2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2010)

Extracting Product Features and Opinions from Product Reviews Using Dependency Analysis

Gamgarn Somprasertsri*
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang
Bangkok 10520, Thailand
s7066001@kmitl.ac.th

Pattarachai Lalitrojwong
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang
Bangkok 10520, Thailand
pattarachai@it.kmitl.ac.th

***Abstract*—In web pages, the reviews are written in natural language and are unstructured-free-texts scheme. Online product reviews is considered as a significant informative resource which is useful for both potential customers and product manufacturers. The task of manually scanning through large amounts of review one by one is computational burden and is not practically implemented with respect to businesses and customer perspectives. Therefore it is more efficient to automatically process the various reviews and provide the necessary information in a suitable form. The task of product feature and opinion is to find product features that customers refer to their topic reviews. It would be useful to characterize the opinions about product. In this paper, we propose an approach to extract product features and to identify the opinions associated with these features from reviews through syntactic information based on dependency analysis.**

Keywords-customer review; opinion extraction; opinion mining; dependency analysis

I. INTRODUCTION

Online customer reviews become a cognitive source of information which is very useful for both potential customers and product manufacturers. Customers have utilized this piece of this information to support their decision on whether to purchase the product. For product manufacturer perspective, understanding the preferences of customers is highly valuable for product development, marketing and consumer relationship management. In a general web page, the reviews are written in natural language scheme and are free of texts with unstructured paradigm. With the great and rapid growth of web contents, customer reviews become available where a customer is able to express opinions on products and services. This trend has seen increasingly attention in sentiment analysis or opinion mining. In the opinion mining community, there are many challenging research topics such as subjectivity classification, sentiment classification, and opinion summarization.

Subjectivity classification is a task for classifying the sentences or the documents which contain opinions from factual, as in [1][2]. It is useful for many natural language processing applications such as question answering, information extraction, and so on. The task of sentiment classification is to judge whether a review expresses a positive or negative opinion. For example, [3][4] developed methods for sentiment classification in document level. The systems assign a positive or negative sentiment for the whole review document. The sentiment of phrases and sentences has also been studied in [5][6]. Even if sentiment classification is useful, it does not find what the reviewer liked and disliked. Review mining and summarization is the task of producing a sentiment summary, which consists of sentences from reviews that capture the author's opinion. Review summarization is interested in features or objects on which customers have opinions. It also determines whether the opinions are positive or negative. This makes it differ from traditional text summarization. Most existing works on review mining and summarization mainly focus on product reviews. For example, [7][8][9] concentrated on mining and summarizing reviews by extracting opinion sentences regarding product features. In another domain, [10] proposed a multi-knowledge based approach for movie review mining and summarization.

In general, mining and summarizing customer reviews involve three tasks: firstly, feature and opinion extraction identifies object features that have been commented in each review; secondly, sentiment assignment determines the polarity of each feature to be positive or negative; and thirdly, summary visualization summarizes the result in order to show this result more effectively.

The high-level problem of opinion summarization addresses how to determine the opinion that an author expresses in natural language text with respect to a certain feature. Let us consider an example of a customer review of a digital camera.

"This camera is very easy to use. The viewing screen is easy to see and very clear. The pictures are clear and good color. To compare other digital cameras we have used, this one if definitely superior and we would highly recommend."

In this example, we can extract several phrases such as *"very easy to use"*, *"viewing screen is easy to see and very clear"*, and *"pictures are clear and good color"*. The phrases represent the customer's opinion rather than facts. Particularly, opinion words such as *"very easy to use"*, *"easy to see"*, *"very clear"*, *"clear"*, and *"good color"* are used to express customer's positive sentiment regarding the product features which are referred by *"to use"*, *"viewing screen"*, and *"picture"*. The task of manually scanning through large amounts of review one by one requires a lot of time and cost for both businesses and customers.

* Corresponding author. Faculty of Informatics, Mahasarakham University, Mahasarakham, Thailand, gamgarns@msu.ac.th

In this study, we address how to associate descriptions of different product features with opinion expressions found in a review. Our goal is to develop ways to establish a correct relationship between the product feature (the topic of the sentiment) and the opinion word (the subjective expression of the product feature). We propose an approach to extract product features and opinions from product reviews through incorporating the syntactic information based on dependency analysis. Our work is mainly focused on product reviews but the methodology in general works for a boarder range of opinions.

The rest of this paper is constructed as follows: Section II presents related work. Section III briefly describes the syntactic information based on dependency analysis, and in Section IV, we illustrate our approach. Experimental results and discussion are given in Section V. Finally Section VI concludes this work.

II. Related Work

There are many methods developed for the solution to opinion summarization problems. Most researchers work on product reviews. Other researchers have studied in another domain [10] proposed a multi-knowledge based approach for movie review mining and summarization.

Hu and Liu's work in [7] can be considered as the pioneer work on feature-based opinion summarization. Their feature extraction algorithm is based on heuristics that depend on feature terms' respective occurrence counts. They use association rule mining based on the Apriori algorithm to extract frequent itemsets as explicit product features (only in the form of noun phrases). In association rule mining, the algorithm does not consider the position of the words in a sentence. In order to remove incorrect frequent features, they use feature pruning that consists of compactness pruning and redundancy pruning. To improve the work over [7], Liu, Hu, and Cheng [11] propose a technique based on language pattern mining to identify product features from pros and cons in reviews in the form of short sentences. They also make an effort to extract implicit features. Moreover, Carenini, Ng and Zwart [12] proposed feature extraction for capturing knowledge from product reviews. In their method, the output of Hu and Liu's system [7] was used as the input to their system, and the input was mapped to the user-defined taxonomy features hierarchy thereby eliminating redundancy and providing conceptual organization.

Popescu and Etzioni [8] developed an unsupervised information extraction system called OPINE, which extracted product features and opinions from reviews. OPINE first extracts noun phrases from reviews and retains those with frequency greater than an experimentally set threshold and then assesses those by OPINE's feature assessor for extracting explicit features. The assessor evaluates a noun phrase by computing a Point-wise Mutual Information score between the phrase and meronymy discriminators associated with the product class.

The work of Yi and Niblack [13] is based on a set of feature term extraction heuristics and selection algorithms for extracting a feature term from product reviews. The feature term is part of a relationship with the given topic, an attribute of a relationship with the given topic, and an attribute of a relationship with a known feature of the given topic. In the first step, they extract a noun phrase with the beginning define Base Noun Phrase (bBNP) heuristics. Then, they select a feature term from the noun phrase using the likelihood score.

Corresponding to these issues, we have carried out some studies on product feature extraction as reported in [14]. In our previous work, we have used combining lexical and syntactic features with the maximum entropy model for extracting the product features. There is an important difference between our approach and Hu and Liu's approach: they do not use both the context information and syntactic structure but we use the syntactic dependency and context information for determining whether the word is a product feature or non-product feature.

To identify the expressions of opinions associated with features. Some researchers considered that a product feature and its opinion words usually co-occur within a certain distance in the sentence. Hu and Liu [7] focused on adjacent adjectives that modify feature nouns or noun phrases. They use adjacent adjectives as opinion words that associated with features. Kim and Hovy [5] explored the following four sizes of regions which may contain both of product features and their opinions. The four regions are: (1) full sentences; (2) words between the opinion holder and the topic; (3) region 2 +/- two words; and (4) from the first word behind the holder to the end of sentences. In other research, Popescu and Etzioni [8] apply manual extraction rules in order to find the opinion words. This idea is similar to that of [7] and [5], but instead of using a window of size or adjacent adjectives they define extraction rules to find the expressions of opinions.

In conclusion, the above methods simply analyze co-occurences of expressions within a short distance or patterns. Some important links between product feature and opinion may be missed. In view of these limitations of the existing approaches, we proposed a method to exploit syntactic information to deal with the semantic relationship between the product feature and the opinion words. Our motivation is that the dependency relation may be useful for extracting the product features and identifying opinions that associate with product features in each sentence.

III. Syntactic Information Basaed on Dependency Analysis

Dependency grammars represent sentence structures as a set of dependency relationships. A dependency relationship is an asymmetric binary relationship between a word called head, and another word called modifier. Fig. 1 shows the dependency tree for a sentence *"The movie mode is also working great."*.

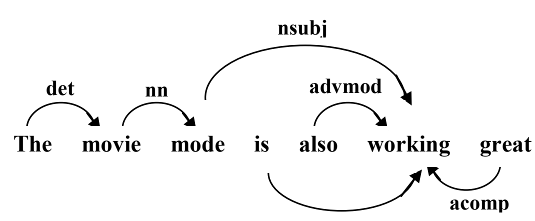


Figure 1. Example of dependency tree

The dependency tree is derived from the syntactic parse tree. We compute the syntactic parse tree by using the Stanford lexicalized parser [15].

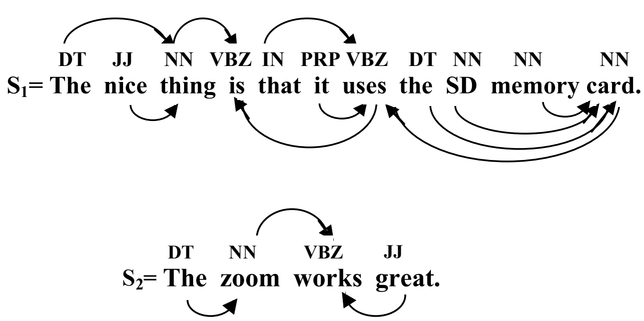
In the relationship of feature and opinion words, the terms have much more varied semantics. There is a large variety of linguistic constructions that express the relation between them. To reduce the variation of linguistic constructions, we assume that the shortest dependency path tracing from a product feature through the dependency tree to an opinion word gives a concrete syntactic structure expressing a relation between the pair. The dependency path and syntactic relationship are used together to find relationships between the product feature and opinion.

We adopted a syntactic relationship consisting of six different relationships as shown in Table 1.

TABLE I. SYNTACTIC RELATIONSHIPS

Relationship	Explanations
Child	Product feature depends on the opinion word
GrandChild	Product feature depends on the word which depends on the opinion word
Sibling	Both opinion word and product feature depend on the same word
Parent	Opinion word depends on the product feature
GrandParent	Opinion word depends on the word which depends on the product feature
Indirect	None of the above relationships

Fig. 2 we show two examples of sentences as dependency trees and the dependency paths linking product features and opinion words in the sentences.



S_1: Feature= "SD memory card" Opinion word = "nice"	**NP → VBZ → VBZ ← NN ← JJ**
S_2: Feature = "zoom" Opinion word = "great"	**NN → VBZ ← JJ**

Figure 2. Example of sentences as dependency tree and dependency paths of relation

IV. SYSTEM ARCHITECTURE

The architectural of the proposed approach consists of three main modules: firstly to perform such as parsing sentences, analyzing noun phrases, and analyzing dependency; secondly to train the ME model; and thirdly to extract product feature-opinion pairs from unlabeled reviews. We first prepare a training data set by manually labeling product feature-opinion pairs of reviews. Detailed steps are given as follows.

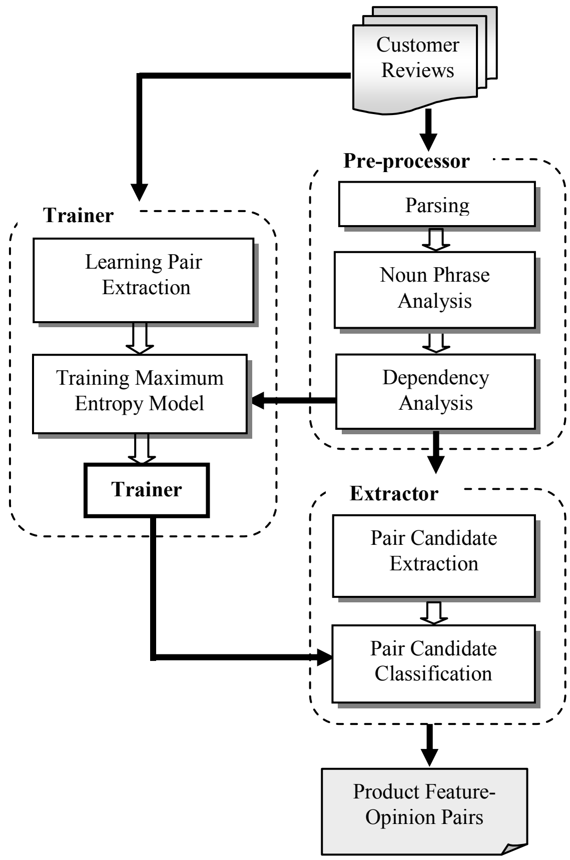


Figure 3. Architecture of System

1) Pre-processing module

To start the pre-processor, reviews are submitted to a pipeline including parsing, noun phrase analysis, and dependency analysis. First, we parse the review sentences with a Stanford lexicalized parser. The output syntactic parse trees are automatically converted into their dependency representations.

In general, most product features indicating words are nouns or noun phrases. Therefore, after parsing the sentence, the next step is to identify a noun phrase as a product feature candidate. We adopted linguistic filtering pattern (e.g. NN, NN NN, JJ NN, NN NN NN, JJ NN NN, JJ JJ NN, NN IN NN, and NN IN DT NN) to extracting noun phrases. Where NN, JJ, DT, and IN are the part-of-speech (POS) tags for noun, adjective, determiner, and preposition respectively defined by the Penn Treebank [16]. Next, for each noun phrase in every dependency parse tree, we exhaustively generate potential syntactic information of noun phrase-adjective word pair.

2) Training module

The training ME model consists of three steps. Firstly, prepare training data which includes pre-processing and product feature-opinion pair annotation. Secondly, extract learning features of each pair in the training data. Thirdly, train the model by using a maximum entropy model. The result of the training model is the weight of each feature function.

To use the maximum entropy to extract product feature-opinion pairs, we define features or important information in order to constrain the model. We denote the features employed for learning as learning features, discriminative from the product features we discussed above. For each pair from the training data, we compute several features automatically. The features are as follow.

Product feature word: potential product feature as a noun or noun phrase.

Opinion word: potential opinion word as an adjective.

Dependency path: The shortest path between feature word and opinion word in a dependency graph.

Syntactic relationship: The classes of syntactic relationship between feature word and opinion word.

3) Extraction module

In order to extract product features and to identify of opinions associated with these features (product feature-opinion pairs), we rely on the observation that there are characteristic words used to describe the product feature and the opinion word. We found that most opinion expressions indicating words are adjectives whereas the nouns build the product features. Therefore, this module extracts pairs that are noun-adjective word pairs. Each such pair becomes a pair candidate. So, there may be more than one pair candidate on a sentence. Next, the trained model is used to predict product feature-opinion pair candidates from unlabeled reviews after parsing. We will simply choose the class with the highest conditional probability.

V. EXPERIMENT AND DISCUSSION

For our experiments, we used reviews on digital cameras from the Amazon web site. We used 1,250 sentences and conducted 5-fold cross validation on that dataset. This set of data was split into a training set of 80% and a testing set of 20%. As pre-processing we parsed this corpus using the Stanford lexicalized parser. We employed the OpenNLP Maxent [17] as our classification tool. The parameters of the maximum entropy model can be trained with 100 iterations of the Generalized Iterative Scaling algorithm. More than 100 iterations would not affect to the increase of accuracy of the parameters. To evaluate the method, we use precision, recall, and F-score to measure the effectiveness of our approach. When dealing with multiple datasets, we adopted the macro average to assess the overall performance across all datasets. The macro average is calculated by simply taking the average performance obtained for each dataset.

We compare the product features-opinion pairs extracted by our approach with co-occurrence approach. We conducted the experiments to compare with adjacent based method [7]. Beside, the pattern based method used by [8] is adopted to compare with our method. The result is compared with two approaches because they are the opinion summarization most relevant to our work and they have evaluated their performance on product review datasets.

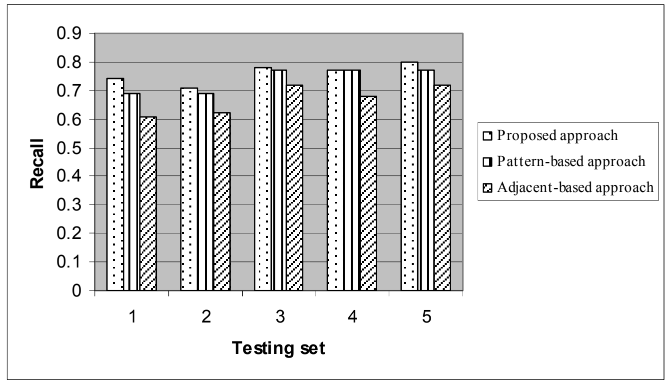


Figure 4. The recall of different approaches on test data

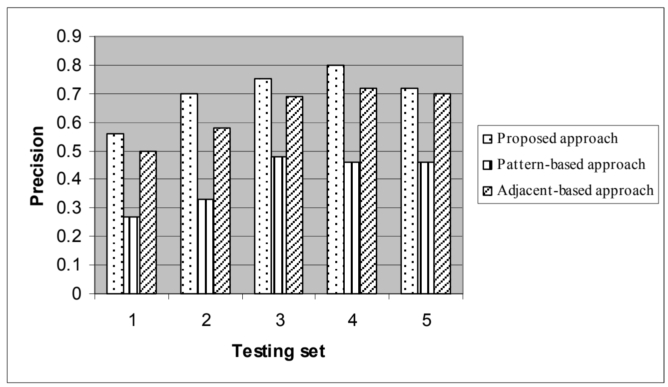


Figure 5. The precision of different approaches on test data

TABLE II. AVERAGE RESULTS ON TEST DATA OF DIFFERENT APPROACHES

Approaches	Precision	Recall	F-score
Proposed approach	**0.71**	**0.76**	**0.73**
Adjacent approach	0.64	0.67	0.65
Pattern approach	0.40	0.74	0.52

Fig. 4 and Fig. 5 show recall and precision of the different approaches on the test sets, respectively. These figures present the results obtained with the proposed approach, which outperforms adjacent-based approach and pattern-based approach. Table 2 demonstrates the average results calculated from five test sets. The macro-averaged F-score of the proposed approach is 0.73, whereas macro-averaged F-score of adjacent-based and pattern-based are 0.65 and 0.52, respectively. Their intuition is that a opinion expression associated with a product feature will occur in its vicinity, whereas our approach takes advantage of the dependency relationship and machine learning. There are two reasons behind the satisfactory performance of the proposed approach. Firstly, the dependency path and syntactic relationship between product feature and opinion expression are useful for identifying the relation between them. Secondly, a maximum entropy classifier may be doing a good job on separating the opinion-relevant product feature pairs from the opinion-irrelevant product feature ones.

For further improvement, we have examined the extraction results manually. It has been found the errors are caused mostly by complex sentences. For example, a complex sentence such as "*It's difficult to take a good picture with this camera." confuses our method, because the sentence that describes negative expression and it's not relevant to extract "good picture*". However, our method can solve the problem which is more than one product feature in a sentence.

VI. Conclusion

In this paper, we tried to solve the problem of extracting the product feature and identifying opinions that associate with product features in each review sentence. We have proposed a novel way to recognize product feature-opinion pairs which uses a probabilistic model with syntactic information based on dependency relationship. The experiments of extracting product features and identifying opinions associated with these features show encouraging results. As part of our future work, we would like to understand the reasons behind the unsatisfactory performance on the complex sentence. The possible improvements could consist of using more natural language processing techniques.

Acknowledgment

The authors would like to thank Mahasarakham University for overseas conference funding.

References

[1] Riloff E, Janyce W, Theresa W. Learning subjective nouns using extraction pattern bootstrapping. In Proc. of ACL SIGNLL Conference, 2003, pp. 25-32.

[2] Hatzivassiloglou V, Janyce W. Effects of adjective orientation and gradability on sentence subjectivity. In Proc. of COLING Conference, 2000.

[3] Pang B, Lee L, Vaithyanathan S. Thumbs up? sentiment classification using machine learning techniques. In Proc. of EMNLP Conference, 2002.

[4] Turney P D. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In Proc. of ACL Conference, 2002, pp. 417-424.

[5] Kim S M, Hovy E. Determining the sentiment of opinion. In Proc. of COLING Conference, 2004, pp. 1367–1373.

[6] Wilson T, Wiebe J, Hoffmann P. Recognizing contextual polarity in phrase-level sentiment analysis. In Proc. of HLT/EMNLP Conference, 2005.

[7] Hu M, Liu B. Mining and summarizing customer reviews. In the Proc. of the ACM SIGKDD Conference, 2004, pp. 168–177.

[8] Popescu A M, Etzioni O. Extracting product features and opinions from reviews. In the Proc. of the EMNLP Conference, 2005, pp. 339–346.

[9] Shi B, Chang K. Mining chinese reviews. In the Proc. of the ICDMW Workshops, 2006.

[10] Zhuang L, Jing F, Zhu X Y. Movie review mining and summarization. In the Proc. of the CIKM Conference, 2006.

[11] Liu B, Hu M, Cheng J. Opinion observer: analyzing and comparing opinions on the web. In Proc. 14th Int. Conf. World Wide Web, Chiba, Japan, 2005, pp.342-351.

[12] Carenini G, Ng R T, Zwart E. Extracting knowledge from evaluative text. In Proc. 3rd Int. Conf. Knowledge Capture, 2005, pp.11–18.

[13] Yi J, Niblack W. Sentiment mining in WebFountain. In Proc. 21st Int. Conf. Data Engineering, 2005, pp.1073-1083.

[14] Somprasertsri G, Lalitrojwong P. Automatic Product Feature Extraction from Online Product Reviews Using Maximum Entropy with Lexical and Syntactic Features. In Proc. IEEE Int. Conf. Information Reuse and Integration, 2008, pp.250–255.

[15] Stanford Lexicalized Parser - a probabilistic lexicalized NL CFG parser, 2006. http://nlp.stanford.edu/downloads/lex-parser

[16] Marcus M P, Santorini B, Marcinkiewicz M A. Building a large annotated corpus of English: the penn treebank. Computational Linguistics, 1993, 19.

[17] Baldridge J, Morton T, Bierner G. Java-based opennlp maximum entropy package: Maxent. http://maxent.sourceforge.net