the IBM-3 model, the simpler models (IBM-1, IBM-2 and HMM) are sequentially trained as the initial alignments for the subsequent model. Next, the hill-climbing algorithm, a greedy algorithm, is used to find a local optimal alignment.

### 4.2 Partially-Supervised Word Alignment Model

As mentioned in the first section, the standard word alignment model is usually trained in a completely unsupervised manner, which may not obtain precise alignment results. Thus, to improve alignment performance, we perform a partial supervision on the statistic model and employ a partially-supervised alignment model to incorporate partial alignment links into the alignment process. Here, the partial alignment links are regarded as constraints for the trained alignment model. Formally, given the partial alignment links $\hat{A} = \{(i, a_i)|i \in [1, n], a_i \in [1, n]\}$, the optimal alignment $A^*$ in Eq. (1) is rewritten as follows:

$$A^* = \underset{A}{argmax}\, P(A \mid S, \hat{A}). \tag{3}$$

#### 4.2.1 Parameter Estimation for the PSWAM

Unlike the unsupervised word alignment model, the alignments generated by the PSWAM must be as consistent as possible with the labeled partial alignments. To fulfill this aim, we adopt an EM-based algorithm. For training a simpler alignment model, such as the IBM-1 and IBM-2 models, we easily obtain all possible alignments from the observed data. Those inconsistent alignments with pre-provided partial alignment links (illegal alignments) could be filtered out; therefore, they would not be counted for parameter estimation in subsequent iterations. However, in this paper, we select a more complex alignment model, the IBM-3 model, which is a fertility-based model. As mentioned in [26], for training IBM-3 model, it is NP-complete and impossible to enumerate all potential alignments. This indicates that the standard EM training algorithm is time consuming and impractical. To resolve this problem, GIZA++ provides a hill-climbing algorithm, which is a local optimal solution to accelerate the training process. In practice, GIZA++ first sequentially trains the simple models (IBM-1, IBM-2, HMM) as the initial alignments for the IBM-3 model. Next, a greedy search algorithm is used to find the optimal alignments iteratively. The search space for the optimal alignment is constrained on the "neighbor alignments" of the current alignment, where "neighbor alignments" denote the

alignments that could be generated from the current alignment by one of the following operators:

1) MOVE operator $m_{i,j}$, which changes $a_j = i$.

2) SWAP operator $s_{j_1,j_2}$, which exchanges $a_{j_1}$ and $a_{j_2}$.

In practice, GIZA++ creates two matrices, called the MOVE matrix $M$ and the SWAP matrix $S$, to record all possible MOVE or SWAP costs, respectively, between two different alignments. These operation costs are calculated as follows:

$$M_{ij} = \frac{Pr(m_{i,j}(a) \mid e, f)}{Pr(a \mid e, f)}(1 - \delta(a_j, i)),$$

$$S_{j_1,j_2} = \begin{cases} \frac{Pr(s_{j_1,j_2}(a) \mid e,f)}{Pr(a \mid e,f)}(1 - \delta(a_{j_1}, a_{j_2}))\ if\ a_{j_1} < a_{j_2}, \\ 0, \qquad otherwise. \end{cases}$$

After obtaining the optimal alignment from neighbor alignments, the next search is started in the neighbors of the current optimal alignment. At the same time, the operation cost values in $M$ and $S$ are also updated. The algorithm does not end until no new optimal alignment is found. Additionally, the statistics of the neighbor alignments of the final optimal alignment are counted for calculating the parameters.

Under partial supervision, to make the trained alignments consistent with the pre-provided partial alignments, we set illegal operation costs in $M$ and $S$ to $-1$. In this way, those inconsistent alignments would never be picked up. In general, using the given labeled partial alignments, we employ a variation of the hill-climbing algorithm mentioned above, named as the constrained hill-climbing algorithm [26], to estimate the parameters. The details of this algorithm are shown in Algorithm 1. In the training process, the constrained hill-climbing algorithm ensures that the final model is marginalized on the partial alignment links. More specifically, there are two primary steps involved.

*1) Optimize toward the constraints*. This step aims to generate an initial alignment for our alignment model close to the constraints. First, the simpler alignment models (IBM-1, IBM-2, HMM etc.) are sequentially trained. Second, evidence that is inconsistent with the partial alignment links is eliminated by using the MOVE operator $m_{i,j}$ and the SWAP operator $s_{j_1,j_2}$. Third, the alignment is updated iteratively until no additional inconsistent links can be removed (lines 2-7 in Algorithm 1), where $nb(\cdot)$ denotes the neighbor alignments and $N_{ill}(\cdot)$ denotes the total number of inconsistent links in the current alignment.

*2) Towards the optimal alignment under the constraints*. This step aims to optimize towards the optimal alignment under the constraints that start from the aforementioned initial alignments. Gao et al. [26] set the corresponding cost value of the invalid move or swap operation in $M$ and $S$ as negative. In this way, the invalid operators are never chosen, which guarantees that the final alignment links have a high probability of being consistent with the pre-provided partial alignment links (lines 8-28 in Algorithm 1), where $\hat{a}$ means the final optimal alignment and $\hat{A}$ means the provided set of partial alignment links.

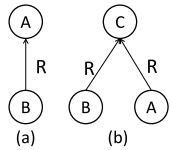



Fig. 5. The types of the used syntactic patterns.

In the M-step, evidence from the neighbors of final alignments is collected so that we can produce the estimation of parameters for the next iteration. In the process, those statistics that came from inconsistent alignment links are not to be picked up. Thus,

$$P(w_i \mid w_{a_i}) = \begin{cases} \lambda, & A\ \ is\ inconsistent\ with\ \hat{A}, \\ P(w_i \mid w_{a_i}) + \lambda, & otherwise, \end{cases} \tag{4}$$

where $\lambda$ is a smoothing factor, which means that we make the soft constraints on the alignment model, and that some incorrect partial alignment links generated through high-precision patterns (Section 4.2.2) may be revised. Next, we perform count collections and normalize to produce the model parameters for the next iteration.

#### 4.2.2 Obtaining Partial Alignment Links by Using High-Precision Syntactic Patterns

For training the PSWAM, the other important issue is to obtain the partial alignment links. Naturally, we can resort to manual labeling. However, this strategy is both time consuming and impractical for multiple domains. We need an automatic method for partial alignment generation. To fulfill this aim, we resort to syntactic parsing. As mentioned in the first section, although current syntactic parsing tools cannot obtain the whole correct syntactic tree of informal sentences, some short or direct syntactic relations can be still obtained precisely. Thus, some high-precision-low-recall syntactic patterns are designed to capture the opinion relations among words for initially generating the partial alignment links. These initial links are then fed into the alignment model.

To guarantee that the used syntactic patterns are high precision, we use the constraint that the syntactic patterns are based solely on the direct dependency relations defined in [7]. A direct dependency indicates that one word depends on the other word without any additional words in their dependency path or that these two words both directly depend on a third word. As shown on the left side ((a) and (b)) of Fig. 5, $A$ directly depends on $B$ in (a), and $A$ and $B$ both directly depend on $C$ in (b). Qiu et al. [7] also defined some indirect dependency relations. We do not use them because introducing indirect dependency relations may decrease the precision. Specifically, we employ the Minipar[1] as the English sentence parser, which was also used in [7]. For Chinese sentences, we employ the Stanford Parser.[2] The right side of Fig. 5 shows the utilized syntactic pattern types corresponding to two direct dependency relation types. In Fig. 5, $A$ and $B$ denote a potential opinion word ($OC$) or a potential opinion target ($TC$). Moreover, in

1. http://webdocs.cs.ualberta.ca/lindek/minipar.htm
2. http://nlp.stanford.edu/software/lex-parser.shtml

TABLE 1
Some Examples of Used Syntactic Patterns

| |
|---|
| Pattern#1: <OC> $\xrightarrow{mod}$ <TC> |
| Example: This phone has an **amazing design** |
| Pattern#2: <OC> $\xrightarrow{pnmod}$ <TC> |
| Example: the **buttons easier** to use |
| Pattern#3: <OC> $\xrightarrow{rcmod}$ <TC> |
| Example: **漂亮** 的 **外观** (**beautiful design**). |
| Pattern#4: <OC> $\xrightarrow{nsbuj}$ <TC> |
| Example: 这款 **手机** **不错** (This **phone** is **good**) |
| Pattern#5: <OC> $\xrightarrow{mod}$ (NN) $\xleftarrow{subj}$ <TC> |
| Example: **IPhone** is a **revolutionary** smart phone |
| Pattern#6: <OC> $\xrightarrow{pnmod}$ (NN) $\xleftarrow{subj}$ <TC> |
| Example: **S3** is the phone **cheaper** to obtain. |

(b) of Fig. 5, $A$ and $B$ both depend on the other word $C$, where $C$ is any word. In addition, to obtain precise alignment links, in our patterns, we constrain the dependency relation labels output by the syntactic parser in $R$, i.e., $R \in \{mod, pnmod, subj, s\}$ for the Minipar, and $R \in \{amod, rcmod, nsubjpass, nsubj\}$ for the Stanford Parser. For clarity, we provide some syntactic pattern examples in Table 1, where the first four patterns belong to the direct dependency type (a) and the last two patterns belong to the direct dependency type (b).

### 4.3 Calculating the Opinion Associations Among Words

From the alignment results, we obtain a set of word pairs, each of which is composed of a noun/noun phrase (opinion target candidate) and its corresponding modified word (opinion word candidate). Next, the alignment probabilities between a potential opinion target $w_t$ and a potential opinion word $w_o$ are estimated using

$$P(w_t \mid w_o) = \frac{Count(w_t, w_o)}{Count(w_o)},$$

where $P(w_t \mid w_o)$ means the alignment probability between these two words. Similarly, we obtain the alignment probability $P(w_o \mid w_t)$ by changing the alignment direction in the alignment process. Next, we use the score function in [24] and [25] to calculate the opinion association $OA(w_t, w_o)$ between $w_t$ and $w_o$ as follows:

$$OA(w_t, w_o) = (\alpha * P(w_t \mid w_o) + (1 - \alpha) P(w_o \mid w_t))^{-1}, \quad (5)$$

where $\alpha$ is the harmonic factor used to combine these two alignment probabilities. In this paper, we set $\alpha = 0.5$.

## 5 Estimating Candidate Confidence with Graph Co-Ranking

After mining the opinion associations between opinion target candidates and opinion word candidates, we complete the construction of the *Opinion Relation Graph*. We then calculate the confidence of each opinion target/word candidate on this graph, and the candidates with higher confidence than a threshold are extracted as opinion targets or opinion words. We assume that two candidates are likely to belong to a similar category if they are modified by similar opinion words or modify similar opinion targets. If we know one of them to be an opinion target/word, the other one has a high probability of being an opinion target/word. Thus, we can forward the confidences among different candidates, which indicates that the graph-based algorithms are applicable.

### 5.1 Estimating Candidate Confidence by Using Random Walking

Naturally, we can use a standard random walk with restart algorithm to estimate the confidence of each candidate. Thus, we have

$$\begin{aligned} C_t^{k+1} &= (1 - \mu) \times M_{to} \times C_o^k + \mu \times I_t, \\ C_o^{k+1} &= (1 - \mu) \times M_{to}^T \times C_t^k + \mu \times I_o, \end{aligned} \quad (6)$$

where $C_t^{k+1}$ and $C_o^{k+1}$ are the confidence of an opinion target candidate and opinion word candidate, respectively, in the $k+1$ iteration. $C_t^k$ and $C_o^k$ are the confidence of an opinion target candidate and opinion word candidate, respectively, in the $k$ iteration. $M_{to}$ records opinion associations among candidates. $m_{ij} \in M_{to}$ means the opinion association between the ith opinion target candidate and the $j$-th opinion word candidate, which can be computed by using Eq. (5). In Eq. (6), we can see that $C_t^{k+1}$ and $C_o^{k+1}$ are determined by two parts. One is $M_{to} \times C_o^k$ and $M_{to}^T \times C_t^k$, which mean that the confidence of an opinion target (opinion word) candidate is obtained through aggregating confidences of all neighboring opinion word (opinion target) candidates together according to their opinion associations. The other ones are $I_t$ and $I_o$, which denote prior knowledge of candidates being opinion targets and opinion words, respectively. Section 5.3 will describe how to calculate them in detail. $\mu \in [0, 1]$ means the impact of prior knowledge on the final results. When $\mu = 1$, candidate confidence is completely determined by prior knowledge; and when $\mu = 0$, candidate confidence is determined by candidate opinion relevance.

### 5.2 Penalizing on High-Degree Vertices

Notwithstanding the above, we observe that the standard random walk algorithm (Eq. (6)) could be dominated by high-degree vertices, which may introduce noise. As high-degree vertices link with more vertices, these high-degree vertices are prone to collecting more information from the neighbors and have a significant impact on other vertices when performing random walks. If a vertex connects with a high-degree vertex, it would have a larger possibility to be reached by a walker. In review texts, these high-degree vertices usually represent *general words*. For example, "*good*" may be used to modify multiple objects, such as "*good design*", "*good feeling*" and "*good things*". "*Good*" is a *general word*, and its degree in the *Opinion Relation Graph* is high. If we know that "*design*" has higher confidence to be an opinion target, its confidence will be propagated to "*feeling*" and "*thing*" through "*good*". As a result, "*feeling*" and "*thing*" most likely have higher confidence as opinion targets. This is unreasonable. Meanwhile, the same problem may occur in opinion word extraction. To resolve this problem, we are required to penalize these high-degree vertices to weaken

their impact and decrease the probability of the random walk running into the unrelated regions [27], [28]. In this way, errors may be avoided as much as possible. In practice, we employ information entropy to measure the degree of the vertices. Based on this information, we automatically adjust the weight of each vertex in the graph.

Specifically, when the random walk reaches a vertex $v$, we believe that there are three choices for the walker: (a) continue the random walk to the neighbors of $v$, (b) abandon the random walk or (c) stop the walk and emit a confidence according to prior knowledge. We assume that the probabilities of these three events are $P_{con}(v)$, $P_{abnd}(v)$ and $P_{inj}(v)$, respectively. Thus, the co-ranking algorithm in Eq. (6) is rewritten as follows:

$$\begin{aligned} C_t^{i+1} &= P_{con}(t) \times M_{to} \times C_o^i + P_{inj}(t) \times I_t \\ &\quad + P_{abnd}(t) \times I_\phi, \\ C_o^{i+1} &= P_{con}(o) \times M_{to}^T \times C_t^i + P_{inj}(o) \times I_o \\ &\quad + P_{abnd}(o) \times I_\phi, \end{aligned} \tag{7}$$

where $C_t^{i+1}$, $C_o^{i+1}$, $C_o^i$, $C_t^i$, $M_{to}$, $I_t$ and $I_o$ have the same means in Eq. (6). Additionally, $I_\phi$ represents a lack of information about the opinion target/word confidence of the vertex $v$, and we set the value of all cells in $I_\phi$ to 1. $P_{abnd}(\cdot)$ is used to mitigate the effect of the transition into unrelated regions on the graph when reaching high-degree vertices [28]. To penalize high-degree vertices, we adopt the heuristics from [28]. For each vertex $v$, let

$$c_v = \frac{log(\beta)}{log(\beta + exp(H(v)))},$$

where $\beta = 2$ and $H(v) = -\sum_u p_{uv} \times log(p_{uv})$ denotes the entropy of vertex $v$. $p_{uv} = \frac{W(u,v)}{\sum_u W(u,v)}$. Meanwhile, we have $j_v = (1 - c_v) \times \sqrt{H(v)}$ and $z_v = max(c_v + j_v, 1)$. We then set

$$P_{inj}(v) = \frac{j_v}{z_v}, \quad P_{con}(v) = \frac{c_v}{z_v},$$

$$P_{abnd}(v) = 1 - P_{con}(v) - P_{inj}(v).$$

If $v$ has a higher degree, $H(v)$ will be larger and $c_v$ will be lower. Thus, $P_{con}(v)$ will have a lower value. Accordingly, the contribution of high-degree vertices is restricted.

The algorithm is run until convergence, which is achieved when the confidence on each vertex ceases to change within a tolerance value. Finally, candidates with higher confidence are extracted as opinion targets or opinion words.

### 5.3 Calculating Candidate Prior Knowledge

Candidate prior knowledge ($I_t$ and $I_o$ in Eq. (6) and Eq. (7)) is important for estimating each candidate's confidence. We notice that users usually express opinions on some unrelated objects in reviews, such as "good feelings", "wonderful time" and "bad mood". Obviously, "feelings", "time" and "mood" are not real opinion targets. However, because they occur frequently and are modified by real opinion words ("good", "wonderful" and "bad", etc.), only employing opinion relations could not filter them out. Therefore, we need to assign these unrelated objects low confidence as prior knowledge ($I_t$ and $I_o$) and incorporate them in our co-ranking algorithm. In this way, confidence estimation would be more precise. In detail, we employ different strategies to calculate $I_t$ and $I_o$ as follows:

*Calculating the prior confidences of opinion target candidates*. To calculate $I_t$, [4] and [29] used a TF-IDF like measure. They believe that if a candidate is frequently mentioned in reviews, it is likely to be an opinion target. However, those false opinion targets may occur frequently in reviews and will have high TF-IDF scores. Thus, using TF-IDF scores as the prior knowledge will not result in the expected performance. To address this issue, we resort to exterior resources. We notice that a large proportion of these false opinion targets ("feelings", "time" and "mood") are not domain-specific words and occur frequently in common texts. Therefore, we generate a small domain-independent general noun (GN) corpus from a large web corpora to cover some of the most frequently occurring noises. Specifically, we extract the 1,000 most frequent nouns in Google's n-gram corpus.[3] In addition, we add all the nouns in the top three levels of hyponyms in four WordNet (Miller, 1995) synsets "object", "person", "group" and "measure" into the GN corpus. Our intuition is based on the fact that a term is more general when it occurs at higher levels in the WordNet hierarchy. For Chinese, we generate general nouns in a similar way from HowNet [30]. In total, 3,071 English words and 3,493 Chinese words are selected for the GN corpus.

Employing the GN corpus merely covers a portion of noises. To increase the coverage, we use a machine learning based technique. We measure the possibility of a candidate being an opinion target from several views rather than only using a TF-IDF score. A linear regression model is then constructed to combine these multiple possibilities together for a candidate prior confidence calculation.

To generate training data for the model, we first calculate the TF-IDF score for each candidate $v$, i.e., $\frac{tf(v)idf(v)}{\sum_v tf(v)idf(v)}$. $tf(v)$ is the frequency of $v$ in the corpus. $idf(v)$ is the inverse document frequency and is computed by using Google's n-gram corpus. Next, the top $N$ target candidates with larger TF-IDF scores but that are not in our GN corpus are regarded as the positive instances, and their confidences are set to 1. The top $N$ target candidates with higher TF-IDF scores that occur in the GN list are selected as the negative instances and their prior confidences are set to 0. In the experiments, we set $N = 50$. Next, we use the following features to represent candidates:

1) *Salience feature*. This feature indicates the salience degree of a candidate in reviews. Similar to [4], we use the TF-IDF to measure the salience value of each candidate .
2) *Domain relevance feature*. We observe that opinion targets are usually domain-specific, and there are remarkable distribution differences between them in different domains (in-domain $D_{in}$ versus out-domain $D_{out}$). Thus, we use a domain relevance ratio proposed

3. http://books.google.com/ngrams/datasets

in [31], $R(v) = \frac{R(v,D_{in})}{R(v,D_{out})}$, to calculate feature values, where $R(v, D)$ means a domain relevance measure between candidate $v$ and domain $D$, and detailed information was described in [31]. We use the given reviews as the in-domain collection $D_{in}$ and Google's n-gram corpus as the out-domain collection $D_{out}$.

3) *Lexical feature*. For each candidate, all words having opinion relations with it are selected as lexical features. Feature values are computed using Eq. (5).

Next, we used a linear regression model to estimate the prior confidence of each candidate

$$PriorScore(c_i^t) = \theta^T \cdot \Phi(c_i^t),$$

where $PriorScore(c_i^t)$ denotes the estimated prior confidence of the $i$th opinion target candidate, $\Phi(c_i^t)$ is the feature vector and $\theta$ is the corresponding feature weight vector. However, the number of aforementioned generated labeled instances is small for training a robust regression model. Thus, we use the semi-supervised algorithm [32], which can incorporate unlabeled candidates into the training process and would have better generalization, to train our regression model. Finally, the regressed value $PriorScore(c_i^t)$ is set as the corresponding entry in $I_t$. To avoid negative values, we set all negative values in $I_t$ to zero.

*Estimating the prior confidences of opinion word candidates*. In contrast with opinion targets, opinion words are complicated. Some opinion words are domain independent. In different domains, users may use the same words to express their opinions, such as "*good*", "*bad*" and "*sad*". In addition, some opinion words are domain dependent, such as "*delicious*" in restaurant reviews or "*powerful*" in auto reviews. It is difficult to generate a corpus to flag false opinion words by means of frequency statistic or exterior resources such as opinion target prior knowledge calculations. As a result, negative labeled instances for training a regression model is missing. Thus, we simply make use of exterior manual labeled resources such as SentiWordNet[4] and Hownet Sentiment Word Collection[5] (HSWC) to flag a portion of correct opinion words. For English candidates, if a candidate is in SentiWordNet, its prior confidence value in $I_o$ is the subjective score ($PosScore + NegScore$) annotated in SentiWordNet; otherwise, it is 0. For Chinese candidates, if a candidate is in HSWC, its prior confidence value in $I_o$ is 1; otherwise, it is 0.

## 6 Experiments

### 6.1 Data Sets and Evaluation Metrics

We select three datasets to evaluate our approach. The first dataset is the *Customer Review Datasets* (*CRD*), which includes English reviews of five products. *CRD* was also used in [5], [7]. The second dataset is *COAE 2008 dataset2*,[6] which contains Chinese reviews of four types of products: cameras, cars, laptops and phones. The third dataset is *Large*, which includes three corpora with different languages from three domains including hotels, mp3s and restaurants. For each domain in Large, we randomly crawl 6,000 sentences. Additionally, the opinion targets and opinion words in Large were manually annotated as the gold standard for evaluations. Three annotators are involved in the annotation process. Two annotators were required to judge whether every noun/noun phrase (adjectives/verbs) is an opinion target (opinion word) or not. If a conflict occurred, a third annotator makes a judgment for the final results. The inter-agreement was 0.72 for opinion target annotation and 0.75 for opinion word annotation. Statistical information of each dataset is shown in Table 2, where #OW and #OT stand for the numbers of annotated opinion words and opinion targets, respectively.

TABLE 2
The Detailed Information of Data Sets

| Datset | Domain | Language | #Sentence | #OW | #OT |
|---|---|---|---|---|---|
| **Large** | **Restaurant** | Chinese | 6,000 | 451 | 949 |
| | **Hotel** | English | 6,000 | 398 | 872 |
| | **MP3** | English | 6,000 | 503 | 924 |
| **CRD** | **D1** | English | 597 | 175 | 109 |
| | **D2** | English | 346 | 182 | 98 |
| | **D3** | English | 546 | 261 | 177 |
| | **D4** | English | 1,716 | 138 | 73 |
| | **D5** | English | 740 | 164 | 103 |
| **COAE 2008** | **Camera** | Chinese | 2075 | 351 | 892 |
| | **Car** | Chinese | 4,783 | 622 | 1,179 |
| | **Laptop** | Chinese | 1,034 | 475 | 518 |
| | **Phone** | Chinese | 2,644 | 538 | 1,125 |

In the experiments, reviews are first segmented into sentences according to punctuation. Next, sentences are tokenized, with part-of-speech tagged using the Stanford NLP tool.[7] We then use the Minipar toolkit to parse English sentences and the Stanford Parsing tool to parse Chinese sentences. The method in [33] is used to identify noun phrases. We select precision ($P$), recall ($R$) and F-measure ($F$) as the evaluation metrics.

### 6.2 Our Methods versus State-of-the-art Methods

For comparison, we select the following methods as baselines.

- **Hu** is the method described in [5]. It used nearest neighbor rules to identify opinion relations among words. Opinion targets and opinion words are then extracted iteratively using a bootstrapping process.
- **DP** is the method proposed by [7]. They designed several syntax-based patterns to capture opinion relations in sentences, and used a bootstrapping algorithm (called Double Propagation) to extract opinion targets and opinion words.
- **Zhang** is the method proposed by [3]. It is an extension of DP. Besides the syntactic patterns used in DP, Zhang designed some heuristic patterns to indicate opinion target candidates. An HITS [18] algorithm combined with candidate frequency is then employed to extract opinion targets.
- **OursWAM** uses an unsupervised word alignment model (described in Section 4.1) to mine the associations between words. A standard random walk

4. http://sentiwordnet.isti.cnr.it/
5. http://www.keenage.com/html/c_index.html
6. http://ir-china.org.cn/coae2008.html
7. http://nlp.stanford.edu/software/tagger.shtml

TABLE 3
Experimental Results of Opinion Target Extraction on *Large* and *COAE 2008*

| Methods | Hotel | | | MP3 | | | Restaurant | | | Camera | | | Car | | | Laptop | | | Phone | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | P | R | F | P | R | F | P | R | F | P | R | F | P | R | F | P | R | F | P | R | F |
| **Hu** | 0.60 | 0.65 | 0.62 | 0.61 | 0.68 | 0.64 | 0.64 | 0.69 | 0.66 | 0.63 | 0.65 | 0.64 | 0.62 | 0.58 | 0.60 | 0.51 | 0.67 | 0.58 | 0.69 | 0.60 | 0.64 |
| **DP** | 0.67 | 0.69 | 0.68 | 0.69 | 0.70 | 0.69 | 0.74 | 0.72 | 0.73 | 0.71 | 0.70 | 0.70 | 0.72 | 0.65 | 0.68 | 0.58 | 0.69 | 0.63 | 0.78 | 0.66 | 0.72 |
| **Zhang** | 0.67 | 0.76 | 0.71 | 0.67 | 0.77 | 0.72 | 0.75 | 0.79 | 0.77 | 0.71 | 0.78 | 0.74 | 0.69 | 0.68 | 0.68 | 0.57 | 0.80 | 0.67 | 0.80 | 0.71 | 0.75 |
| **OursWAM** | 0.73 | 0.82 | 0.77 | 0.72 | 0.84 | 0.78 | 0.81 | 0.84 | 0.82 | 0.78 | **0.82** | **0.80** | 0.71 | **0.72** | 0.71 | 0.65 | 0.85 | 0.74 | 0.84 | **0.75** | 0.79 |
| **OursPSWAM** | **0.78** | **0.83** | **0.80** | **0.77** | **0.85** | **0.81** | **0.86** | **0.85** | **0.85** | **0.79** | 0.81 | **0.80** | **0.75** | **0.72** | **0.73** | **0.69** | **0.86** | **0.77** | **0.86** | **0.75** | **0.80** |

TABLE 4
Experimental Results of Opinion Target Extraction on *Customer Review Data Set*

| Methods | D1 | | | D2 | | | D3 | | | D4 | | | D5 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | P | R | F | P | R | F | P | R | F | P | R | F | P | R | 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 |
| **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 |
| **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 |
| **OursWAM** | 0.86 | **0.85** | **0.85** | 0.88 | **0.85** | **0.86** | 0.89 | 0.89 | 0.89 | 0.81 | **0.85** | **0.83** | 0.89 | **0.87** | 0.88 |
| **OursPSWAM** | **0.87** | 0.84 | **0.85** | 0.89 | 0.84 | **0.86** | **0.90** | **0.90** | **0.90** | **0.82** | 0.83 | 0.82 | **0.92** | **0.88** | **0.90** |

TABLE 5
Experimental Results of Opinion Word Extraction on *Large* and *COAE 2008*

| Methods | Hotel | | | MP3 | | | Restaurant | | | Camera | | | Car | | | Laptop | | | Phone | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | P | R | F | P | R | F | P | R | F | P | R | F | P | R | F | P | R | F | P | R | F |
| **Hu** | 0.50 | 0.67 | 0.57 | 0.54 | 0.68 | 0.60 | 0.69 | 0.77 | 0.73 | 0.72 | 0.74 | 0.73 | 0.7 | 0.71 | 0.70 | 0.65 | 0.7 | 0.67 | 0.69 | 0.71 | 0.70 |
| **DP** | 0.59 | 0.66 | 0.62 | 0.62 | 0.68 | 0.65 | 0.74 | 0.70 | 0.72 | 0.80 | 0.73 | 0.76 | 0.79 | 0.71 | 0.75 | 0.73 | 0.68 | 0.70 | 0.76 | 0.69 | 0.72 |
| **OursWAM** | 0.60 | 0.71 | 0.65 | 0.64 | **0.75** | **0.69** | 0.73 | **0.79** | 0.76 | 0.80 | **0.78** | 0.79 | 0.79 | **0.78** | 0.78 | 0.74 | 0.75 | **0.74** | 0.80 | **0.77** | 0.78 |
| **OursPSWAM** | **0.64** | **0.72** | **0.68** | **0.66** | 0.73 | **0.69** | **0.79** | 0.77 | **0.78** | **0.84** | 0.77 | **0.80** | **0.82** | 0.76 | **0.79** | **0.79** | **0.76** | **0.77** | **0.83** | **0.77** | **0.80** |

TABLE 6
Experimental Results of Opinion Word Extraction on *Customer Review Data Set*

| Methods | D1 | | | D2 | | | D3 | | | D4 | | | D5 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | P | R | F | P | R | F | P | R | F | P | R | F | P | R | 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 |
| **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 |
| **OursWAM** | 0.62 | **0.76** | 0.68 | 0.57 | 0.79 | 0.66 | 0.63 | 0.77 | 0.69 | 0.62 | **0.71** | 0.66 | 0.70 | **0.71** | 0.70 |
| **OursPSWAM** | **0.65** | **0.76** | **0.70** | **0.59** | **0.80** | **0.68** | **0.66** | **0.78** | **0.71** | **0.64** | 0.70 | **0.67** | **0.72** | **0.71** | **0.71** |

based algorithm, described in Eq. (6), is used to estimate the candidate confidences for each candidates. Subsequently, candidates with high confidence will be extracted as opinion targets/words.

- **OursPSWAM** is the method described in this paper. It uses a partially-supervised word alignment model to mine the opinion relations between words. Next, a graph-based co-ranking algorithm (Eq. (7)) is used to extract opinion targets and opinion words.

In reviewing these comparative methods, we see that Hu represents those methods based on nearest neighbor rules, DP and Zhang represent syntax-based methods, and OursWAM and OursPSWAM represent word alignment based methods. Moreover, it is worth noting that Zhang does not extract opinion words. The patterns in his method are specially designed to extract opinion targets. Therefore, the results for opinion words are not taken into account. The parameter settings of Hu, DP and Zhang are the same as the original papers. In OursWAM and OursPSWAM, we set $\phi_{max} = 2$ when using the word alignment model to capture opinion relations among words (Eq. (2)). In OursWAM, we set $\mu = 0.3$ in Eq. (6) to indicate the impact of prior knowledge. The results of the opinion target extraction on each dataset are shown in Tables 3 and 4. The results of the opinion word extraction are shown in Tables 5 and 6. In these tables, "P" denotes precision, "R" denotes recall and "F" denotes F-measure. Significance is tested using paired $t$-test with $p < 0.05$. The underline "~" denotes statistical significance compared with the corresponding best performance of baselines (Hu, DP and Zhang). The wavy line "_" denotes the improvement made by OursPSWAM against OursWAM is statistically significant. From these tables, we make the following observations.

1) From opinion target extraction results, we see that the OursPSWAM outperforms baselines in most domains, where the differences in F-measure are statistically significant ($p < 0.05$) in the ten of the twelve domains. From the opinion word extraction results, we obtain similar observations. The differences in F-measure are statistically significant ($p < 0.05$) in all twelve domains. Those indicate the effectiveness of our method.
2) The methods based on word alignment models (OursWAM and OursPSWAM) significantly improve the performance of other baselines ($p < 0.05$ in F-measure) in most domains, except for extracting opinion targets

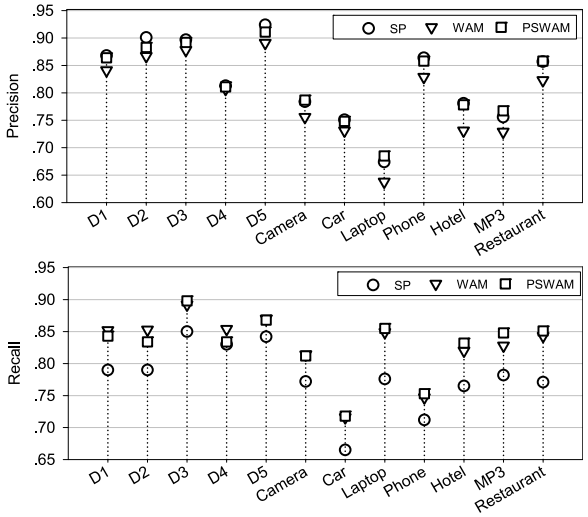



Fig. 6. Experimental comparison among different opinion relation identification methods for opinion target extraction.

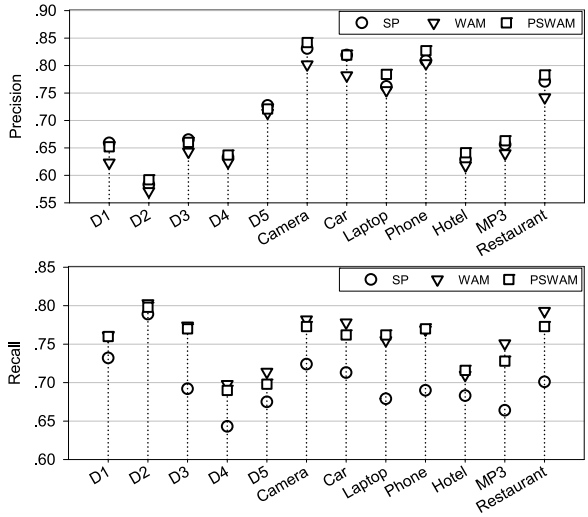



Fig. 7. Experimental comparison among different opinion relation identification methods for opinion word extraction.

in D2 and D4. Especially, they outperform syntax-based methods (DP and Zhang). We believe this is because the methods based on the word alignment model can effectively avoid parsing errors for informal texts, and more precise opinion relations among words are captured. Moreover, the syntactic patterns used in DP and Zhang are designed manually, where they cannot cover all potential opinion targets/words in reviews. In contrast, these two alignment based methods regard all nouns/noun phrases as potential opinion targets and all adjectives/verbs as opinion word candidates. Then a graph co-ranking algorithm is employed to extract the correct opinion targets/words through confidence estimation. Therefore, they have better recall.

3) The OursPSWAM outperforms the OursWAM in most domains, where the improvement in F-measure is statistically significant ($p < 0.05$) in the eight of the twelve domains for opinion target extraction and significant ($p < 0.05$) in the eight of the twelve domains for opinion word extraction. Although the recall of the OursPSWAM drops slightly compared to the OursWAM in several domains (such as MP3, Camera, Car and Restaurant in Table 5), the OursPSWAM has better precision than the OursWAM and the differences in precision are statistically significant ($p < 0.05$) in all domains. We believe there are two reasons for this. First, the OursPSWAM identifies opinion relations by performing the WAM under partial supervision. High-precision syntactic patterns are employed to obtain partial alignment links which are used as constraints for training our alignment model. This strategy is effective for improving the precision of opinion relation identification. Second, to estimate the confidence of each candidate in the graph, we penalize the high-degree vertices to decrease the probability of the random walk running into unrelated regions. In this way, some errors introduced by *general words* can be effectively alleviated.

4) In Tables 4 and 6, we observe that the F-measure improvement made by OursPSWAM compared with baselines is smaller than it in Tables 3 and 5. We argue that *CRD* is too small to provide sufficient data for training a word alignment model. *Large* is larger than *COAE 2008*, and both of these corpora are larger than *CRD*. This indicates that the proposed method is more appropriate for larger corpora.

### 6.3 Effect of the Partially-Supervised Word Alignment Model

In this subsection, we aim to prove the effectiveness of the utilized partially-supervised word alignment model for capturing opinion relations in sentences. To make a fair comparison, we select three methods: **SP**, **WAM** and **PSWAM**. The SP uses the syntactic patterns used in Section 4.2.2 to identify opinion relations in sentences. The WAM employs an unsupervised word alignment model to perform this task. The PSWAM employs a partially-supervised word alignment model to identify opinion relations in sentences. The method in Section 4.3 is then used to estimate opinion associations among words. Finally, the graph co-ranking algorithm in Eq. (7) is used to co-extract opinion targets/words. The experimental results are shown in Figs. 6 and 7.

In Fig. 6, we observe that SP has worse recall compared to alignment based methods (WAM and PSWAM) in most domains. The differences against the corresponding worse performance between WAM and PSWAM are statistically significant ($p < 0.05$) with the paired $t$-test for opinion target extraction in all domains excluding D4. These differences are significant ($p < 0.05$) for opinion word extraction in all domains excluding D2. This is because the syntactic patterns used in SP are high-precision, which can only capture a portion of the opinion relations in sentences. Only those opinion targets/words that satisfy the given syntactic patterns can be extracted. It may lose many potential opinion targets/words. The WAM and the PSWAM utilize word alignment instead of syntactic patterns to identify opinion relations among words. Accordingly, more opinion

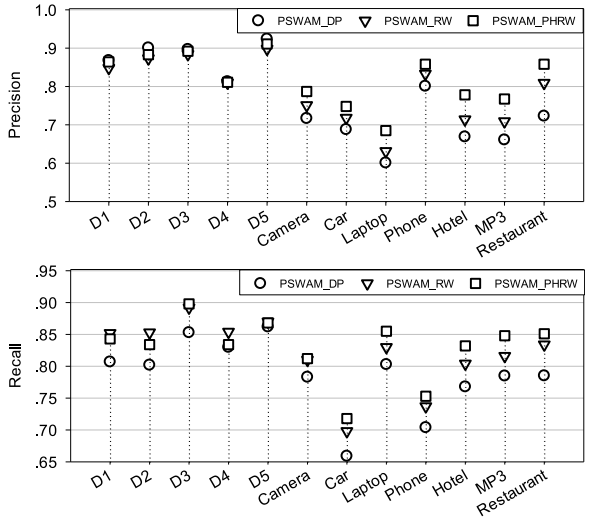



Fig. 8. Experimental comparison among different ranking methods for opinion target extraction.

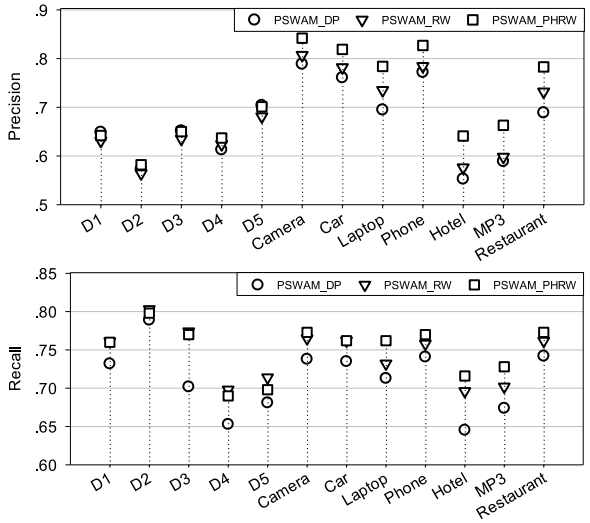



Fig. 9. Experimental Comparison among different ranking methods for opinion word extraction.

relations, rather than just the relations defined by syntactic patterns, can be obtained. Therefore, the methods based on an alignment model have better recall. Moreover, the PSWAM has better precision than the WAM. The improvement is statistically significant ($p < 0.05$) in all domains excluding D4 for extracting opinion targets and the improvement is significant ($p < 0.05$) in all domains excluding D5 for extracting opinion words. PSWAM even obtains competitive precision compared to the SP. This is because the alignment performance is improved by using partial supervision from high-precision syntactic patterns. Thus, it proves the effectiveness of our partially-supervised alignment model for opinion target/word extraction.

### 6.4 Effect of our Graph-Based Co-Ranking Algorithm

To estimate the confidence of each candidate with the graph co-ranking algorithm, we penalize the high-degree vertices to decrease the probability of a random walk running into the unrelated regions in the graph. Therefore, in this experiment, we aim to prove the effectiveness of this strategy for our tasks. We specifically design three comparative methods: **PSWAM_DP**, **PSWAM_RW** and **PSWAM_PHRW**. All of these methods use a partially-supervised alignment model to mine opinion relations between words. Next, the same method set out in Section 4.3 is used to estimate opinion associations between words. Finally, the PSWAM_DP uses the bootstrapping algorithm (Double Propagation in [7]) to extract opinion targets/words. The PSWAM_RW uses the random walk algorithm in Eq. (6) to extract opinion targets/words, where $\mu = 0.3$. The PSWAM_PHRW employs the graph-based co-ranking algorithm in Eq. (7) that penalizes high-degree vertices to estimate the candidate confidences. Figs. 8 and 9 give the experimental results.

In Figs. 8 and 9, we observe that the graph-based extraction methods (PSWAM_RW and PSWAM_PHRW) outperform the method based on a bootstrapping framework (PSWAM_DP) for opinion target/word extraction in most domains. We believe that this is because the bootstrapping-based method may have an error propagation problem. The PSWAM_RW and the PSWAM_PHRW regard extraction as a ranking task and calculate the confidence of each candidate in a unified graph through random walks; therefore, the problem of error propagation can be effectively alleviated. Moreover, the PSWAM_PHRW has better precision than the PSWAM_RW in most domains. The improvement in precision is statistically significant ($p < 0.05$) for extracting opinion targets in all domains excluding D2, D3 and D4, and the improvement is statistically significant ($p < 0.05$) for extracting opinion words in all domains excluding D1 and D4. We believe the main reason is that we penalize high-degree vertices (*general words*) in the graph according to the vertex entropy. Some errors introduced by *general words* can be filtered. Therefore, performance can be improved.

### 6.5 The Effect of Syntactic Information on the Partially Supervised Word Alignment Model

Although we have proven that using the PSWAM can effectively improve the performance of opinion target/word extraction, we are still curious about how performance varies when we incorporate different amounts of syntactic information into the PSWAM. In this experiment, we first rank the syntactic patterns mentioned in Section 4.2.2 according to the number of alignment links extracted by these patterns. The top K syntactic patterns are then selected and are incorporated into the PSWAM in Section 4.2 in turns. We define $1 \leq K \leq 7$. With larger values of K, more syntactic information is incorporated. The extraction performance variations are shown in Figs. 10 and 11, respectively.

In Fig. 10, we observe that syntactic information mainly affects precision but has little impact on recall. In Fig. 11, we make the same observation. We believe this is because feeding more partial alignments mined by high-precision syntactic patterns can effectively correct errors generated by a completely unsupervised trained word alignment model.

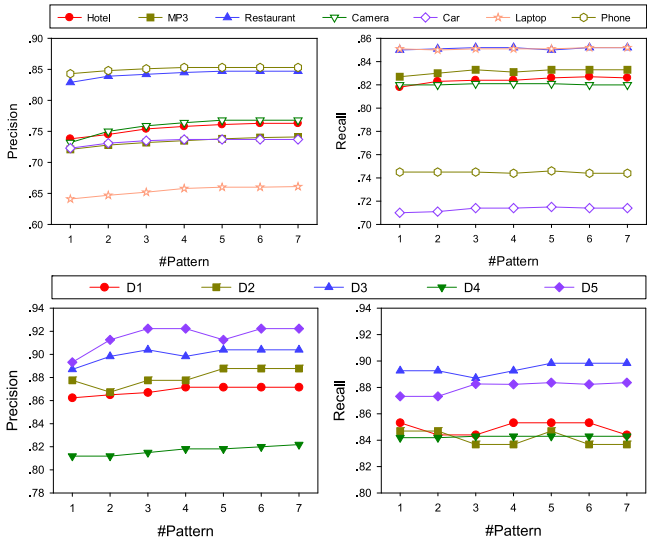



Fig. 10. The impacts of incorporating different syntactic information into word alignment model for opinion target extraction.

This further proves the effectiveness of our partially-supervised word alignment model for this task.

### 6.6 The Effect of Prior Knowledge

In this section, we discuss the effects of prior knowledge of candidates on extraction performance. In the experiments of opinion target extraction, we design four comparison methods: **NoPrior**, **Prior_TFIDF**, **Prior_Resourse** and **Prior_Learning**. The NoPrior does not incorporate any prior knowledge of candidates when estimating candidates' confidences, which is equivalent to setting $I_t$ to zero in Eq. (7). The Prior_TFIDF calculates $I_t$ using the TF-IDF score as in [4]. The Prior_Resourse uses the generated general nouns (GN) corpus (in Section 5.3) to filter the general nouns in the results. The Prior_Learning is the proposed method, which uses a semi-supervised regression model to calculate the prior confidence of candidates. Fig. 12 shows the results. From the results, we observe that the NoPrior obtains the worst precision, which demonstrates that prior knowledge is useful for opinion target extraction. We further observe that the Prior_Learning has better precision than the Prior_ TFIDF in most domains, where the differences between them are statistically significant ($p < 0.05$ ) in all domains excluding D3 and D4. It indicates that only employing the TF-IDF score is not enough to flag false opinion targets. In addition, the Prior_Learning outperforms the Prior_Resourse in precision significantly ($p < 0.05$) in all domains excluding D4. We believe it is because the learning-based approach has a better generalization than only employing exterior resources for indicating noises.

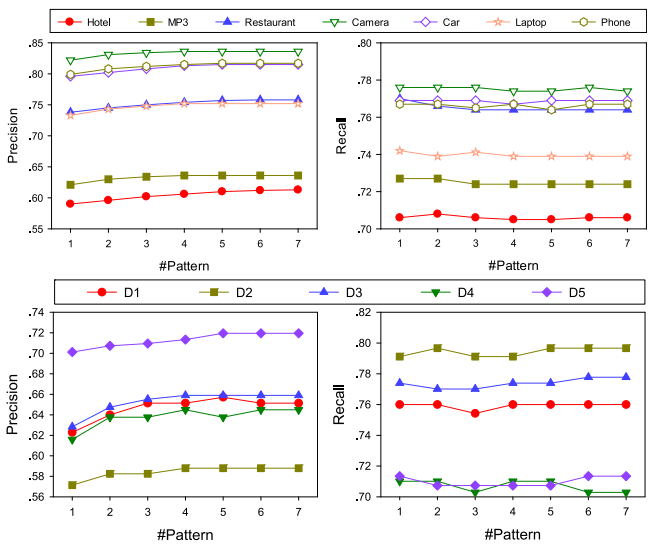



Fig. 11. The impacts of incorporating different syntactic information into word alignment model for opinion word extraction.

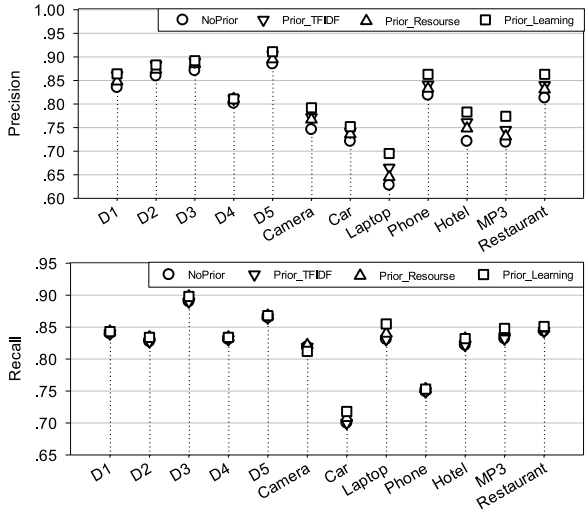



Fig. 12. Experimental comparison among different ranking methods for opinion targets extraction.

For opinion word extraction, we only employ exterior resources to flag real opinion words in $I_o$, and we do not employ a learning-based approach as in opinion target extraction. Thus, we only design two baselines: **NoPrior** and **Prior_Exterior**. The NoPrior sets all entries in $I_o$ to zero in Eq. (7). The Prior_Exterior uses an existing sentiment lexicon to flag correct opinion words (in Section 5.3). Fig. 13 shows the results. From the results, we can see that the Prior_Exterior outperforms the NoPrior and the improvement in precision is significant ($p < 0.05$) in all domains, which indicates that prior knowledge is useful for opinion word extraction.

## 7 Conclusions and Future Work

This paper proposes a novel method for co-extracting opinion targets and opinion words by using a word alignment model. Our main contribution is focused on detecting opinion relations between opinion targets and opinion words. Compared to previous methods based on nearest neighbor rules and syntactic patterns, in using a word alignment model, our method captures opinion relations more precisely and therefore is more effective for opinion target and opinion word extraction. Next, we construct an *Opinion Relation Graph* to model all candidates and the detected opinion relations among them, along with a graph co-ranking algorithm to estimate the confidence of each candidate. The items with higher ranks are extracted out. The

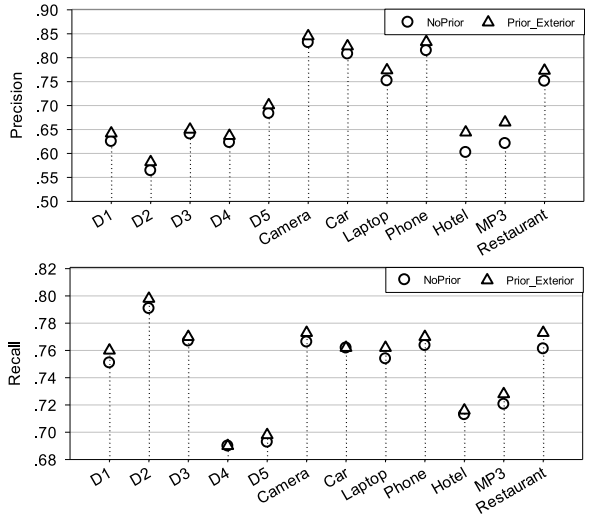



Fig. 13. Experimental comparison among different ranking methods for opinion target extraction.

experimental results for three datasets with different languages and different sizes prove the effectiveness of the proposed method.

In future work, we plan to consider additional types of relations between words, such as topical relations, in *Opinion Relation Graph*. We believe that this may be beneficial for co-extracting opinion targets and opinion words.

## ACKNOWLEDGMENTS

This work was sponsored by the National Basic Research Program of China (No. 2014CB340503), the National Natural Science Foundation of China (No. 61272332, No. 61202329 and No. 61333018), and CCF-Tencent Open Research Fund.

## REFERENCES

[1] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Seattle, WA, USA, 2004, pp. 168–177.

[2] F. Li, S. J. Pan, O. Jin, Q. Yang, and X. Zhu, "Cross-domain co-extraction of sentiment and topic lexicons," in *Proc. 50th Annu. Meeting Assoc. Comput. Linguistics*, Jeju, Korea, 2012, pp. 410–419.

[3] L. Zhang, B. Liu, S. H. Lim, and E. O'Brien-Strain, "Extracting and ranking product features in opinion documents," in *Proc. 23th Int. Conf. Comput. Linguistics*, Beijing, China, 2010, pp. 1462–1470.

[4] K. Liu, L. Xu, and J. Zhao, "Opinion target extraction using word-based translation model," in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn.*, Jeju, Korea, Jul. 2012, pp. 1346–1356.

[5] M. Hu and B. Liu, "Mining opinion features in customer reviews," in *Proc. 19th Nat. Conf. Artif. Intell.*, San Jose, CA, USA, 2004, pp. 755–760.

[6] A.-M. Popescu and O. Etzioni, "Extracting product features and opinions from reviews," in *Proc. Conf. Human Lang. Technol. Empirical Methods Natural Lang. Process.*, Vancouver, BC, Canada, 2005, pp. 339–346.

[7] G. Qiu, L. Bing, J. Bu, and C. Chen, "Opinion word expansion and target extraction through double propagation," *Comput. Linguistics*, vol. 37, no. 1, pp. 9–27, 2011.

[8] B. Wang and H. Wang, "Bootstrapping both product features and opinion words from chinese customer reviews with cross-inducing," in *Proc. 3rd Int. Joint Conf. Natural Lang. Process.*, Hyderabad, India, 2008, pp. 289–295.

[9] B. Liu, *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*, series Data-Centric Systems and Applications. New York, NY, USA: Springer, 2007.

[10] G. Qiu, B. Liu, J. Bu, and C. Che, "Expanding domain sentiment lexicon through double propagation," in *Proc. 21st Int. Jont Conf. Artif. Intell.*, Pasadena, CA, USA, 2009, pp. 1199–1204.

[11] R. C. Moore, "A discriminative framework for bilingual word alignment," in *Proc. Conf. Human Lang. Technol. Empirical Methods Natural Lang. Process.*, Vancouver, BC, Canada, 2005, pp. 81–88.

[12] X. Ding, B. Liu, and P. S. Yu, "A holistic lexicon-based approach to opinion mining," in *Proc. Conf. Web Search Web Data Mining*, 2008, pp. 231–240.

[13] F. Li, C. Han, M. Huang, X. Zhu, Y. Xia, S. Zhang, and H. Yu, "Structure-aware review mining and summarization." in *Proc. 23th Int. Conf. Comput. Linguistics*, Beijing, China, 2010, pp. 653–661.

[14] Y. Wu, Q. Zhang, X. Huang, and L. Wu, "Phrase dependency parsing for opinion mining," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Singapore, 2009, pp. 1533–1541.

[15] T. Ma and X. Wan, "Opinion target extraction in chinese news comments." in *Proc. 23th Int. Conf. Comput. Linguistics*, Beijing, China, 2010, pp. 782–790.

[16] Q. Zhang, Y. Wu, T. Li, M. Ogihara, J. Johnson, and X. Huang, "Mining product reviews based on shallow dependency parsing," in *Proc. 32nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, Boston, MA, USA, 2009, pp. 726–727.

[17] W. Jin and H. H. Huang, "A novel lexicalized HMM-based learning framework for web opinion mining," in *Proc. Int. Conf. Mach. Learn.*, Montreal, QC, Canada, 2009, pp. 465–472.

[18] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM*, vol. 46, no. 5, pp. 604–632, Sep. 1999.

[19] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai, "Topic sentiment mixture: modeling facets and opinions in weblogs," in *Proc. 16th Int. Conf. World Wide Web*, 2007, pp. 171–180.

[20] I. Titov and R. McDonald, "A joint model of text and aspect ratings for sentiment summarization," in *Proc. 46th Annu. Meeting Assoc. Comput. Linguistics*, Columbus, OH, USA, 2008, pp. 308–316.

[21] W. X. Zhao, J. Jiang, H. Yan, and X. Li, "Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Cambridge, MA, USA, 2010, pp. 56–65.

[22] A. Mukherjee and B. Liu, "Modeling review comments," in *Proc. 50th Annu. Meeting Assoc. Comput. Linguistics*, Jeju, Korea, Jul. 2012, pp. 320–329.

[23] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Comput. Linguist.*, vol. 19, no. 2, pp. 263–311, Jun. 1993.

[24] Z. Liu, H. Wang, H. Wu, and S. Li, "Collocation extraction using monolingual word alignment method," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Singapore, 2009, pp. 487–495.

[25] Z. Liu, X. Chen, and M. Sun, "A simple word trigger method for social tag suggestion," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Edinburgh, U.K., 2011, pp. 1577–1588.

[26] Q. Gao, N. Bach, and S. Vogel, "A semi-supervised word alignment algorithm with partial manual alignments," in *Proc. Joint Fifth Workshop Statist. Mach. Translation MetricsMATR*, Uppsala, Sweden, Jul. 2010, pp. 1–10.

[27] S. Baluja, R. Seth, D. Sivakumar, Y. Jing, J. Yagnik, S. Kumar, D. Ravichandran, and M. Aly, "Video suggestion and discovery for youtube: taking random walks through the view graph," in *Proc. 17th Int. Conf. World Wide Web*, Beijing, China, 2008, pp. 895–904.

[28] P. P. Talukdar, J. Reisinger, M. Pasca, D. Ravichandran, R. Bhagat, and F. Pereira, "Weakly-supervised acquisition of labeled class instances using graph random walks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Honolulu, Hawaii, 2008, pp. 582–590.

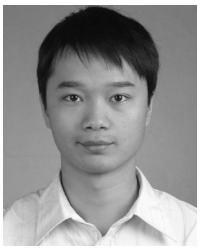
[29] K. Liu, H. L. Xu, Y. Liu, and J. Zhao, "Opinion target extraction using partially-supervised word alignment model," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, Beijing, China, 2013, pp. 2134–2140.

[30] K. W. Gan and P. W. Wong, "Annotating information structures in chinese texts using hownet," in *Proc. 2nd Workshop Chin. Lang. Process.: Held Conjunction 38th Annu. Meeting Assoc. Comput. Linguistics*, Hong Kong, 2000, pp. 85–92.

[31] Z. Hai, K. Chang, J.-J. Kim, and C. C. Yang, "Identifying features in opinion mining via intrinsic and extrinsic domain relevance," *IEEE Trans. Knowledge Data Eng.*, vol. 26, no. 3, p. 623–634, 2014.

[32] Z.-H. Zhou and M. Li, "Semi-supervised regression with co-training," in *Proc. 15th Int. Joint Conf. Artif. Intell.*, Edinburgh, Scotland, U.K.

[33] J. Zhu, H. Wang, B. K. Tsou, and M. Zhu, "Multi-aspect opinion polling from textual reviews," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, Hong Kong, 2009, pp. 1799–1802.



**Kang Liu** received the BSc and MS degrees in 2002 and 2005, respectively, from Xidian University. He received the PhD degree from NLPR, Institute of Automation, Chinese Academy of Sciences. He is currently an assistant professor at the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences. His research interests include opinion mining, information extraction, and machine learning.

**Liheng Xu** is currently an assistant professor at the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences. His research interest is fine-gained opinion mining.

**Jun Zhao** is a professor at the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include information extraction and question answering.