A Bootstrapping Method for Finer-Grained Opinion Mining Using Graph Model

Shu Zhang, Yingju Xia, Yao Meng, and Hao Yu

Fujitsu Research and Development Center
Dong Si Huan Zhong Rd, Chaoyang District, Beijing 100025, China
{zhangshu, yjxia, mengyao, yu}@cn.fujitsu.com

Abstract. Pursuing on the analysis of product reviews, a bootstrapping method is proposed to find the product features and opinion words in iterative steps. Different from conventional methods, a graph model is built to link and measure the relationship between the pairs of product features and opinion words. A rule-based method is presented to get the initial seeds of product features and opinion words automatically. Our experimental results on electronic product reviews are encouraging, which prove the proposed method and techniques are effective in performing this task of feature level opinion mining.

Keywords: opinion mining, product reviews, bootstrapping, graph model

1 Introduction

In recent years, there has been a wealth of opinion source in the web. These opinions have attracted some special groups of people. For example, the government and politicians are interested in the response to governmental policies, merchants and potential customers are interested in the feedback of and comment on commercial products. Therefore, how to analyze and monitor the tide of prevalent attitudes on the web, how to extricate people from wading through a large number of opinions to find their interest, have received considerable attention in research community.

A series of topic symposiums and evaluation sessions on opinion mining have appeared in TREC and NTCIR. For the opinion mining on document and sentence level, the task is to classify either positively or negatively in a review. However, the sentiment orientation of a review is not sufficient for many applications. Opinion mining begins to focus on the finer-grained features level mining. The task is to find not only the sentiment orientation but also the commented features. This information could be used to deeply analyze prevalent attitudes or generate various types of opinion summaries.

In feature level opinion mining, most researches are oriented on product reviews. The task is typically divided into three main subtasks: (i) identifying product features, (ii) identifying opinions regarding the product features, and (iii) determining the sentiment orientation of the opinions. In this paper, we focus on the first two steps to find the product features and opinion words in product reviews. Different from the previous work, we unify these two separate tasks into one process. A bootstrapping method is proposed to iteratively find both of them. A graph is built by linking pairs of product features and opinion words. The short length reviews are separately analyzed by rule-based method to find the initial seeds used in the bootstrapping.

The remainder of the paper is organized as follows: section 2 describes the related work on feature level opinion analysis. Section 3 gives our approach of bootstrapping. Section 4 presents the graph model. Section 5 describes the techniques of initial seeds selection and candidates extraction. Section 6 gives the experiments and results. Finally section 7 summarizes this paper.

2 Related Work

With the study of finer-grained opinion mining, there are some feature level based opinion mining systems. For example, Opinion Observer (Liu *et al.*, 2005) which focuses on customer reviews on the web and provides the visual comparison of consumer opinions. Red Opal (Scaffidi *et al.*, 2007) as a search system which enables customers to locate products rapidly based on features.

At present, the techniques on identifying product features are primarily based on unsupervised mining. The most representative research is that of (Hu and Liu, 2004). They adopt association rule mining for extracting nouns as frequent features. Compactness pruning and redundancy pruning are used to filter the incorrect features. Then frequent features are used to find adjacent opinion words (adjectives) that modify them. Infrequent features are extracted with these opinion words. Their method is effective in finding the product features. However, there is a problem that some frequent nouns and noun phrases may not be real product features. Directly using frequent nouns as features may lead to the reduction of precision.

Popescu and Etzioni (2005) utilize relation-specific extraction patterns with web PMI assessor to assess feature candidates. They consider the product features finding as a domain words extraction process. Using the web as the resource, product features are identified without opinion words.

For the identification of opinion words, the method of the nearest vicinity match is simple and effective. Further, Researchers (Su *et al.,* 2008; Du and Tan, 2009) focus on the association between the features and opinion words by mutual reinforcement approach. Ding and Liu (2007) combines the linguistic rules and opinion aggregation function to determine the semantic orientation.

Our work is different from theirs: the short length reviews are first separately analyzed for their simple structures. A graph model is built to capture the relationship between product features and opinion words. With an iterative reinforcement, product features and opinion words are found in one stage.

3 Bootstrapping

Given the product reviews, the task is to find the product features and opinion words. As observed, the opinion words mostly appear around the features in the review sentences. They are highly dependent on each other. So we adopt bootstrapping method to find both of them in a unified process.

Bootstrapping (Abney, 2002) refers to a problem setting in which one is given a small set of labeled data and a large set of unlabeled data, and the task is to induce a classifier. It has been proved effective in semantic lexicon construction (Widdows and Dorow, 2002; Thelen and Riloff, 2002). In our task, it also includes the following steps:

(i) In the product reviews, choose a small set of features and opinions as "seed words"
(ii) Count co-occurrence of candidates and seed words in the product reviews
(iii) Use a figure of merit based upon these counts to select new seed words
(iv) Return to step (ii) and iterate *n* times

As product features are mostly nouns and noun phrases. Adjectives are normally used to express opinions in reviews. In this paper, we take nouns and noun phrases as the feature candidates, the adjectives as the opinion word candidates, though there are other components that seem like candidates, such as verb phrases as features, nouns as opinion words bringing lots of noises at the same time.

Both product features and opinion words are found in the bootstrapping process, which makes the process complex. Figure 1 shows the framework of the bootstrapping. The bootstrapping process begins with initial seeds of product features and opinion words, as well as the candidate sets of product features and opinion words.

The iterative process includes two steps: the product features selection and opinion words selection. Based on the set of product features, each candidate in the set of opinion candidates is scored by graph model judgment. Then m candidates are added as new elements into the set of opinion words. In the same way, n candidates of feature candidates are added as new product features by measuring them with the set of opinion words. In Figure 1, these two steps are tagged with different lines.

Previous work on product features finding is mainly based on their frequency in the reviews. Though candidate frequency is a good indicator, some of them may not be the product features, such as camera, Canon, people…in digital camera reviews. For the product features, we are more interested in the opinionated features. According to observation, people are accustomed to express their opinions of a product features using opinion words that are located around the feature in the sentence. So we consider the relationship between product features and opinion words, and import a graph model to measure them in the iterative steps.

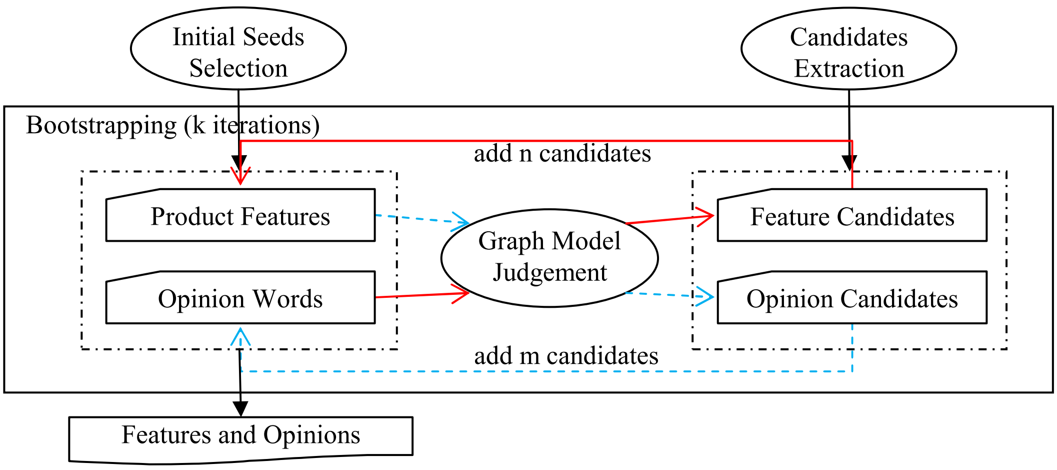

Figure 1: Bootstrapping process

4 Graph Model

In this section we describe how to build a graph to represent the relationship between product features and opinion words. Then an algorithm based on mutual reinforcement between product features and opinion words is proposed to add the most related candidates to the existing collections.

4.1 Graph Building

For the graph, we consider two sets: the set of product features F and the set of opinion words P. A bipartite link between the elements in F and O is built, then a graph G(F, O, R) is constructed. Here, R is the weight of the link, which measures the relationship between product feature and opinion word.

In this paper, we hypothesize that opinion words appear around product features. If an opinion word is co-occur with a product feature within a given distance in a sentence, a bipartite link is built. Otherwise they are considered to be unrelated. We set the weight by the frequency of co-occurrence of the product feature and the opinion word. Figure 2 shows the main idea. There are three relations in the graph: one to many, one to one and one to none. For example, the product feature candidates *time* and *people* have no link with the set of opinion words, so they are not regarded as the real product features. Here, we set the threshold at four. If the relative distance between an opinion word and a product feature is less than four words, then they are regarded to co-occur with each other.

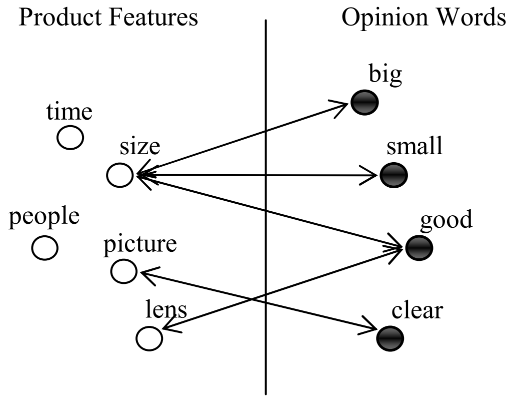

Figure 2: Relationship graph

4.2 Incremental Algorithm

After the graph building, we adopt an algorithm to measure the relative important relationships between the candidates and the existing collection.

Let u be an element in the candidate set U, $N(u)$ be the occurrence of u in the reviews. Let v_i be an element in the existing collection V, $R(u, v_i)$ be the weight of the pair u and v_i in the graph. Then the score of u is calculated in the following:

$$GraphWeight(u) = \sum_{v_i \in V} R(u, v_i) / N(u) \tag{1}$$

When selecting product features, u is a candidate in the set of product feature candidates, v_i is the opinion words in the already selected opinion words set. We adopt a ratio as the weight of product feature u. The numerator is its co-occurrence with the words in the set of opinion words. The denominator is its occurrence in the reviews. For opinion words selection, it is opposite to compute the weight of opinion word with the set of product features.

The ratio used to select new ones tends not to select higher frequency words in the reviews. For instance, suppose that two product feature candidates occur ten and twenty times in the reviews. And they co-occur with the opinion words in the set of opinion words five times. The product feature occurring ten times is more related with opinion words than the one occurring twenty times. We adopt the simple ratio because it is suited to dealing with infrequent occurrences. Suffering from sparse data, it is important to broaden the coverage of candidates from which additional likely candidates can be selected out.

5 Initial Seeds Selection and Candidates Extraction

In the semantic lexicon extraction task, initial seeds are usually selected among the most frequent candidates in the corpus or given by humans. In the reviews, especially in Chinese reviews, people like to express their opinion in short and simple sentence, like the form of “product feature” + “opinion word”. Therefore, we preprocess this type of sentences with rule-based method.

Sentences shorter than the threshold will be listed into short ones. If there is only one adjective and one noun (or noun phrase) in the short sentence, they will be selected as opinion word and product feature. Then initial seed set is constructed. Here, we set the threshold at five words. If there is no initial seed selected from the reviews, the general opinion words such as “good”, “excellent” are inputted as the initial seeds.

The method of candidates extraction is similar with that of (Hu and Liu, 2004). We only consider the adjacent words as product feature candidates, select the candidates with its minimum support larger than the threshold. With redundancy pruning, we remove redundant features which are more likely to be the subset of other candidates.

6 Experiments

6.1 Corpus and Evaluation Metric

We have conducted experiments on the Chinese customer reviews of two kinds of electronic products. Each of them contains 300 reviews. They are extracted from several auto review websites. A human tagger manually read all the reviews and produced the manual lists for product features and opinion words. Table 1 shows the number of manual product features and opinion words tagged from the reviews.

Table 1: Detail of test corpus

Product	No. of manual Features	No. of manual Opinions
Digital camera	162	123
Notebook	176	92

We testify the performance of the proposed techniques from two perspectives:
(i) The effectiveness of product features extraction
(ii) The effectiveness of opinion words extraction
We use precision and recall to evaluate the performance as that of (Hu and Liu, 2004).

6.2 Results and Analysis

For our candidates extraction method is similar with the work of (Hu and Liu, 2004), we take it as the baseline, and give minimum support 1% as the threshold following their work. Table 2 presents precision and recall results of product features extraction. The column of minimum support is the result of the baseline. The column of bootstrapping is the proposed method.

Table 2: Results of the system on product features extraction

Product	Minimum support		Bootstrapping	
	Precision	Recall	Precision	Recall
Digital camera	0.5053	0.2963	0.5083	0.3889
Notebook	0.5288	0.2614	0.5429	0.4545

Compared with the baseline, the results of bootstrapping method improve both the precision and recall for product features extraction. The baseline is mainly to extract the nouns and noun phrases by their frequency, with not only the frequent features in the result but also a lot of errors included in it. There are two types of nouns which are not the real product features. One is the common words such as “时间”(time), “消费”(consumption), and “专业”(profession). They are more likely extracted as features with high frequency in the reviews. The other is the named entity, such as person, brands. They have little probability of being product features, though they are frequent words in the reviews. As a result, it is not enough for this task to use the information of frequency only. It has been proved useful and effective to consider the opinion words in bootstrapping process with its higher precision. The result shows both the recall and precision are improved.

In order to see the influence of the techniques of initial seeds selection and candidates extraction for the system, we evaluate the results at these two steps in Table 3.

The column of initial seeds is the result of product features gotten after the initial seeds selection. The column of candidates is the result of candidates extraction. Here, the threshold in candidates extraction stage is set at 0.5% in order to get more candidates before bootstrapping. And the column of combination is the result of these two steps, which gives the performance of the system before bootstrapping.

Table 3: Results at each steps of the system on product features extraction

Product	Initial seeds		Candidates		Combination	
	Precision	Recall	Precision	Recall	Precision	Recall
Digital camera	0.7500	0.0185	0.3726	0.4877	0.3726	0.4877
Notebook	0.8974	0.1989	0.4140	0.4375	0.4559	0.5284

Table 4: Results of the system on opinion words extraction

Product	Bootstrapping	
	Precision	Recall
Digital camera	0.6696	0.6260
Notebook	0.7200	0.7826

From Table 3, we can see that the performance of initial seeds is highly dependent on the reviews. Its recall is higher with more short sentences in the reviews. Its result is credible with high precision. The initial seeds have no contribution on digital camera reviews, for they are all the frequent candidates and have been included in the frequent candidates. So the column of combination is the same with the column of candidates. Comparing the results on digital camera and notebook, the better the initial seeds get the better whole performance of the results, as shown in Table 2.

For the column of candidates, it shows our aim at getting more candidates with the loss of precision. The column is the result of combining both the initial seeds and candidates as the input for the bootstrapping. Comparing the results of combination and bootstrapping, it indicates that the bootstrapping process improves the precision with a little loss of recall. So it is effective to measure the product feature candidates with its related opinion words. Here, we set the k=30, m=n=3 in the experiment.

Table 4 shows the result of opinion words extraction. The results of opinion words extraction are more acceptable than that of product features extraction, though they are two different tasks. The candidates of opinion words are extracted by the method of vicinity match, the bootstrapping process is just to find more credible ones from this set.

7 Conclusion

In this paper, we probe into the problem of feature level product opinions analysis. We unify the tasks of finding product features and opinion words into one process. With automatically finding the initial seeds, we find the product features and opinion words in iterative steps by the bootstrapping method. A graph model is used to measure the relationship between product features and opinion words. Our experimental results are primary and encouraging, they prove the proposed techniques are effective in performing the task.

Compared with human annotation, there is still a gap needed to be filled by further improving these techniques. For example, the graph building might adopt the rule-based pattern recognization or deeply analysis on the syntactic structures to find more accurate relations between product features and opinion words. The incremental algorithm may adopt different statistic metrics to test its effectiveness in measuring the relationship between product features and opinion words.

References

Abney, S. 2002. Bootstrapping. *Proceedings of the Annual Meeting of the Association for Computational Linguistics,* pp. 360-367.

Ding, X. W. and B. Liu. 2007. The Utility of Linguistic Rules in Opinion Mining. *Proceedings of Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 811-812.

Du, W.F. and S.B. Tan. 2009. An Iterative Reinforcement Approach for Fine-Grained Opinion Mining. *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 486-492.

Hu, M.Q. and B. Liu. 2004. Mining and Summarizing Customer Reviews. *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pp. 168-177.

Hu, M.Q. and B. Liu. 2004. Mining Opinion Features in Customer Reviews. *Proceedings of the American Association for Artificial Intelligence*, pp. 775-760.

Liu, B., M.Q. Hu and J.S. Cheng. 2005. Opinion Observer: Analyzing and Comparing Opinions on the Web. *Proceedings of the International World Wide Web Conference*, pp. 342-351.

Popescu, A. and O. Etzioni. 2005. Extracting Product Features and Opinions from Reviews. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 339-346.

Scaffidi, C., K. Bierhoff, E. Chang, M. Felker, H. Ng and C. Jin. 2007. Red Opal: Product-Feature Scoring from Reviews. *Proceedings of the 8th ACM Conference on Electronic Commerce*, pp.182-191.

Su, Q., X. Xu, H. Guo, Z. Guo, X. Wu, X. Zhang, B. Swen and Z. Su. 2008. Hidden Sentiment Association in Chinese Web Opinion Mining. *Proceedings of the Seventeenth International Conference on World Wide Web*, pp. 959-968.

Thelen, M. and E. Riloff. 2002. A Bootstrapping Method for Learning Semantic Lexicons using Extraction Pattern Contexts. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 214-221.

Widdows, D. and B. Dorow. 2002. A Graph Model for Unsupervised Lexical Acquisition. *Proceedings of the Nineteenth International Conference on Computational Linguistics*, pp. 1093-1099.