Fang L, Liu B, Huang ML. Leveraging large data with weak supervision for joint feature and opinion word extraction. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 30(4): 903–916 July 2015. DOI 10.1007/s11390-015-1569-3

Leveraging Large Data with Weak Supervision for Joint Feature and Opinion Word Extraction

Lei Fang (房 磊), Biao Liu (刘 彪), and Min-Lie Huang* (黄民烈), *Member, CCF*

State Key Laboratory on Intelligent Technology and Systems, Department of Computer Science and Technology Tsinghua University, Beijing 100084, China

E-mail: fang-l10@mails.tsinghua.edu.cn; liubiao2638@gmail.com; aihuang@tsinghua.edu.cn

Received September 12, 2014; revised May 4, 2015.

Abstract Product feature and opinion word extraction is very important for fine granular sentiment analysis. In this paper, we leverage large-scale unlabeled data for joint extraction of feature and opinion words under a knowledge poor setting, in which only a few feature-opinion pairs are utilized as weak supervision. Our major contributions are two-fold: first, we propose a data-driven approach to represent product features and opinion words as a list of corpus-level syntactic relations, which captures rich language structures; second, we build a simple yet robust unsupervised model with prior knowledge incorporated to extract new feature and opinion words, which obtains high performance robustly. The extraction process is based upon a bootstrapping framework which, to some extent, reduces error propagation under large data. Experimental results under various settings compared with state-of-the-art baselines demonstrate that our method is effective and promising.

Keywords opinion mining, sentiment analysis, prior knowledge, feature extraction

1 Introduction

Online reviews and opinions have become more and more valuable to consumers. According to online surveys, 70% consumers refer to reviews or ratings before online or offline purchasing[1]. Though most websites provide review-level rating statistics, there are much more demands for obtaining more detailed, complete, and specific information from textual reviews. For example, a user may want to buy a cell phone which has good ratings on battery life and screen. This requires deeper analysis, that is, fine granular sentiment analysis such as aspect-level review analysis, on consumer reviews. Aspect-level review analysis aims to process reviews according to the properties or topics of a product or service, and as a result, it may generate a concise and comprehensive picture for users.

Aspect-level or feature-level sentiment analysis is a central task in opinion mining. Compared with traditional document-level sentiment analysis[2], fine granular review analysis[3-4] provides detailed opinions in terms of different product properties (or features, aspects), which better satisfy the users' information needs. Among recent research work in sentiment analysis and opinion mining, feature and opinion extraction[3,5], which targets at extracting feature/aspect words or opinion words from reviews, is a key problem since it is a precursor to further analysis.

Feature and opinion word extraction is very challenging in that different users often make use of different words or phrases to comment on the same aspect or to express opinions. It is impractical to manually collect all the feature and opinion words, particularly when the size of data is very large. Existing studies for this task fall into two lines: one is based on rules[5-6] and statistics[7], and the other is based on generative

Regular Paper

Special Section on Social Media Processing

This work is partly supported by the National Basic Research 973 Program of China under Grant Nos. 2012CB316301 and 2013CB329403, the National Natural Science Foundation of China under Grant Nos. 61332007 and 61272227, and the Beijing Higher Education Young Elite Teacher Project.

*Corresponding Author

topic models[8]. Approaches of the first line are usually started with some given seeds, and work well on rather small corpora, since different feature-opinion pairs may share similar grammatical structures and the structures can be discovered by statistical measures. Typical studies for the second line are topic model approaches[9-11], where the models formulate the generative process of reviews and aspects in an unsupervised manner.

However, the following issues have not been fully addressed in previous studies:

1) Performance can be hampered by different initialization of seed words; error propagation would significantly degrade the performance when the size of dataset is very large (many iterations are needed); [rule- and statistics-based]

2) Rules or patterns need to be redefined for new languages or domains, which limits the capability of domain adaption; [rule-based]

3) Incapability to scale to large datasets①. [topic model-based]

In this paper, we have new considerations to address the problem of feature and opinion word extraction. The first consideration is that prior knowledge will play a key role in dealing with large-scale corpora as we are always suffered from heavy instance annotation. Not surprisingly, knowledge can help us to build learning models efficiently and effectively. In many problems, we possess a wealth of knowledge. For instance, in sentiment classification, we know that words like {"amazing", "wonderful", "impressive"} are more likely to express positive sentiment and words like {"disgusted", "ugly", "bad"} often talk about negative sentiment. In sentiment extraction, we know about some feature-opinion pairs for some aspects, such as "the story is moving" in movie reviews or "considerate service" in restaurant reviews. Such knowledge could be fully exploited so that we do not need to manually define new rules or patterns for each domain or language. Encoding such knowledge may also help to overcome the limitations of error propagation in aforementioned rule- and statistics-based approaches, as stated soon later in this paper.

As data-driven methods have already shown great success to difficult problems[13], and data itself may be the essential key to many problems[14], our second consideration comes to leverage large-scale data. We have very convenient access to large-scale data due to the prosperity of social websites. We are motivated by the fact that rich language structures between opinion and feature words can be more easily discovered with larger corpora. Given a feature-opinion pair (as prior knowledge), we can find a rich representation of language structure for the pair. For example, to represent *moving-story*, we can find all grammatical relations between "story" and "moving" in large data and use those dominant relations as features to find new feature-opinion pairs.

Thanks to the easy availability of large-scale data and the knowledge we possess for the task, we propose that a practical model should have following properties:

• be simple to leverage the large amount of information buried in huge data;

• leveraging prior knowledge, and be insensitive to what provided.

In this paper, we propose an effective approach to extract feature and opinion words with the above properties. The input to our approach is a large number of reviews and only a few feature-opinion pairs which are served as prior knowledge, enabling our approach more easily scalable to new domains. Our main contributions are two-fold.

1) Instead of heavy engineering on machine learning features or handcrafting linguistic rules, or constructing complicated probabilistic models, we propose a data-driven approach to represent product feature and opinion words as a sequence of corpus-level syntactic relations to capture rich language structures.

2) We build a simple yet robust weakly supervised learning model with prior knowledge incorporated, which obtains high performance robustly.

The rest of this paper is organized as follows. In Section 2, we briefly introduce some related work. Section 3 presents details about our approach to jointly extracting feature and opinion words. In Section 4, we discuss the experimental settings and results. We summarize our work in the last section.

2 Related Work

2.1 Extracting Feature and Opinion Words

Recently, there are many existing studies on feature and opinion word extraction, and they generally fall into two categories: supervised approaches and unsupervised approaches.

For supervised learning, Liu *et al.*[4] extracted product feature words by a supervised pattern discovery

①Though parallel Gibbs Sampling[12] or parallel Collapsed Variable Bayes are implemented for LDA to learn topics from large datasets, the extensions cannot be easily scaled.

method; Kobayashi *et al.*[15] formulated feature and opinion word extraction as a relation extraction problem, and learned a discrimination function using contextual and statistical clues; Wu *et al.*[16] defined a tree kernel over phrase dependency trees to extract relations between opinion words and product features using SVM; other supervised models formulate the feature and opinion word extraction as a sequential learning problem using conditional random fields[17-18]. For supervised methods, the merits lie in that rich features can be utilized to train the model, and parameters can be tuned to perform well on the given domain. However, these approaches are limited due to the heavy load of data annotation.

For unsupervised methods②, they can be summarized as follows.

Statistics-Based Methods. Hu and Liu[3] proposed a method to generate a feature-specific review summary, where the feature and opinion words are extracted by frequent itemset mining. Popescu and Etzioni[19] leveraged point-wise mutual information to quantify the association between product features and opinion words. Kaji and Kitsuregawa[20] used Chi-square and point-wise mutual information to extract sentiment lexicon. Hai *et al.*[7] proposed likelihood ratio tests to extract feature and opinion words.

Rule Based Methods. Zhuang *et al.*[6] proposed to extract feature-opinion pairs via some grammatical rules. Guo *et al.*[21] proposed to extract product feature with the structural cue inferred from that reviewers often briefly enumerate their concerned product-features and opinions in pros and cons. Qiu *et al.*[5] utilized several predefined grammatical relation patterns to iteratively extract feature words and opinion words, which they termed as "Double Propagation". Zhang *et al.*[22] extended the work of Qiu *et al.* by adopting other patterns to increase recall, and the HITS algorithm was employed to rank the extracted opinion targets. Gindl *et al.*[23] also used syntactic patterns to extract aspect, with anaphora resolution taken into consideration during the extraction process.

Topic Model Based Methods. Various extensions to topic models were widely studied[10-11,24-29]. These models generally describe the structure of feature and opinion words, and document-level polarity in a generative process, in which product feature is modeled by certain topic. There is also much work other than aspect feature extraction, such as feature-level rating[30], feature ranking[31-32], or feature-specific summarization[9,33]. It should be noted that parameters of these models usually need to be carefully tuned, and the obtained topics are difficult to interpret. Also, these approaches are not easy to be scaled to large corpora.

Graph Based Methods. Liu *et al.*[34] proposed to extract features using word alignment model. Liu *et al.*[35] combined syntactic patterns with alignment model to extract features, and they showed that syntax-based methods are effective when the data size is small, alignment-based methods are more useful for the medium data size, and the combination (syntax and alignment) is also effective when the data size is small or medium. However, the performance gap between different methods decreases when the data size becomes larger. Xu *et al.*[36] proposed a sentiment graph walking algorithm that incorporates the confidence of syntactic patterns to mine opinion and feature words, and a self-learning method was employed to refine the results.

2.2 Incorporating Prior Knowledge

Many research studies in data mining or machine learning attempt to promote the performance by incorporating prior knowledge. For example, Andrzejewski *et al.*[37-38] introduced knowledge to topic models. Li *et al.*[39] and Shen and Li[40] introduced lexical knowledge to the matrix factorization for sentiment analysis. Chen *et al.*[29] introduced domain knowledge to topic models to extract aspect terms. Fang *et al.*[41] encoded knowledge to latent SVM[42] to provide sentence-level aspect identification. There are many other studies about modeling prior knowledge[43-44]. A full survey is beyond the scope of the paper.

3 Leveraging Large Data to Extract Feature and Opinion Words

3.1 Overview

We propose to leverage corpus-level syntactic relations for joint extraction of feature and opinion words. Note that different users have different interpretations for the same meaning, reviews are usually written informally with various writing styles, and the grammatical relations between feature and opinion words are considerably sparse, particularly when the size of data is large.

Since it is impossible to rely on manually crafted rules or patterns to extract feature or opinion words,

②Though some approaches are initialized with seeds, most methods do not require instance annotation.

we leverage large dataset to learn relations between feature and opinion words. Our approach mainly includes the following steps.

Step 1: *Feature-Opinion Representation*. We propose a novel corpus-level syntactic representation for a feature-opinion pair, which has two advantages: 1) our representation captures rich language structures at the corpus level, which benefit the extraction of new feature and opinion words; 2) our representation is very flexible and can serve as the input for various machine learning techniques.

Step 2: *Weakly Supervised Learning*. In this step, we address the problem of extending new feature (opinion) words for one given opinion (feature) word. A few (or even only one) feature-opinion pairs are considered as prior knowledge. We then learn a weakly supervised discriminant function using this prior knowledge together with label sparsity regularization from large-scale unlabeled data (see Subsection 3.4 for details).

Step 3: *Bootstrapping Framework*. For the joint extraction of feature and opinion words, we iteratively learn the discriminant function (as explained in step 2) and utilize the discriminant function to predict new feature and opinion words in a bootstrapping framework, which, to some extent, reduces the risk of error propagation.

3.2 Notations

Prior knowledge, denoted by $\mathcal{K}$, consists of a few $(feature, opinion)$ pairs. Such prior knowledge can be easily obtained in that we need only a few such pairs (or even only one). If an opinion word modifies a feature word, the linguistic structures (dependency paths) between them may be shared by other $(feature, opinion)$ pairs. Thus we can use these structures to find new pairs, and from those new pairs, we may find new structures. By this way, our method is quite different from those methods which start from two separate lists, i.e., one list for opinion words and the other for feature words.

Table 1 presents notations we will use throughout this paper. Similar to other studies, we consider nouns or noun phrases in $\mathcal{R}$ as candidate feature word set $\mathcal{CF}$, verbs or adjectives as candidate opinion word set $\mathcal{CO}$. The feature word set $\mathcal{F}$ and the opinion set $\mathcal{O}$ are initialized by selecting corresponding feature and opinion words from the provided pairs in $\mathcal{K}$. Our approach also outputs the extracted feature-opinion pairs $\mathcal{S}$.

Table 1. Basic Notations

Symbol	Description
$\mathcal{R}$	Collection of reviews
$\mathcal{CF}$	Candidate feature set
$\mathcal{CO}$	Candidate opinion set
$\mathcal{F}$	Extracted feature set
$\mathcal{O}$	Extracted opinion set
$\mathcal{S}$	Extracted feature-opinion pairs

To expand new opinion words corresponding to a known feature word f, where $f \in \mathcal{F}$, our goal is to learn a function $\mathcal{G}(K_f, (f, co))$ that outputs the probability of f and co being a feasible feature-opinion pair using all the unlabeled pairs that contain feature word f. $K_f \in \mathcal{S}$ is obtained by aggregating all known pairs that contain feature word f from extracted known pairs $\mathcal{S}$, and $co \in \mathcal{CO}$ is a candidate opinion word. The process is similar for feature word extraction.

3.3 Feature-Opinion Representation

For a single review, we first parse sentences using Stanford Parser③. This step can be easily parallelized on Hadoop④ to handle large data. Fig.1 presents the basic dependencies for the snippet "we are all moved to tears by the moving story"⑤.

Then for each sentence in the review, we represent the relation between candidate feature word (cf) and candidate opinion word (co) by the shortest dependency path $\pi(cf \rightarrow co)$ or $\pi(co \rightarrow cf)$ in the corresponding dependency parse tree. Note that cf and co are any two candidate words (noun and verb or adjective) in the sentence. For the example shown in Fig.1, we have the corresponding shortest dependency path from *moving* to *story* and *tears* as:

$$\begin{aligned} & \pi(moving \rightarrow story) \\ = \ & [moving(VBG) \overset{amod}{\longleftarrow} story(NN)]; \\ & \pi(moving \rightarrow tears) \\ = \ & [moving(VBG) \overset{amod}{\longleftarrow} story(NN) \\ & \overset{prep\_by}{\longleftarrow} moved(VBD) \overset{prep\_to}{\longrightarrow} tears(NNS)]. \end{aligned}$$

Since $(moving, tears)$ is not a valid pair, the model will give a low score to $\pi(moving \rightarrow tears)$.

③http://nlp.stanford.edu/software/lex-parser.shtml, May 2015.

④Hadoop is an open-source implementation of MapReduce[45]. http://hadoop.apache.org/, May 2015.

⑤We visualize the basic dependencies with Stanford CoreNLP demo. http://nlp.stanford.edu:8080/corenlp/, May 2015.

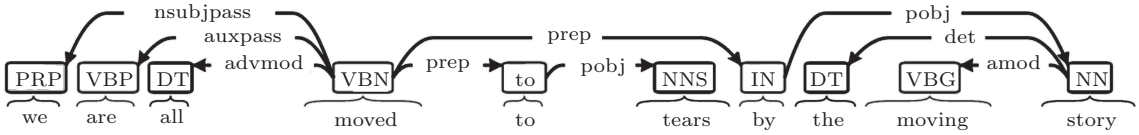


Fig.1. Basic dependencies.

For review corpus, we aggregate all the sentence-level shortest dependency paths for pair (cf, co) as⑥

$$(cf, co) = \{\pi_1 : x_1,\ \pi_2 : x_2,\ \ldots\},$$

where π_i is a dependency path from cf to co, and x_i is the number of times reaching co from cf with path π_i in the corpus. For simplicity, we use $y \in \{0, 1\}$ to indicate whether (cf, co) is a feasible feature-opinion pair or not, and the corresponding path vector is denoted by $\boldsymbol{x}$, where $\boldsymbol{x} = (x_1, x_2, \ldots)$. Thus we have $y = 1$ for all pairs in $\mathcal{K}$ and $\mathcal{S}$, the prior knowledge and the extracted pairs respectively.

It can be seen that our representation captures rich language structures at the corpus level; besides, the joint extraction of feature opinion words can be formulated as a classification problem based on this representation, and various machine learning techniques might be utilized.

3.4 Weakly Supervised Learning

Up to this point, our problem turns into a weakly supervised learning problem with only one or several feature-opinion pairs in $\mathcal{K}$ as prior knowledge. To solve this problem, we extend the generalized expectation criteria[43] to learn the discriminant function.

3.4.1 Generalized Expectation Criterion

A generalized expectation (GE) criterion[43] is a term in a parameter estimation objective function that assigns scores to values of a model expectation. GE prefers parameter settings where model expectations are close to certain reference expectation, and it is a general framework for learning from labeled features and unlabeled data.

Labeled features can be considered as domain knowledge which is in forms of affinities between input features and class labels. For example, in text classification for baseball documents vs hockey documents, even without any labeled data, the presence of the word *puck* is a strong indicator of hockey. Suppose that we specify the reference expectation for labeled feature *puck* as $\hat{p}(baseball|puck) = 0.1$ and $\hat{p}(hockey|puck) = 0.9$, GE criterion can be considered as minimizing certain distance function, say the KL divergence, between reference expectation $\hat{p}(c|puck)$ and model expectation $\tilde{p}(c|puck)$, where c is the class label, $c \in \{baseball, hockey\}$.

For our task here, we introduce two types of prior knowledge: positive labeled features and label sparsity regularization.

Positive Labeled Features. Druck *et al.*[43] demonstrated that it is effective to generate labeled features from labeled instances, but we only have several positive instances⑦ (prior knowledge $\mathcal{K}$ or extracted pairs in $\mathcal{S}$), which leads to that only positive labeled features can be utilized.

For each known pair (f, o), π_i is considered as a labeled feature if the following equation holds:

$$\sigma_i = \frac{x_i}{\sum_i x_i} > \sigma,$$

where σ is a predefined threshold (we empirically set σ to 0.1 in experiments). Recall the example shown in Fig.1, suppose we have the knowledge that *moving* and *story* are a pair of opinion and feature in $\mathcal{K}$. We enumerate all possible dependency paths from *moving* to *story* with corresponding occurrence counts in the corpus level, and find that proportion of total counts for dependency path $[moving(VBG) \overset{amod}{\longleftarrow} story(NN)]$ is above σ. Then we consider $[moving(VBG) \overset{amod}{\longleftarrow} (NN)]$ as a positive labeled feature when extending new feature words given opinion word *moving*, and $[(VBG) \overset{amod}{\longleftarrow} story(NN)]$ for extending new opinion words given feature word *story*. Note that these labeled features are automatically obtained from large data instead of manually crafted, which is different from previous rule- or pattern-based methods.

As it is difficult to accurately estimate reference expectation for these positive labeled features, we set the

⑥The proposed representation applies on all noun-verb/adjective pairs: *f-o*, *cf-o*, *f-co* and *cf-co*.

⑦We have $y = 1$ for positive instances and $y = 0$ for negative ones according to corpus-level syntactic representations.

reference expectation to a fixed value, and further experimental results show that reference expectation is not sensitive to the extraction performance when the reference expectation is above certain value.

It should also be noted that π_i, the chosen labeled feature, might occur in many instances, which makes the model expectation $\tilde{p}_\theta(y|x_i)$ deviate greatly from the human-provided reference expectation $\hat{p}(y|x_i)$. For example, there might exist a candidate feature word cf matching the labeled feature $[moving(VBG) \overset{amod}{\longleftarrow} cf(NN)]$ due to errors in parsing sentences, where cf cannot be a feasible feature word. Therefore, we take x_i, the occurrence count for π_i, instead of whether or not π_i occurs, to calculate the model expectation $\tilde{p}_\theta(y|x_i)$.

Label Sparsity Regularization. It is insufficient to train a classifier with only positive labeled features, as we need balanced knowledge on both class labels. To overcome this limitation, we introduce label sparsity regularization to ensure that the marginal distribution of our model matches the real situation.

It is common that for a given feature word, it has strong associations with only a few opinion words compared with all candidate opinion words co-occurred, and so is that for a given opinion word. That is, the label proportions for positive and negative instances are very imbalanced. Therefore, we set the expectation of model marginal distribution to $\hat{p}(y)$ and penalize classifiers whose marginal distribution $\tilde{p}_\theta(y)$ deviates from $\hat{p}(y)$. For our task here, $\hat{p}(y=1)$ is quite small.

3.4.2 Training Binary Classifiers

With positive labeled features and label sparsity regularization, we are able to train a binary classifier using GE criterion. Following previous work on GE applying to log-liner models[43], we define $p_\theta(y|\boldsymbol{x})$ parameterized by θ as

$$p_\theta(y|\boldsymbol{x}) = \frac{\exp(\sum_i \theta_{yi} x_i)}{Z(\boldsymbol{x})},$$

where $Z(\boldsymbol{x}) = \sum_y \exp(\sum_i \theta_{yi} x_i)$. Suppose L is the labeled feature set, by introducing a zero-mean σ^2-variance Gaussian prior on parameters, our goal is to minimize the objective:

$$\mathcal{O} = \underbrace{\sum_{x_i \in L} D(\hat{p}(y|x_i)||\tilde{p}_\theta(y|x_i))}_{\text{positive labeled features}} + \underbrace{\lambda D(\hat{p}(y)||\tilde{p}_\theta(y))}_{\text{label sparsity}} + \sum_{y,j} \frac{\theta_{yj}^2}{2\sigma^2}, \quad (1)$$

where λ is the parameter balancing the weight between positive labeled features and label sparsity regularizer, and $D(\cdot||\cdot)$ denotes the KL divergence. We use L-BFGS to solve the optimization problem. The gradient of the labeled feature part in (1) is the same as in [43], and for the label sparsity regularizer, the gradient with respect to the model parameter for feature j and labels y', $\theta_{y'j}$, has the form as

$$\begin{aligned} &\frac{\partial}{\partial \theta_{y'j}} D(\hat{p}(y)||\tilde{p}_\theta(y)) \\ =& -\frac{\partial}{\partial \theta_{y'j}} \sum_y \hat{p}(y) \log \tilde{p}_\theta(y) \\ =& -\frac{1}{|C|} \sum_y \frac{\hat{p}_\theta(y)}{\tilde{p}_\theta(y)} \sum_{\boldsymbol{x} \in C} p_\theta(y|\boldsymbol{x}) \times \\ & \Big(I(y=y') x_j - p_\theta(y'|\boldsymbol{x}) x_j \Big), \end{aligned}$$

where C is the total number of training instances, $I(y=y')$ is an indicator function with 1 when $y=y'$ and 0 elsewhere.

For our task, we define $\mathcal{G}$ for finding new opinion words given known feature word f as

$$\mathcal{G}(K_f, (f, co)) = p_{\theta_f}(y=1|\boldsymbol{x}),$$

and recall that $p_{\theta_f}(y|\boldsymbol{x})$ is a trained log-linear model parameterized by θ_f, and training data is all unlabeled pairs that contain feature word f. The prior knowledge $\mathcal{K}$ and the extracted known feature-opinion pairs in $\mathcal{S}$ are fully used as we obtain K_f by aggregating all known pairs that contain feature word f. Positive labeled features are then generated from K_f. Similarly, we have $\mathcal{G}(K_o, (cf, o))$ and $p_{\theta_o}(y|\boldsymbol{x})$ when extending feature words given known opinion word o.

3.5 Bootstrapping Framework

In order to discover new product features and opinion words, we propose a bootstrapping framework to iteratively extract product features and opinion words, as shown in Algorithm 1. In this framework, $\mathcal{HF}$ and $\mathcal{HO}$ represent the extracted feature and opinion words with high confidence scores, and *con* $(\mathcal{G})$ is the condition for whether or not the candidate is a feasible feature or opinion. We may define the condition as whether or not the score $\mathcal{G}$ is above a predefined classification threshold or the score $\mathcal{G}$ is ranked in top N positions. In our experiment, we choose the second option, and extract only top scored words.

Algorithm 1. Bootstrapping Framework
Input:
a set of feature-opinion pairs, $\mathcal{K}$;
a collection of unlabeled reviews, $\mathcal{R}$;
Output:
extracted product feature-opinion pairs, $\mathcal{S}$;
extracted opinion words, $\mathcal{O}$;
extracted product features, $\mathcal{F}$;
1: $\mathcal{CF} \leftarrow$ candidate features extracted from $\mathcal{R}$;
2: $\mathcal{CO} \leftarrow$ candidate opinions extracted from $\mathcal{R}$;
3: $\mathcal{F} \leftarrow$ all feature words that occur in $\mathcal{K}$;
4: $\mathcal{O} \leftarrow$ all opinion words that occur in $\mathcal{K}$;
5: $\mathcal{S} = \mathcal{K}$; $\mathcal{HF} = \mathcal{F}$; $\mathcal{HO} = \mathcal{O}$
6: **repeat**
7: **for all** known feature word $f \in \mathcal{HF}$ **do**
8: $K_f \leftarrow$all known pairs that contain f from $\mathcal{S}$
9: Learn parameter θ_f of $\mathcal{G}(K_f,(f,co))$
10: **for all** $co \in \mathcal{CO} - \mathcal{O}$ **do**
11: **if** $con(\mathcal{G}(K_f,(f,co))) =$ true **then**
12: $\mathcal{S} = \mathcal{S} \cup \{(f,co)\}$; $\mathcal{O} = \mathcal{O} \cup \{co\}$
13: **end if**
14: **end for**
15: **end for**
16: **for all** known opinion word $o \in \mathcal{HO}$ **do**
17: $K_o \leftarrow$all known pairs that contain o from $\mathcal{S}$
18: Learn parameter θ_o of $\mathcal{G}(K_o,(cf,o))$
19: **for all** $cf \in \mathcal{CF} - \mathcal{F}$ **do**
20: **if** $con(\mathcal{G}(K_o,(cf,o))) =$ true **then**
21: $\mathcal{S} = \mathcal{S} \cup \{(cf,o)\}$; $\mathcal{F} = \mathcal{F} \cup \{cf\}$
22: **end if**
23: **end for**
24: **end for**
25: $\mathcal{HF} \leftarrow$high confidence feature words in $\mathcal{F}$
26: $\mathcal{HO} \leftarrow$high confidence opinion words in $\mathcal{O}$
27: **until** no new opinion/feature words are identified
28: **return** $\mathcal{S}, \mathcal{O}, \mathcal{F}$;

The confidence score for new extracted opinion (feature) word o_{new} (f_{new}) given known feature (opinion) f_{old} (o_{old}) is defined as follows:

$$s_j(o_{\text{new}}) = \mathcal{G}(K_f, (f, o_{\text{new}})) \times s_{j-1}(f_{\text{old}}),$$
$$s_j(f_{\text{new}}) = \mathcal{G}(K_o, (f_{\text{new}}, o)) \times s_{j-1}(o_{\text{old}}),$$

where j is the index of iterations, and initially, we set the confidence score to 1 for feature or opinion words in $\mathcal{K}$. Note that since $\mathcal{G} < 1$ holds, the confidence score for new extracted words will decrease after each iteration. After that, words with high confidence serve as seeds to extract new words for further iterations. Our framework has a snowballing effect as knowledge grows, because when expanding new words, we update K_f and K_o by aggregating all known pairs that contain feature word f or opinion word o (see line 11 and line 20); then we re-estimate parameters θ_f and θ_o from unlabeled data using K_f and K_o.

Our approach reduces the risk of error propagation from two perspectives: 1) unlike previous studies, we expand new words only with high confidence score; 2) benefited from feature-opinion representation, our model has less chance to make errors under this bootstrapping framework while in rule- or pattern-based approaches, errors might be more easily included by single rule or pattern matching.

A limitation of our approach is that there are too many models since we train a classifier for every feature or opinion word in each iteration. Though it is possible to share a common model for different feature-opinion pairs, a common model is less accurate because different feature-opinion pairs might have entirely different dependency paths. Fortunately, the proposed approach can be easily parallelized and it is very fast to train a single model.

4 Experiments

4.1 Dataset

We employ two datasets to evaluate our approach: restaurant reviews from Dianping[⑧] and movie reviews from douban[⑨]. We do not present the results on other public corpora used in previous studies due to the fact that the size of these corpora is rather small. We then split these reviews into sentences, and the sentences are parsed by Stanford parser[46].

Table 2. Data Statistics

Domain	Number of Reviews	Average Number of Sentences
Movie	5 327 438	2.8
Restaurant	4 851 247	12.3

Table 2 shows the number of reviews and the average number of sentences in review for movie and restaurant domains, respectively. It can be seen that our dataset is considerably larger than that of previous studies.

We choose the following state-of-the-art baselines:

• DP (Double Propagation)[5] proposed by Qiu *et al.* First some dependency-rule based patterns are manually defined to represent the syntactic relations between

⑧http://www.dianping.com/, May 2015.
⑨http://movie.douban.com/, May 2015.

feature words and opinion words. With several feature and opinion words as initialization seeds, new feature and opinion words are then extracted through pattern matching in a bootstrapping framework.

• DP-HITS[22] proposed for feature extraction. It extends DP by introducing new patterns to increase recall. The extracted feature candidates are ranked by relevance and frequency using HITS.

• LRTBoot[7]. It uses likelihood ratio test to model the statistical association between any two words. With several feature words as initialization seeds, a bootstrapping framework (similar to DP and DP-HITS) is employed to mine new feature and opinion words that have strong statistic association with extracted ones.

For DP-HITS, we do not present the result on opinion word extraction, as the original work only focused on feature word extraction. For LRTBoot, we use the same parameters as in [7]. We do not compare it with topic models and extensions as they cannot be easily scaled. Graph-based approaches are not compared, either, as parameters need to be carefully tuned with different sizes of data.

Prior Knowledge. The prior knowledge is supplied with respect to aspect. For example, the aspects are "story", "music", "acting", "picture" and "director" for movie reviews; "taste", "ambience", "service", "price" and "location" for restaurant reviews. Without loss of generality, we manually select only one feature-opinion pair for each aspect as prior knowledge.

Evaluation Metrics. Previous studies[5,34,36] evaluate the extraction performance in terms of precision, recall and F_1 score on a rather small dataset. As our dataset is very large, it is very difficult to create a golden standard manually. Further, we will show that the recall metric is inappropriate for large data.

Fig.2 shows the percentage of extracted feature opinion words from the corresponding candidates. Note that the number of candidates is at the order of magnitude 10^5 on our corpora, and it is unlikely to have so many feature or opinion words. Furthermore, we find that only the top hundreds of results are valid when ranking with document frequency in decreasing order (the results of DP-HITS are ranked by the output ranking score). The precision of the remainder words is fairly low because the employed bootstrapping framework suffers from severe error propagation when the size of data is very large (many iterations are needed to find a sufficient number of words). Therefore, we choose precision@k for evaluation measure and only manually annotate top hundreds of results. Statistics on the labeled data show that the precision of the chosen baselines is about 0.5 for top 1 000 features and opinions (on both movie and restaurant reviews), which suggests that recall is unnecessary to be assessed.

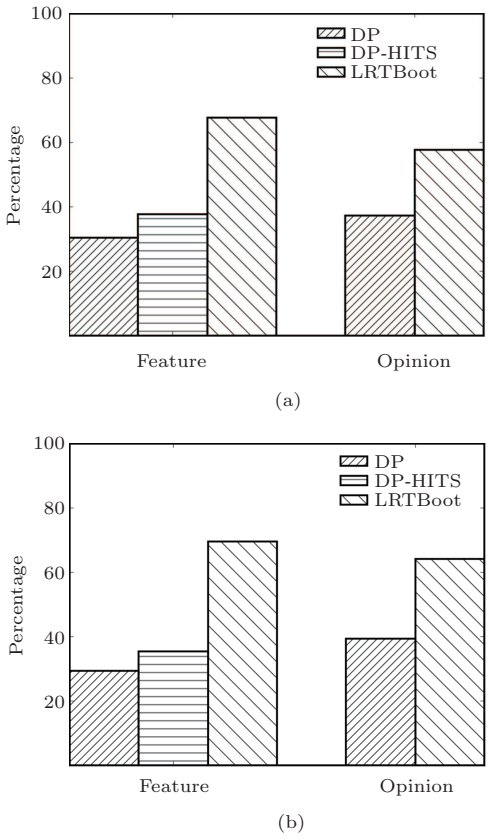


Fig.2. Extraction percentage of baselines. (a) Movie. (b) Restaurant.

Parameter Setting. Our approach has three parameters: the number of initial feature-opinion pairs, the reference expectation (RE) for positive labeled features, and the minimum confidence (MC) score for extracted results as new seeds for further extraction. To ensure high accuracy, only top 10 scored words are extracted as new feature or opinion words, and we accordingly set the labeled sparsity regularizer $\hat{p}(y = 1) = 10/T$ where T is the size of training data. We empirically set $\lambda = 5P$ where P is the size of positive labeled features. For default settings, we have five pairs (one pair for each aspect) as prior knowledge for each domain, and $MC = 0.85$, $RE = 0.95$. We will give detailed discussions about the parameters to demonstrate that our approach is robust under different settings.

Table 3. Case Studies on Feature and Opinion Word Extraction

画面(picture) ⇌ 精致(delicacy)		剧情(plot) ⇌ 老套(cliche)		环境(ambiance) ⇌ 优雅(elegant)		服务(service) ⇌ 周到(considerate)	
Feature	Opinion	Feature	Opinion	Feature	Opinion	Feature	Opinion
台词(lines)	一流(first-class)	情节(scenario)	不错(not bad)	地方(location)	一流(first-class)	店家(restaurant)	一流(first-class)
电影(movie)	不错(not bad)	故事(story)	丰富(rich)	布置(decorating)	不同(characteristic)	态度(attitude)	不好(not good)
细节(detail)	优美(graceful)	桥段(moment)	乱(mess)	感觉(feeling)	不好(not good)	服务业(service)	不怎么样(not good)
场景(scene)	华丽(gorgeous)	结局(ending)	假(fake)	装修(decorating)	不怎么样(not good)	服务员(waitor)	不行(not good)
镜头(shot)	唯美(aestheticism)	题材(theme)	傻(stupid)	餐厅(dinning hall)	不错(good)	服务生(waiter)	不错(good)
制作(manufacture)	够(enough)	内容(content)	冗长(lengthiness)	音乐(music)	乱(mess)	小姐(waitress)	专业(professional)
场面(scene)	好(good)	电影(movie)	单一(invariant)	店(restaurant)	优美(elegant)	招呼(serve)	亲切(kind)
布景(setting)	好看(beautiful)	主题(subject)	单调(tedium)	灯光(light)	别致(exquisite)	热情(warmly)	优质(high quality)
特效(special-effects)	完美(perfect)	手法(technique)	合理(reasonable)	气氛(atmosphere)	压抑(depressed)	礼貌(polite)	体贴(considerate)

4.2 Case Studies on Feature and Opinion Word Extraction

We first show some case studies on feature and opinion word extraction before further analysis. Table 3 presents several cases using the provided feature-opinion pair. The left two blocks are from movie reviews and the right two blocks are from restaurant reviews. For all the cases shown here, results are obtained with only one seed as prior knowledge. It can be seen that even with only one feature-opinion pair as prior knowledge, our approach is capable of extracting feature and opinion words with a relatively high precision. It also explains that our approach of training binary classifiers for the task of feature and opinion word extraction is effective and promising.

4.3 Comparison with Baselines

We compare our approach with the aforementioned baselines.

Fig.3 and Fig.4 illustrate the comparisons with baselines for feature and opinion word extraction in movie reviews, and Fig.5 and Fig.6 are for restaurant reviews. It can be seen that the performance of our approach outperforms that of the baselines, and when k increases, the performance of our approach falls more slowly than that of baselines, suggesting that our model has a lower risk of error propagation.

DP and LRTBoot have almost the same performance, because words with high frequency would simultaneously have strong statistical association and match predefined grammatical rules, particularly when the size of data is large. Our approach has very stable or even slightly increased precision as k increases, which demonstrates that it is effective to incorporate prior knowledge, and our bootstrapping framework has a snowballing effect as knowledge grows when more feature and opinion words are extracted.

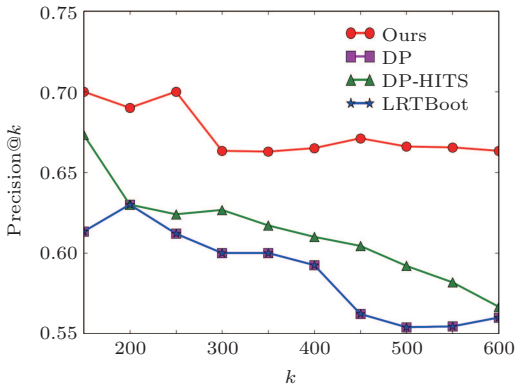


Fig.3. Extraction performance for movie feature.

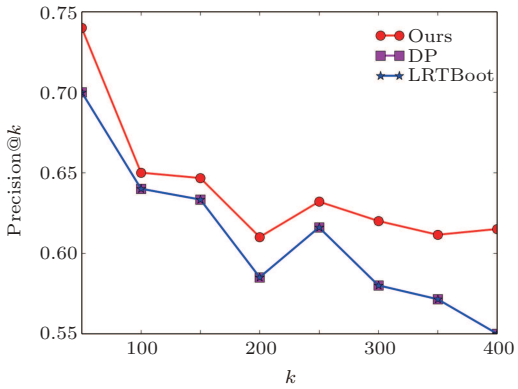


Fig.4. Extraction performance for movie opinion.

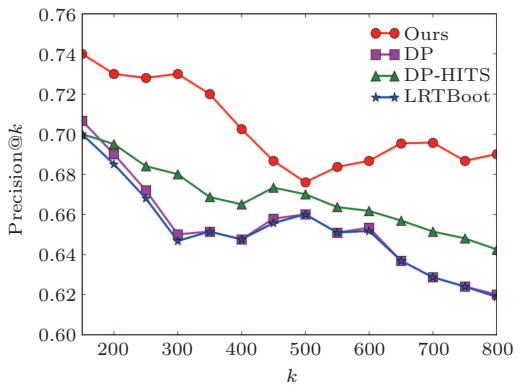


Fig.5. Extraction performance for restaurant feature.

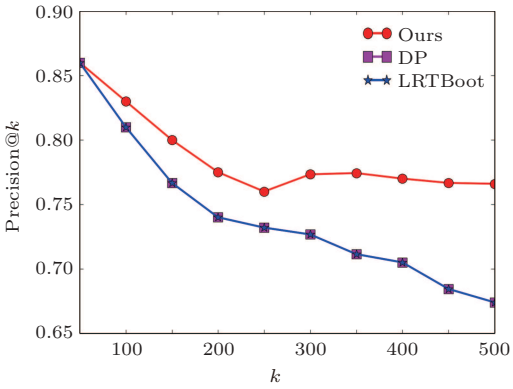


Fig.6. Extraction performance for restaurant opinion.

For further evaluations, we merge the extraction results of feature and opinion words, and use precision@k to evaluate the overall performance. We also add the results of DP with all pairs as initial seeds for fair comparison. Though DP-HITS obtains a slightly better performance than other baselines, it is mainly focusing on feature word extraction. Therefore, we do not include it in the subsequent experiments.

4.4 Evaluation of Sensitivity to Prior Knowledge

As our approach starts with feature-opinion pairs as prior knowledge, we shall justify whether our approach is sensitive to the supplied knowledge.

4.4.1 With Just One Seed Pair

We evaluate the performance with only one feature-opinion pair as prior knowledge. By default, we have five pairs for each domain. We test five runs for each domain, and each run has only one pair. Then we calculate the mean and variance of precision@k for these five runs.

Fig.7 and Fig.8 show the averaged performance with the variance of movie and restaurant reviews. It can be seen that our approach is stable (in that the variance is low) when different feature-opinion pairs are encoded as prior knowledge.

4.4.2 With More Seed Pairs

We further evaluate the performance with different numbers of seed feature-opinion pairs. We choose 1, 3 and 5 feature-opinion pairs as prior knowledge for each domain respectively.

Fig.9 and Fig.10 show the extraction performance for movie and restaurant reviews, respectively. It clearly shows that under different amounts of prior knowledge, our method stays stable with high performance, and for the restaurant domain, the precision improves slightly when more prior knowledge is introduced.

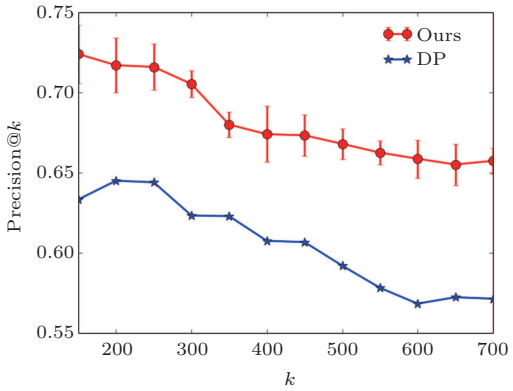


Fig.7. Averaged performance with variance of movie reviews (one feature-opinion pair).

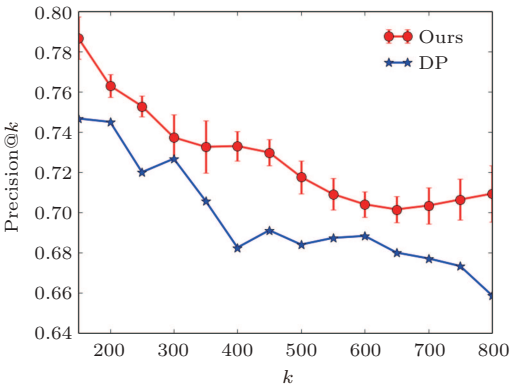


Fig.8. Averaged performance with variance of restaurant reviews (one feature-opinion pair).

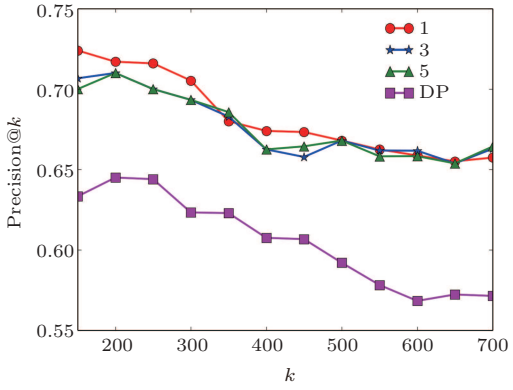


Fig.9. Performance under different amounts of prior knowledge (movie reviews).

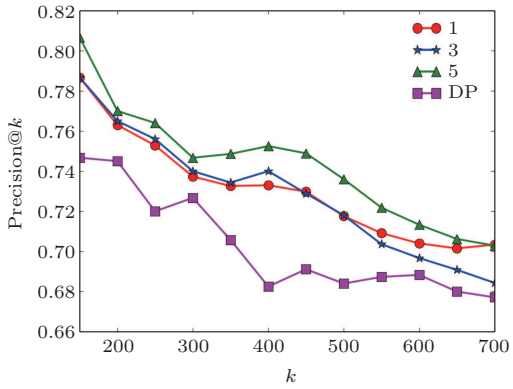


Fig.10. Performance under different amounts of prior knowledge (restaurant reviews).

The above two experiments show that our approach achieves rather stable performance under different prior knowledge with different sizes. It explains that our approach is insensitive to the prior knowledge provided. We attribute the robust performance to the corpus-level representation for feature word and opinion word under large data, since with large data, the rich syntactic relations between feature word and opinion word can be better captured and modeled.

4.5 Sensitivity of Reference Expectations

We evaluate the extraction performance under different reference expectations. Reference expectation can be viewed as the confidence for labeled features. We start our approach by setting the reference expectation of positive labeled features to 0.8, 0.85, 0.9 and 0.95 respectively. The goal is to demonstrate that it is easy to select parameters for our approach, and thus reference expectations of much lower values are not discussed here.

Fig.11 and Fig.12 show the averaged extraction performance with variance when varying reference expectation for each domain. It can be seen that the overall performance of our approach is robust under different reference expectations.

4.6 Sensitivity of Confidence Threshold

In our approach, the extracted new feature or opinion words with high confidence scores (above the confidence threshold) are considered as seeds for expanding new feature or opinion words in the next iterations. We shall evaluate whether the confidence threshold affects the extraction performance. In a similar way, we set the minimum confidence to 0.8, 0.85 and 0.9, respectively.

Fig.13 and Fig.14 present the averaged overall extraction performance with variance for movie and restaurant reviews, respectively. It shows that our approach is robust and achieves stable performance over different confidence thresholds.

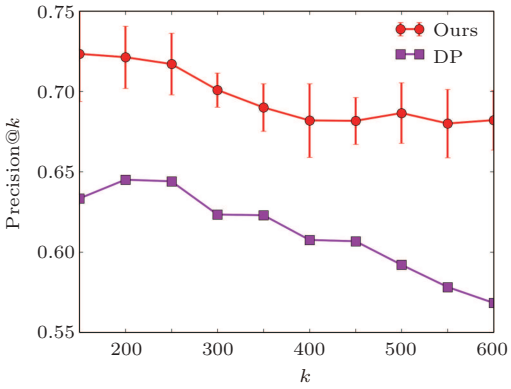


Fig.11. Averaged performance with variance under different reference expectations (movie reviews).

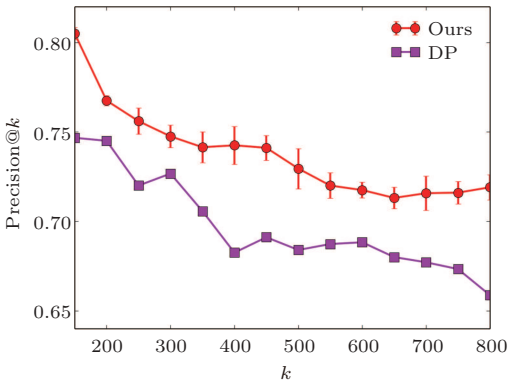


Fig.12. Averaged performance with variance under different reference expectations (restaurant reviews).

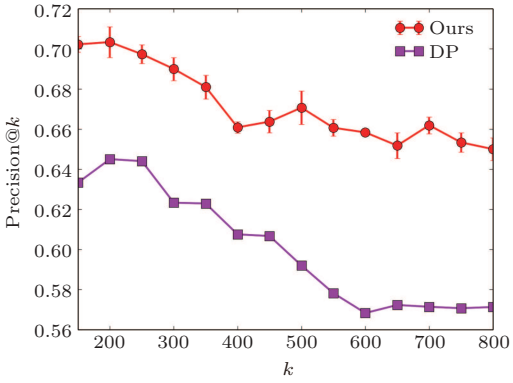


Fig.13. Averaged performance with variance under different minimum confidences (movie reviews).

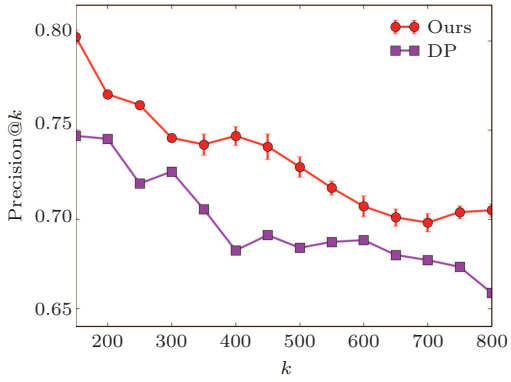


Fig.14. Averaged performance with variance under different minimum confidences (restaurant reviews).

To summarize, we have evaluated the performance of our approach under various settings.

• The case studies show that our approach is capable of extracting feature and opinion words even with only one feature-opinion pair as prior knowledge.

• Comparisons with state-of-the-art baselines demonstrate that it is effective to have prior knowledge encoded, and our approach has a lower risk of error propagation.

• The performance under different prior knowledge shows that our approach is insensitive to the knowledge provided.

• The performance under different reference expectations and the performance under different minimum confidence scores demonstrate that our approach is robust under different parameter settings.

Experimental results demonstrate that our approach of leveraging large data with weak supervision for joint feature and opinion word extraction is effective and promising.

5 Conclusions and Future Work

In this paper, we proposed a simple yet robust approach to jointly extract feature and opinion words by leveraging large-scale data. We formulated the extraction problem as learning a dependency path scoring function using labeled features under the generalized expectation criterion. Labeled features are generated from large-scale data using weak supervision. The extraction process is based upon a bootstrapping framework which, to some extent, reduces error propagation. Our method achieves a relative robust high performance compared with state-of-the-art baselines under various settings.

For future work, we plan to investigate other types of labeled features as prior knowledge to promote the extraction performance, such as corpus-level statistics or semantic coherence. We are employing our results for further fine granular sentiment analysis, such as aspect-level review summarization, phrase-level review visualization, and service or product recommendation.

Acknowledgement We thank the anonymous reviewers for their valuable comments.

References

[1] Ante S E. Amazon: Turning consumer opinions into gold. *Business Week*. http://www.bloomberg.com/bw/magazine/content/09_43/b4152047039565.htm, May 2015.

[2] Pang B, Lee L, Vaithyanathan S. Thumbs up?: Sentiment classification using machine learning techniques. In *Proc. the ACL-02 Conference on Empirical Methods in Natural Language Processing*, Jul. 2002, pp.79-86.

[3] Hu M, Liu B. Mining and summarizing customer reviews. In *Proc. the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2004, pp. 168-177.

[4] Liu B, Hu M, Cheng J. Opinion observer: Analyzing and comparing opinions on the web. In *Proc. the 14th International Conference on World Wide Web*, May 2005, pp.342-351.

[5] Qiu G, Liu B, Bu J, Chen C. Opinion word expansion and target extraction through double propagation. *Comput. Linguist.*, 2011, 37(1): 9-27.

[6] Zhuang L, Jing F, Zhu X Y. Movie review mining and summarization. In *Proc. the 15th ACM International Conference on Information and Knowledge Management*, Nov. 2006, pp.43-50.

[7] Hai Z, Chang K, Cong G. One seed to find them all: Mining opinion features via association. In *Proc. the 21st ACM International Conference on Information and Knowledge Management*, Oct. 29 – Nov. 2, 2012, pp.255-264.

[8] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation. *Journal of Machine Learning Research*, 2003, 3: 993-1022.

[9] Titov I, McDonald R. A joint model of text and aspect ratings for sentiment summarization. In *Proc. the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Jun. 2008, pp.308-316.

[10] Zhao W X, Jiang J, Yan H, Li X. Jointly modeling aspects and opinions with a Maxent-LDA hybrid. In *Proc. the 2010 Conference on Empirical Methods in Natural Language Processing*, Oct. 2010, pp.56-65.

[11] Mukherjee A, Liu B. Aspect extraction through semi-supervised modeling. In *Proc. the 50th Annual Meeting of the Association for Computational Linguistics*, Jul. 2012, pp.339-348.

[12] Newman D, Asuncion A, Smyth P, Welling M. Distributed algorithms for topic models. *Journal of Machine Learning Research*, 2009, 10: 1801-1828.

[13] Lin J, Kolcz A. Large-scale machine learning at Twitter. In *Proc. the 2012 ACM SIGMOD International Conference on Management of Data*, May 2012, pp.793-804.

[14] Halevy A, Norvig P, Pereira F. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 2009, 24(2): 8-12.

[15] Kobayashi N, Inui K, Matsumoto Y. Extracting aspect-evaluation and aspect-of relations in opinion mining. In *Proc. the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Jun. 2007, pp.1065-1074.

[16] Wu Y, Zhang Q, Huang X, Wu L. Phrase dependency parsing for opinion mining. In *Proc. the 2009 Conference on Empirical Methods in Natural Language Processing*, Aug. 2009, pp.1533-1541.

[17] Li F, Han C, Huang M, Zhu X, Xia Y J, Zhang S, Yu H. Structure-aware review mining and summarization. In *Proc. the 23rd International Conference on Computational Linguistics*, Aug. 2010, pp.653-661.

[18] Choi Y, Cardie C. Hierarchical sequential learning for extracting opinions and their attributes. In *Proc. the ACL 2010 Conference Short Papers*, Jul. 2010, pp.269-274.

[19] Popescu A M, Etzioni O. Extracting product features and opinions from reviews. In *Proc. the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Oct. 2005, pp.339-346.

[20] Kaji N, Kitsuregawa M. Building lexicon for sentiment analysis from massive collection of HTML documents. In *Proc. the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, June 2007, pp.1075-1083.

[21] Guo H, Zhu H, Guo Z, Zhang X, Su Z. Product feature categorization with multilevel latent semantic association. In *Proc. the 18th ACM Conference on Information and Knowledge Management*, Nov. 2009, pp.1087-1096.

[22] Zhang L, Liu B, Lim S H, O'Brien-Strain E. Extracting and ranking product features in opinion documents. In *Proc. the 23rd International Conference on Computational Linguistics*, Aug. 2010, pp.1462-1470.

[23] Gindl S, Weichselbraun A, Scharl A. Rule-based opinion target and aspect extraction to acquire affective knowledge. In *Proc. the 22nd International Conference on World Wide Web Companion*, May 2013, pp.557-564.

[24] Mei Q, Ling X, Wondra M, Su H, Zhai C. Topic sentiment mixture: Modeling facets and opinions in weblogs. In *Proc. the 16th International Conference on World Wide Web*, May 2007, pp.171-180.

[25] Brody S, Elhadad N. An unsupervised aspect-sentiment model for online reviews. In *Proc. Human Language Technologies: The 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Jun. 2010, pp.804-812.

[26] Jo Y, Oh A H. Aspect and sentiment unification model for online review analysis. In *Proc. the 4th ACM International Conference on Web Search and Data Mining*, Feb. 2011, pp.815-824.

[27] Lu B, Ott M, Cardie C, Tsou B K. Multi-aspect sentiment analysis with topic models. In *Proc. the 11th IEEE International Conference on Data Mining Workshops*, Dec. 2011, pp.81-88.

[28] Moghaddam S, Ester M. ILDA: Interdependent LDA model for learning latent aspects and their ratings from online product reviews. In *Proc. the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul. 2011, pp.665-674.

[29] Chen Z, Mukherjee A, Liu B, Hsu M, Castellanos M, Ghosh R. Exploiting domain knowledge in aspect extraction. In *Proc. the 2013 Conference on Empirical Methods in Natural Language Processing*, Oct. 2013, pp.1655-1667.

[30] Wang H, Lu Y, Zhai C. Latent aspect rating analysis on review text data: A rating regression approach. In *Proc. the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Jul. 2010, pp.783-792.

[31] Snyder B, Barzilay R. Multiple aspect ranking using the good grief algorithm. In *Proc. Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Apr. 2007, pp.300-307.

[32] Yu J, Zha Z J, Wang M, Chua T S. Aspect ranking: Identifying important product aspects from online consumer reviews. In *Proc. the 49th Annual Meeting of the Association for Computational Linguistics*, Jun. 2011, pp.1496-1505.

[33] Li P, Wang Y, Gao W, Jiang J. Generating aspect-oriented multi-document summarization with event-aspect model. In *Proc. the Conference on Empirical Methods in Natural Language Processing*, Jul. 2011, pp.1137-1146.

[34] Liu K, Xu L, Zhao J. Opinion target extraction using word-based translation model. In *Proc. the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Jul. 2012, pp.1346-1356.

[35] Liu K, Xu L, Zhao J. Syntactic patterns versus word alignment: Extracting opinion targets from online reviews. In *Proc. the 51st Annual Meeting of the Association for Computational Linguistics*, Aug. 2013, pp.1754-1763.

[36] Xu L, Liu K, Lai S, Chen Y, Zhao J. Mining opinion words and opinion targets in a two-stage framework. In *Proc. the 51st Annual Meeting of the Association for Computational Linguistics*, Aug. 2013, pp.1764-1773.

[37] Andrzejewski D, Zhu X, Craven M. Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *Proc. the 26th Annual International Conference on Machine Learning*, Jun. 2009, pp.25-32.

[38] Andrzejewski D, Zhu X, Craven M, Recht B. A framework for incorporating general domain knowledge into latent dirichlet allocation using first-order logic. In *Proc. the 22nd International Joint Conference on Artificial Intelligence*, Jul. 2011, pp.1171-1177.

[39] Li T, Zhang Y, Sindhwani V. A non-negative matrix tri-factorization approach to sentiment classification with lexical prior knowledge. In *Proc. the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, Aug. 2009, pp.244-252.

[40] Shen C, Li T. A non-negative matrix factorization based approach for active dual supervision from document and word labels. In *Proc. the Conference on Empirical Methods in Natural Language Processing*, Jul. 2011, pp.949-958.

[41] Fang L, Huang M, Zhu X. Exploring weakly supervised latent sentiment explanations for aspect-level review analysis. In *Proc. the 22nd ACM International Conference on Information and Knowledge Management*, Oct. 27 – Nov. 1, 2013, pp.1057-1066.

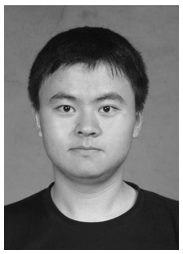
[42] Yu C N J, Joachims T. Learning structural SVMs with latent variables. In *Proc. the 26th Annual International Conference on Machine Learning*, Jun. 2009, pp.1169-1176.

[43] Druck G, Mann G, McCallum A. Learning from labeled features using generalized expectation criteria. In *Proc. the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul. 2008, pp.595-602.

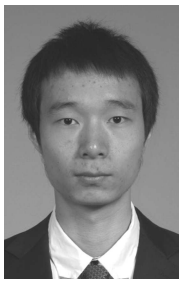
[44] Ganchev K, Graça J, Gillenwater J, Taskar B. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 2010, 11: 2001-2049.

[45] Dean J, Ghemawat S. MapReduce: Simplified data processing on large clusters. *Communications of the ACM*, 2008, 51(1): 107-113.

[46] Klein D, Manning C D. Accurate unlexicalized parsing. In *Proc. the 41st Annual Meeting on Association for Computational Linguistics*, Jul. 2003, pp.423-430.



Lei Fang is a fifth year Ph.D. student in the Department of Computer Science and Technology, Tsinghua University, Beijing. He received his Bachelor's degree in computer science and technology from Harbin Institute of Technology, in 2010. His research interest includes natural language processing, data mining, and machine learning.



Biao Liu is a master candidate in the Department of Computer Science and Technology, Tsinghua University, Beijing. He received his Bachelor's degree in computer science and technology from Tsinghua University, in 2014. His research interest includes natural language processing and machine learning.

Min-Lie Huang is an associate professor in the Department of Computer Science and Technology, Tsinghua University, Beijing. He received his Bachelor's and Ph.D. degrees in computer science from Tsinghua University, in 2000 and 2006 respectively. He has published tens of papers on major conferences including ACL, IJCAI, AAAI, CIKM, EMNLP, COLING, ICDM, etc. His research interests are mainly focused on natural language processing, data mining, and machine learning.