Extracting and Ranking Product Features in Opinion Documents

Lei Zhang
Department of Computer Science
University of Illinois at Chicago
lzhang3@cs.uic.edu

Bing Liu
Department of Computer Science
University of Illinois at Chicago
liub@cs.uic.edu

Suk Hwan Lim
Hewlett-Packard Labs
suk-hwan.lim@hp.com

Eamonn O'Brien-Strain
Hewlett-Packard Labs
eob@hpl.hp.com

Abstract

An important task of opinion mining is to extract people's opinions on features of an entity. For example, the sentence, "*I love the GPS function of Motorola Droid"* expresses a positive opinion on the "*GPS function*" of the Motorola phone. "*GPS function*" is the feature. This paper focuses on mining features. *Double propagation* is a state-of-the-art technique for solving the problem. It works well for medium-size corpora. However, for large and small corpora, it can result in low precision and low recall. To deal with these two problems, two improvements based on *part-whole* and "*no*" patterns are introduced to increase the recall. Then feature ranking is applied to the extracted feature candidates to improve the precision of the top-ranked candidates. We rank feature candidates by feature importance which is determined by two factors: feature relevance and feature frequency. The problem is formulated as a bipartite graph and the well-known web page ranking algorithm HITS is used to find important features and rank them high. Experiments on diverse real-life datasets show promising results.

1 Introduction

In recent years, opinion mining or sentiment analysis (Liu, 2010; Pang and Lee, 2008) has been an active research area in NLP. One task is to extract people's opinions expressed on features of entities (Hu and Liu, 2004). For example, the sentence, "*The picture of this camera is amazing*", expresses a positive opinion on the *picture* of the camera. "*picture*" is the feature. How to extract features from a corpus is an important problem. There are several studies on feature extraction (e.g., Hu and Liu, 2004, Popescu and Etzioni, 2005, Kobayashi et al., 2007, Scaffidi et al., 2007, Stoyanov and Cardie. 2008, Wong et al., 2008, Qiu et al., 2009). However, this problem is far from being solved.

Double Propagation (Qiu et al., 2009) is a state-of-the-art unsupervised technique for solving the problem. It mainly extracts noun features, and works well for medium-size corpora. But for large corpora, this method can introduce a great deal of noise (low precision), and for small corpora, it can miss important features. To deal with these two problems, we propose a new feature mining method, which enhances that in (Qiu et al., 2009). Firstly, two improvements based on *part-whole* patterns and "*no*" patterns are introduced to increase recall. Part-whole or *meronymy* is an important semantic relation in NLP, which indicates that one or more objects are parts of another object.

Coling 2010: Poster Volume, pages 1462–1470,
Beijing, August 2010

For example, the phrase "*the engine of the car*" contains the part-whole relation that "*engine*" is part of "*car*". This relation is very useful for feature extraction, because if we know one object is part of a product class, this object should be a feature. "*no*" pattern is another extraction pattern. Its basic form is the word "no" followed by a noun/noun phrase, for instance, "no noise". People often express their short comments or opinions on features using this pattern. Both types of patterns can help find features missed by double propagation. As for the low precision problem, we present a feature ranking approach to tackle it. We rank feature candidates based on their importance which consists of two factors: feature relevance and feature frequency. The basic idea of feature importance ranking is that if a feature candidate is correct and frequently mentioned in a corpus, it should be ranked high; otherwise it should be ranked low in the final result. Feature frequency is the occurrence frequency of a feature in a corpus, which is easy to obtain. However, assessing feature relevance is challenging. We model the problem as a bipartite graph and use the well-known web page ranking algorithm HITS (Kleinberg, 1999) to find important features and rank them high. Our experimental results show superior performances. In practical applications, we believe that ranking is also important for feature mining because ranking can help users to discover important features from the extracted hundreds of fine-grained candidate features efficiently.

2 Related work

Hu and Liu (2004) proposed a technique based on association rule mining to extract product features. The main idea is that people often use the same words when they comment on the same product features. Then frequent itemsets of nouns in reviews are likely to be product features while the infrequent ones are less likely to be product features. This work also introduced the idea of using opinion words to find additional (often infrequent) features.

Popescu and Etzioni (2005) investigated the same problem. Their algorithm requires that the product class is known. The algorithm determines whether a noun/noun phrase is a feature by computing the pointwise mutual information (PMI) score between the phrase and class-specific discriminators, e.g., "*of xx*", "*xx has*", "*xx comes with*", etc., where *xx* is a product class. This work first used part-whole patterns for feature mining, but it finds part-whole based features by searching the Web. Querying the Web is time-consuming. In our method, we use predefined part-whole relation patterns to extract features in a domain corpus. These patterns are domain-independent and fairly accurate.

Following the initial work in (Hu and Liu 2004), several researchers have further explored the idea of using opinion words in product feature mining. A dependency based method was proposed in (Zhuang et al., 2006) for a movie review analysis application. Qiu et al. (2009) proposed a double propagation method, which exploits certain syntactic relations of opinion words and features, and propagates through both opinion words and features iteratively. The extraction rules are designed based on different relations between opinion words and features, and among opinion words and features themselves. Dependency grammar was adopted to describe these relations. In (Wang and Wang, 2008), another bootstrapping method was proposed. In (Kobayashi et al. 2007), a pattern mining method was used. The patterns are relations between feature and opinion pairs (they call *aspect-evaluation* pairs). The patterns are mined from a large corpus using pattern mining. Statistics from the corpus are used to determine the confidence scores of the extraction.

In general information extraction, there are two approaches: rule-based and statistical. Early extraction systems are mainly based on rules (e.g., Riloff, 1993). In statistical methods, the most popular models are Hidden Markov Models (HMM) (Rabiner, 1989), Maximum Entropy Models (ME) (Chieu et al., 2002) and Conditional Random Fields (CRF) (Lafferty et al., 2001). CRF has been shown to be the most effective method. It was used in (Stoyanov et al., 2008). However, a limitation of CRF is that it only captures local patterns rather than long range patterns. It has been shown in (Qiu et al., 2009) that many feature and opinion word pairs have long range dependencies. Experimental results in (Qiu et al., 2009) indicate that CRF does not perform well.

Other related works on feature extraction mainly use topic modeling to capture topics in reviews (Mei et al., 2007). In (Su et al., 2008), the authors also proposed a clustering based method with mutual reinforcement to identify features. However, topic modeling or clustering is only able to find some general/rough features, and has difficulty in finding fine-grained or precise features, which is more related to information extraction.

3 The Proposed Method

As discussed in the introduction section, our proposed method deals with the problems of double propagation. So let us give a short explanation why double propagation can cause problems in large or small corpora.

Double propagation assumes that features are nouns/noun phrases and opinion words are adjectives. It is shown that opinion words are usually associated with features in some ways. Thus, opinion words can be recognized by identified features, and features can be identified by known opinion words. The extracted opinion words and features are utilized to identify new opinion words and new features, which are used again to extract more opinion words and features. This propagation or bootstrapping process ends when no more opinion words or features can be found. The biggest advantage of the method is that it requires no additional resources except an initial seed opinion lexicon, which is readily available (Wilson et al., 2005, Ding et al., 2008). Thus it is domain independent and unsupervised, avoiding laborious and time-consuming work of labeling data for supervised learning methods. It works well for medium–size corpora. But for large corpora, this method may extract many nouns/noun phrases which are not features. The precision of the method thus drops. The reason is that during propagation, adjectives which are not opinionated will be extracted as opinion words, e.g., "*entire*" and "*current*". These adjectives are not opinion words but they can modify many kinds of nouns/noun phrases, thus leading to extracting wrong features. Iteratively, more and more noises may be introduced during the process. The other problem is that for certain domains, some important features do not have opinion words modifying them. For example, in reviews of mattresses, a reviewer may say "*There is a valley on my mattress*", which implies a negative opinion because "*valley*" is undesirable for a mattress. Obviously, "*valley*" is a feature, but "*valley*" may not be described by any opinion adjective, especially for a small corpus. Double propagation is not applicable in this situation.

To deal with the problem, we propose a novel method to mine features, which consists of two steps: feature extraction and feature ranking. For feature extraction, we still adopt the double propagation idea to populate feature candidates. But two improvements based on part-whole relation patterns and a "no" pattern are made to find features which double propagation cannot find. They can solve part of the recall problem. For feature ranking, we rank feature candidates by feature importance.

A part-whole pattern indicates one object is part of another object. For the previous example "*There is a valley on my mattress*", we can find that it contains a part-whole relation between "*valley*" and "*mattress*". "*valley*" belongs to "*mattress*", which is indicated by the preposition "*on*". Note that "*valley*" is not actually a part of mattress, but an effect on the mattress. It is called a *pseudo part-whole* relation. For simplicity, we will not distinguish it from an actual part-whole relation because for our feature mining task, they have little difference. In this case, "noun_1 on noun_2" is a good indicative pattern which implies noun_1 is part of noun_2. So if we know "*mattress*" is a class concept, we can infer that "*valley*" is a feature for "*mattress*". There are many phrase or sentence patterns representing this type of semantic relation which was studied in (Girju et al, 2006). Beside part-whole patterns, "no" pattern is another important and specific feature indicator in opinion documents. We introduce these patterns in detail in Sections 3.2 and 3.3.

Now let us deal with the first problem: noise. With opinion words, part-whole and "no" patterns, we have three feature indicators at hands, but all of them are ambiguous, which means that they are not hard rules. We will inevitably extract wrong features (also called noises) by using them. Pruning noises from feature candidates is a hard task. Instead, we propose a new angle for solving this problem: feature ranking. The basic idea is that we rank the extracted feature candidates by feature importance. If a feature candidate is correct and important, it should be ranked high. For unimportant feature or noise, it should be ranked low in the final result. Ranking is also very useful in practice. In a large corpus, we may extract hundreds of fine-grained features. But the user often only cares about those important ones, which should be ranked high. We identified two major factors affecting the feature importance: one is feature relevance and the other is feature frequency.

Feature relevance: it describes how possible a feature candidate is a correct feature. We find that there are three strong clues to indicate feature relevance in a corpus. The first clue is that a correct feature is often modified by multiple opinion words (adjectives or adverbs). For example, in the mattress domain, "*delivery*" is modified by "*quick*" "*cumbersome*" and "*timely*". It shows that reviewers put emphasis on the word "*delivery*". Thus we can infer that "*delivery*" is a possible feature. The second clue is that a feature could be extracted by multiple part-whole patterns. For example, in the car domain, if we find following two phrases, "*the engine of the car*" and "*the car has a big engine*", we can infer that "*engine*" is a feature for car, because both phrases contain part-whole relations to indicate "*engine*" is a part of "*car*". The third clue is the combination of opinion word modification, part-whole pattern extraction and "no" pattern extraction. That is, if a feature candidate is not only modified by opinion words but also extracted by part-whole or "no" patterns, we can infer that it is a feature with high confidence. For example, for sentence "*there is a bad hole in the mattress*", it strongly indicates that "*hole*" is a feature for a mattress because it is modified by opinion word "*bad*" and also in the part-whole pattern. What is more, we find that there is a mutual enforcement relation between opinion words, part-whole and "no" patterns, and features. If an adjective modifies many correct features, it is highly possible to be a good opinion word. Similarly, if a feature candidate can be extracted by many opinion words, part-whole patterns, or "no" pattern, it is also highly likely to be a correct feature. This indicates that the Web page ranking algorithm HITS is applicable.

Feature frequency: This is another important factor affecting feature ranking. Feature frequency has been considered in (Hu and Liu, 2004; Blair-Goldensohn et al., 2008). We consider a feature f_1 to be more important than feature f_2 if f_1 appears more frequently than f_2 in opinion documents. In practice, it is desirable to rank those frequent features higher than infrequent features. The reason is that missing a frequently mentioned feature in opinion mining is bad, but missing a rare feature is not a big issue.

Combining the above factors, we propose a new feature mining method. Experiments show good results on diverse real-life datasets.

3.1 Double Propagation

As we described above, double propagation is based on the observation that there are natural relations between opinion words and features due to the fact that opinion words are often used to modify features. Furthermore, it is observed that opinion words and features themselves have relations in opinionated expressions too (Qiu et al., 2009). These relations can be identified via a dependency parser (Lin, 1998) based on the dependency grammar. The identification of the relations is the key to feature extraction.

Dependency grammar: It describes the dependency relations between words in a sentence. After parsed by a dependency parser, words in a sentence are linked to each other by a certain relation. For a sentence, "*The camera has a good lens*", "*good*" is the opinion word and "*lens*" is the feature of camera. After parsing, we can find that "*good*" depends on "*lens*" with relation ***mod***. Here ***mod*** means that "*good*" is the adjunct modifier for "*lens*". In some cases, an opinion word and a feature are not directly dependent, but they directly depend on a same word. For example, from the sentence "*The lens is nice*", we can find that both feature "*lens*" and opinion word "*nice*" depend on the verb "*is*" with the relation ***s*** and ***pred*** respectively. Here ***s*** means that "*lens*" is the surface subject of "*is*" while ***pred*** means that "*nice*" is the predicate of the "*is*" clause.

In (Qiu et al., 2009), it defines two categories of dependency relations to summarize all types of dependency relations between two words, which are illustrated in Figure 1. Arrows are used to represent dependencies.

Direct relations: It represents that one word depends on the other word directly or they both depend on a third word directly, shown in (a) and (b) of Figure 1. In (a), *B* depends on *A* directly, and in (b) they both directly depend on *D*.

Indirect relation: It represents that one word depends on the other word through other words or they both depend on a third word indirectly. For example, in (c) of Figure 1, *B* depends on *A* through *D*; in (d) of Figure 1, *A* depends on *D* through I_1 while *B* depends on *D* through I_2. For some complicated situations, there can be more than one I_1 or I_2.

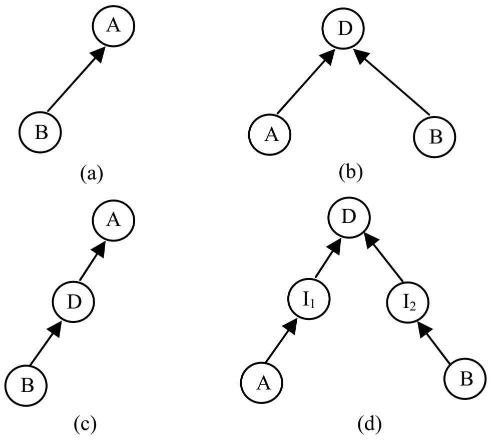

Fig.1 Different relations between A and B

Parsing indirect relations is error-prone for Web corpora. Thus we only use direct relation to extract opinion words and feature candidates in our application. For detailed extraction rules, please refer to the paper (Qiu et al., 2009).

3.2 Part-whole relation

As we discussed above, a part-whole relation is a good indicator for features if the class concept word (the "whole" part) is known. For example, the compound nominal "*car hood*" contains the part-whole relation. If we know "*car*" is the class concept word, then we can infer that "*hood*" is a feature for *car*. Part-whole patterns occur frequently in text and are expressed by a variety of lexico-syntactic structures (Girju et al, 2006; Popescu and Etzioni, 2005). There are two types of lexico-syntactic structures conveying part-whole relations: unambiguous structure and ambiguous structure. The unambiguous structure clearly indicates a part-whole relation. For example, for sentences "*the camera consists of lens, body and power cord.*" and "*the bed was made of wood*". In these cases, the detection of the patterns leads to the discovery of real part-whole relations. We can easily find features of the camera and the bed. Unfortunately, this kind of patterns is not very frequent in a corpus. However, there are many ambiguous expressions that are explicit but convey part-whole relations only in some contexts. For example, for two phrases "*valley on the mattress*" and "*toy on the mattress*", "*valley*" is a part of "*mattress*" whereas "*toy*" is not a part of "*mattress*". Our idea is to use both the unambiguous and ambiguous patterns. Although ambiguous patterns may bring some noise, we can rank them low in the ranking procedure. The following two kinds of patterns are what we have utilized for feature extraction.

3.2.1 Phrase pattern

In this case, the part-whole relation exists in a phrase.

NP + Prep + CP: noun/noun phrase (NP) contains the *part* word and the class concept phrase (CP) contains the *whole* word. They are connected by the preposition word (Prep). For example, "*battery of the camera*" is an instance of this pattern where NP (battery) is the *part* noun and CP (camera) is the *whole* noun. For our application, we only use three specific prepositions: "of", "in" and "on".

CP + with + NP: likewise, CP is the class concept phrase, and NP is the noun/noun phrase. They are connected by the word "*with*". Here NP is likely to be a feature. For example, in a phrase, "*mattress with a cover", "cover*" is a feature for mattress.

NP CP or CP NP: noun/noun phase (NP) and class concept phrase (CP) forms a compound word. For example, "*mattress pad*". Here "*pad*" is a feature of "*mattress*".

3.2.2 Sentence pattern

In these patterns, the part-whole relation is indicated in a sentence. The patterns contain specific verbs. The *part* word and the *whole* word can be found inside noun phrases or prepositional phrases which contain specific prepositions. We utilize the following patterns in our application.

"**CP Verb NP**": CP is the class concept phrase that contains the *whole* word, NP is the noun phrase that contains the *part* word and the verb is restricted and specific. For example, in a sentence, "*the phone has a big screen*", we can infer that "*screen*" is a feature for "*phone*", which is a class concept. In sentence patterns, verbs play an important role. We use indicative verbs to find part-whole relations in a sentence, i.e., “has”, “have” “include” “contain” “consist”, “comprise” and so on (Girju et al, 2006).

It is worth mentioning that in order to use part-whole relations, the class concept word for a corpus is needed, which is fairly easy to find because the noun with the most frequent occurrences in a corpus is always the class concept word based on our experiments.

3.3 “no” Pattern

Besides opinion word and part-whole relation, “no” pattern is also an important pattern indicating features in a corpus. Here “no” represents word *no*. The basic form of the pattern is “no” word followed by noun/noun phrase. This simple pattern actually is very useful to feature extraction. It is a specific pattern for product reviews and forum posts. People often express their comments or opinions on features by this short pattern. For example, in a mattress domain, people always say that “*no noise*” and “*no indentation*”. Here “*noise*” and “*indentation*” are all features for the mattress. We discover that this pattern is frequently used in corpora and a very good indicator for features with a fairly high precision. But we have to take care of the some fixed “no” expression, like “*no problem*” “*no offense*”. In these cases, “*problem*” and “*offense*” should not be regarded as features. We have a list of such words, which are manually compiled.

3.4 Bipartite Graph and HITS Algorithm

Hyperlink-induced topic search (HITS) is a link analysis algorithm that rates Web pages. As discussed in the introduction section, we can apply the HITS algorithm to compute feature relevance for ranking.

Before illustrating how HITS can be applied to our scenario, let us first give a brief introduction to HITS. Given a broad search query *q*, HITS sends the query to a search engine system, and then collects *k* (*k* = 200 in the original paper) highest ranked pages, which are assumed to be highly relevant to the search query. This set is called the root set *R*; then it grows *R* by including any page pointed to a page in *R*, then forms a base set *S*. HITS then works on the pages in *S*. It assigns every page in *S* an **authority score** and a **hub score**. Let the number of pages to be studied be *n*. We use *G* = (*V*, *E*) to denote the (directed) link graph of *S*. *V* is the set of pages (or nodes) and *E* is the set of directed edges (or links). We use *L* to denote the adjacency matrix of the graph.

$$L_{ij} = \begin{cases} 1 & if\ (i,j) \in E \\ 0 & otherwise \end{cases} \qquad (1)$$

Let the authority score of the page *i* be *A(i)*, and the hub score of page *i* be *H(i)*. The mutual reinforcing relationship of the two scores is represented as follows:

$$A(i) = \sum_{(j,i)\in E} H(j) \qquad (2)$$

$$H(i) = \sum_{(i,j)\in E} A(j) \qquad (3)$$

We can write them in a matrix form. We use **A** to denote the column vector with all the authority scores, **A** = $(A(1), A(2), ..., A(n))^T$, and use **H** to denote the column vector with all the hub scores, **H** = $(H(1), H(2), ..., H(n))^T$,

$$\mathbf{A} = L^T\mathbf{H} \qquad (4)$$

$$\mathbf{H} = L\mathbf{A} \qquad (5)$$

To solve the problem, the widely used method is power iteration, which starts with some random values for the vectors, e.g., $A_0 = H_0 = (1, 1, 1, \ldots 1,)$. It then continues to compute iteratively until the algorithm converges.

From the formulas, we can see that the authority score estimates the importance of the content of the page, and the hub score estimates the values of its links to other pages. An authority score is computed as the sum of the scaled hub scores that point to that page. A hub score is the sum of the scaled authority scores of the pages it points to. The key idea of HITS is that a good hub points to many good authorities and a good authority is pointed by many good hubs. Thus, authorities and hubs have a mutual reinforcement relationship.

For our scenario, we have three strong clues for features in a corpus: opinion words, part-whole patterns, and the “no” pattern. Although all these three clues are not hard rules, there exist mutual enforcement relations between them. If an adjective modify many features, it is highly likely to be a good opinion word. If a feature candidate is modified by many opinion words, it is likely to be a genuine feature. The same goes with part-whole patterns, the “no” pattern, or the combination for these three clues. This kind of mutual enforcement relation can be naturally modeled in the HITS framework.

Applying the HITS algorithm: Based on the key idea of HITS algorithm and feature indicators, we can apply the HITS algorithm to obtain the feature relevance ranking. Features act as authorities and feature indicators act as hubs. Different from the general HITS algorithm, features only have authority scores and feature indicators only have hub scores in our case. They form a directed bipartite graph, which is illustrated in Figure 2. We can run the HITS algorithm on this bipartite graph. The basic idea is that if a feature candidate has a high authority score, it must be a highly-relevant feature. If a feature indicator has a high hub score, it must be a good feature indicator.

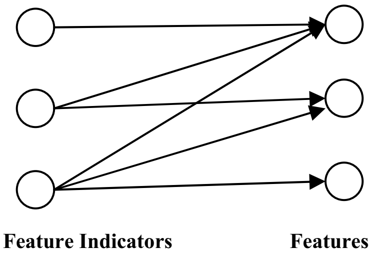

Fig. 2 Relations between feature indicators and features

3.5 Feature Ranking

Although the HITS algorithm can rank features by feature relevance, the final ranking is not only determined by relevance. As we discussed before, feature frequency is another important factor affecting the final ranking. It is highly desirable to rank those correct and frequent features at top because they are more important than the infrequent ones in opinion mining (or even other applications). With this in mind, we put everything together to present the final algorithm that we use. We use two steps:

Step 1: Compute feature score using HITS without considering frequency. Initially, we use three feature indicators to populate feature candidates, which form a directed bipartite graph. Each feature candidate acts as an authority node in the graph; each feature indicator acts as a hub node. For node s in the graph, we let H_s be the hub score and A_s be the authority score. Then, we initialize H_s and A_s to 1 for all nodes in the graph. We update the scores of H_s and A_s until they converge using power iteration. Finally, we normalize A_s and compute the score *S* for a feature.

Step 2: The final score function considering the feature frequency is given in Equation (6).

$$S = S(f)\log(freq(f)) \qquad (6)$$

where $freq(f)$ is the frequency count of ture f, and *S*(*f*) is the authority score of the candidate feature *f*. The idea is to push the frequent candidate features up by multiplying the log of frequency. Log is taken in order to reduce the effect of big frequency count numbers.

4 Experiments

This section evaluates the proposed method. We first describe the data sets, evaluation metrics and then the experimental results. We also compare our method with the double propagation method given in (Qiu et al., 2009).

4.1 Data Sets

We used four diverse data sets to evaluate our techniques. They were obtained from a commercial company that provides opinion mining services. Table 1 shows the domains (based on their names) and the number of sentences in each data set ("Sent." means the sentence). The data in "Cars" and "Mattress" are product reviews extracted from some online review sites. "Phone" and "LCD" are forum discussion posts extracted from some online forum sites. We split each review/post into sentences and the sentences are POS-tagged using the Brill's tagger (Brill, 1995). The tagged sentences are the input to our system.

Data Sets	**Cars**	**Mattress**	**Phone**	**LCD**
# of Sent.	2223	13233	15168	1783

Table 1. Experimental data sets

4.2 Evaluation Metrics

Besides precision and recall, we adopt the **precision@N** metric for experimental evaluation (Liu, 2006). It gives the percentage of correct features that are among the top *N* feature candidates in a ranked list. We compare our method's results with those of double propagation which ranks extracted candidates only by occurrence frequency.

4.3 Experimental Results

We first compare our results with double propagation on recall and precision for different corpus sizes. The results are presented in Tables 2, 3, and 4 for the four data sets. They show the precision and recall of 1000, 2000, and 3000 sentences from these data sets. We did not try more sentences because manually checking the recall and precision becomes prohibitive. Note that there are less than 3000 sentences for "Cars" and "LCD" data sets. Thus, the columns for "Cars" and "LCD" are empty in Table 4. In the Tables, "DP" represents the double propagation method; "Ours" represents our proposed method; "Pr" represents precision, and "Re" represents recall.

	Cars		**Mattress**		**Phone**		**LCD**	
	Pr	**Re**	**Pr**	**Re**	**Pr**	**Re**	**Pr**	**Re**
DP	0.79	0.55	0.79	0.54	0.69	0.23	0.68	0.43
Ours	0.78	0.56	0.77	0.64	0.68	0.44	0.66	0.55

Table 2. Results of 1000 sentences

	Cars		**Mattress**		**Phone**		**LCD**	
	Pr	**Re**	**Pr**	**Re**	**Pr**	**Re**	**Pr**	**Re**
DP	0.70	0.65	0.70	0.58	0.67	0.42	0.64	0.52
Ours	0.66	0.69	0.70	0.66	0.70	0.50	0.62	0.56

Table 3. Results of 2000 sentences

	Cars	**Mattress**		**Phone**		**LCD**
		Pr	**Re**	**Pr**	**Re**	
DP		0.65	0.59	0.64	0.48	
Ours		0.66	0.67	0.62	0.51	

Table 4. Results of 3000 sentences

From the tables, we can see that for corpora in all domains, our method outperforms double propagation on recall with only a small loss in precision. In data sets for "Phone" and "Mattress", the precisions are even better. We also find that with the increase of the data size, the recall gap between the two methods becomes smaller gradually and the precisions of both methods also drop. However, in this case, feature ranking plays an important role in discovering important features.

Ranking comparison between the two methods is shown in Tables 5, 6, and 7, which give the precisions of top 50, 100 and 200 results respectively. Note that the experiments reported in these tables were run on the whole data sets. There were no more results for the "LCD" data beyond top 200 as there were only a limited number of features discussed in the data. So the column for "LCD" in Table 7 is empty. We rank the extracted feature candidates based on frequency for the double propagation method (DP). Using occurrence frequency is the natural way to rank features. The more frequent a feature occurs in a corpus, the more important it is. However, frequency-based ranking assumes the extracted candidates are correct features. The tables show that our proposed method (Ours) outperforms double propagation considerably. The reason is that some highly-frequent feature candidates extracted by double propagation are not correct features. Our method considers the feature relevance as an important factor. So it produces much better rankings.

	Cars	**Mattress**	**Phone**	**LCD**
DP	0.84	0.81	0.64	0.68
Ours	0.94	0.90	0.76	0.76

Table 5. Precision at top 50

	Cars	**Mattress**	**Phone**	**LCD**
DP	0.82	0.80	0.65	0.68
Ours	0.88	0.85	0.75	0.73

Table 6. Precision at top 100

	Cars	**Mattress**	**Phone**	**LCD**
DP	0.75	0.71	0.70	
Ours	0.80	0.79	0.76	

Table 7. Precision at top 200

5 Conclusion

Feature extraction for entities is an important task for opinion mining. The paper proposed a new method to deal with the problems of the state-of-the-art double propagation method for feature extraction. It first uses part-whole and "no" patterns to increase recall. It then ranks the extracted feature candidates by feature importance, which is determined by two factors: feature relevance and feature frequency. The Web page ranking algorithm HITS was applying to compute feature relevance. Experimental results using diverse real-life datasets show promising results. In our future work, apart from improving the current methods, we also plan to study the problem of extracting features that are verbs or verb phrases.

Acknowledgement

This work was funded by a HP Labs Innovation Research Program Award (CW165044).

References

Blair-Goldensohn, Sasha., Kerry, Hannan., Ryan, McDonald., Tyler, Neylon., George A. Reis, Jeff, Reyna. 2008. Building Sentiment Summarizer for Local Service Reviews In *Proceedings of the Workshop of NLPIX . WWW, 2008*

Brill, Eric. 1995. Transformation-Based Error-Driven Learning and Natural Language Processing: a case study in part of speech tagging. *Computational Linguistics, 1995.*

Chieu, Hai Leong and Hwee-Tou Ng. 2002. Name Entity Recognition: a Maximum Entropy Approach Using Global Information. In *Proceedings of the 6th Workshop on Very Large Corpora,* 2002.

Ding, Xiaowen., Bing Liu and Philip S. Yu. 2008. A Holistic Lexicon-Based Approach to Opinion Mining In *Proceedings of WSDM 2008.*

Girju, Roxana., Adriana Badulescu and Dan Moldovan. 2006. “Automatic Discovery of Part-Whole Relations” *Computational Linguistics ,32(1):83-135 2006*

Hu, Mingqin and Bing Liu. 2004. Mining and Summarizing Customer Reviews. In *Proceedings of KDD 2004*

Kleinberg, Jon. 1999. “Authoritative sources in hyperlinked environment” *Journal of the ACM 46 (5): 604-632 1999*

Kobayashi, Nozomi., Kentaro Inui and Yuji Matsumoto. 2007 Extracting Aspect-Evaluation and Aspect-of Relations in Opinion Mining. In *Proceedings of EMNLP*, 2007.

Lafferty, John., Andrew McCallum and Fernando Pereira. 2001 Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of ICML*, 2001.

Lin, Dekang. 1998. Dependency-based evaluation of MINIPAR. In *Proceedings of the Workshop on Evaluation of Parsing System at ICLRE* 1998.

Liu, Bing. 2006. *Web Data Mining: Exploring Hyperlinks, contents and usage data.* Springer, 2006.

Liu, Bing. 2010. Sentiment analysis and subjectivity. *Handbook of Natural Language Processing, second edition*, 2010.

Mei, Qiaozhu, Ling Xu, Matthew Wondra, Hang Su and ChengXiang Zhai. 2007. Topic Sentiment Mixture: Modeling Facets and Opinions in Weblogs. In *Proceedings of WWW, pages 171–180, 2007.*

Pang, Bo., Lillian Lee. 2008. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval pp. 1-135 2008*

Pantel, Patrick., Eric Crestan, Arkady Borkovsky, Ana-Maria Popescu, Vishunu Vyas. 2009. Web-Scale Distributional Similarity and Entity Set Expansion. In *Proceedings of. EMNLP, 2009*

Popescu, Ana-Maria and Oren, Etzioni. 2005. Extracting product features and opinions from reviews. In *Proceedings of EMNLP, 2005.*

Qiu, Guang., Bing, Liu., Jiajun Bu and Chun Chen. 2009. Expanding Domain Sentiment Lexicon through Double Propagation. In *Proceedings of IJCAI 2009.*

Rabiner, Lawrenence. 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of the IEEE, 77(2), 1989.*

Riloff, Ellen. 1993. Automatically Constructing a Dictionary for Information Extraction Tasks. In *Proceedings of AAAI 1993.*

Scaffidi, Christopher., Kevin Bierhoff, Eric Chang, Mikhael Felker, Herman Ng and Chun Jin. 2007. Red opal: Product-feature Scoring from Reviews. In *Proceedings of EC 2007*

Stoyanov, Veselin and Claire Cardie. 2008. Topic Identification for Fine-grained Opinion Analysis. In *Proceedings of COLING 2008*

Su, Qi., Xinying Xu., Honglei Guo, Zhili Guo, Xian Wu, Xiaoxun Zhang, Bin Swen and Zhong Su. 2008. Hidden Sentiment Association in Chinese Web Opinion Mining. In *Proceedings of WWW 2008.*

Wang, Bo., Houfeng Wang. 2008. Bootstrapping both Product Features and Opinion Words from Chinese Customer Reviews with Cross-Inducing In *Proceedings of IJCNLP 2008*

Wilson, Theresa., Janyce Wiebe and Paul Hoffmann. 2005. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *Proceedings of HLT/EMNLP 2005*

Wong, Tak-Lam., Wai Lam and Tik-Sun Wong. 2008. An Unsupervised Framework for Extracting and Normalizing Product Attributes from Multiple Web Sites In *Proceedings of SIGIR 2008*

Zhuang, Li., Feng Jing, Xiao-yan Zhu. 2006. Movie Review Mining and Summarization. In *Proceedings of CIKM 2006*