Journal of Universal Computer Science, vol. 16, no. 6 (2010), 938-955
submitted: 15/9/09, accepted: 4/3/10, appeared: 28/3/10

Mining Feature-Opinion in Online Customer Reviews for Opinion Summarization

Gamgarn Somprasertsri
(King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand
s7066001@kmitl.ac.th)

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

Abstract: Online customer reviews is considered as a significant informative resource which is useful for both potential customers and product manufacturers. In web pages, the reviews are written in natural language and are unstructured-free-texts scheme. 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 high-level problem of opinion summarization addresses how to determine the sentiment, attitude or opinion that an author expressed in natural language text with respect to a certain feature. In this paper, we dedicate our work to the main subtask of opinion summarization. The task of product feature and opinion extraction is critical to opinion summarization, because its effectiveness significantly affects the performance of opinion orientation identification. It is important to properly identify the semantic relationships between product features and opinions. We proposed an approach for mining product feature and opinion based on the consideration of syntactic information and semantic information. By applying dependency relations and ontological knowledge with probabilistic based model, the result of our experiments shows that our approach is more flexible and effective.

Keywords: Opinion Mining, Opinion Summarization, Text Mining, Customer Feedback, Dependency Grammars, Maximum Entropy
Categories: I.2.7, H.2.8, H.3.1, H.3.5

1 Introduction

Recently, a number of online shopping customers have dramatically increased due to the rapid growth of e-commerce, and the increase of online merchants. To enhance the customer satisfaction, merchants and product manufacturers allow customers to review or express their opinions on the products or services. The customers can now post a review of products at merchant sites, e.g., amazon.com, cnet.com, and epinions.com. These online customer reviews, thereafter, 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. In comparison, numerical and categorical data are well structured, which make them relatively easy to handle. On the contrary, customer reviews are unstructured data. To be handled, these data demand knowledge from different areas, e.g., database, information retrieval, information extraction, machine learning, and natural language processing. 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 the task of classifying the sentences or the documents which contain opinions from factual, for instance in [Riloff, 03]. 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, Pang et al [Pang, 02] 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 [Wilson, 05]. Even if sentiment classification is useful, it does not imply the underlining information, i.e. what the reviewer liked and disliked. Opinion summarization [Hu, 04, Popescu, 05, Gamon, 05, Yi, 05, and Carenini, 06] is the task of producing a sentiment summary, which consists of sentences from reviews that capture the author's opinion. The summarization task is interested in features or objects on which customers have opinions. This is different from traditional text summarization that involves reducing a larger corpus of multiple documents into a short of paragraph that conveys the meaning of text. The product reviews on the Web are in three formats [Liu, 07]:

- Format 1 - Pros, cons and the detailed review: The reviewers describe pros and cons in the form of short phrases and also write the detail of reviews separately.
- Format 2 - Pros and cons: The reviewers describe pros and cons in the form of full sentences separately.
- Format 3 - Free format: The reviewers write the reviews in the free form that no separation of pros and cons.

In format 1, pros and cons usually consist of short phrases and incomplete sentences, for example "*pros: fabulous photo quality, large LCD, great battery life, great features*". The reviews of format 2 and 3 usually consist of long sentences and complete sentences, for example "*I have taken hundreds of photos with it and i continue to be amazed by their quality*". However, the product features and opinions extraction from reviews of format 2 and 3 is more challenge because the complete sentences are more complex and contain a large amount of irrelevant information. 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. Therefore, a good summarization system can help them in getting the required and relevant information without going through all the reviews present on the site.

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*".

This study, we address the specific problem that is how to associate descriptions of different product features with opinions found in reviews of format 3. The task is not only technically challenging – applying natural language processing, but also very useful in practice. In feature-opinion mining, most of the existing researches usually depend on the co-occurrence of product features and opinion words. The methods acquire relations based on fixed position of words. However, the approaches are not effective for many cases. Look at the following review sentences.

(1) It has <u>movie mode</u> that works *good* for a digital camera.

(2) It is *great* having the <u>LCD display</u>.

(3) I bought my <u>canon g3</u> about a month ago and i have to say i am very *satisfied*.

(4) The *nice* thing is that it uses the <u>SD memory card</u>.

In these samples, the words in underline are product feature and the words in italic are opinion. The approach of co-occurrence of words is not the way to deal with this kind of problem. In this paper, 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). The basic purpose of our approach is to mine the product features and opinion words that associate with product features in each sentence.

The remainder of the paper is organized as follows. Section 2 describes previous work on the task of product feature and opinion extraction. Section 3 introduces dependency relations for product feature-opinion mining. Section 4 discusses how to mine product feature-opinion pairs from online customer reviews. Section 5 presents and discusses the experimental results. Finally, Section 6 concludes our work.

2 Previous Work

Opinion summarization essentially consists of three main tasks. The first task of opinion summarization is to extract the features of a product and to identify opinions that associate with product features in each sentence and then identify the opinion orientation. Finally produce a structured sentence list according to the feature-opinion pairs as the summary. The task of product feature and opinion extraction is critical to opinion summarization, because its effectiveness significantly affects the performance of opinion orientation identification. Many previous works [Hu, 04, Popescu, 05, Liu, 05, Yi, 05, and Zhuang, 06] usually depend on the co-occurrence of words.

Hu's work in [Hu, 04] 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). Association rule is an implication of the form X=>Y, where X and Y are database itemsets. Two measures have been developed to evaluate association rules, which are support and confidence. Itemsets that have support at least equal to minimum support are called frequent itemsets [Daly, 04]. In Hu's work, each resulting frequent itemset is a possible feature. They define an itemset as frequent if it appears in more than 1% minimum support of the review sentences. In this approach, 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 Hu et al, Liu et al [Liu, 05] proposed 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 et al [Carenini, 05] proposed feature extraction for capturing knowledge from product reviews. In their method, the output of Hu's system 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. To identify the expressions of opinions associated with features. Hu et al focused on adjacent adjectives that modify feature nouns or noun phrases. They use adjacent adjectives as opinion words that associated with features. For each sentence in reviews, if it contains any frequent feature, extract the nearby adjective. It is considered an opinion.

Popescu et al [Popescu, 05] 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. Popescu et al apply manual extraction rules in order to find the opinion words. This idea is similar to that of Hu et al [Hu, 04], but instead of using adjacent adjectives they define extraction rules to find the expressions of opinions. For example,

If $\exists(M, NP = \text{f}) \rightarrow \text{po} = M$: (expensive) scanner

If $\exists(S = \text{f}, P, \text{O}) \rightarrow \text{po} = O$: Lamp has (problems)

If $\exists(S, P, O = \text{f}) \rightarrow \text{po} = P$: I (hate) this scanner

If $\exists(S = \text{f}, P) \rightarrow \text{po} = P$: Program (crashed)

M = modifier, NP = noun phrase, S = subject, P = predicate, O = object, f = feature and po = potential opinion

Yi et al [Yi, 05] developed a set of feature term extraction heuristics and selection algorithms for extracting a feature term from product reviews. The feature term is a part of relationship with the given topic, an attribute of relationship with the given topic, and an attribute of 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. As a processing step to opinion extraction, they utilized some patterns based on sentiment extraction pattern such as

<“impress” + PP(by;with)>: the target or feature is subject phrase (SP) and the opinion is “impress”

<“take” + OP SP>: the target or feature is subject phrase (SP) and the opinion is object phrase (OP)

Zhuang et al [Zhuang, 06] studied in movie review domain. They proposed a multi-knowledge based approach for movie review mining and summarization. They used the keyword list and dependency relation templates together to mine explicit feature-opinion pairs. For example,

NN – amod – JJ

NN – nsubj – JJ

NN – nsubject – VB – dobj - NN

In conclusion, the above methods acquire relations based on explicit adjacency. They 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 and semantic 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. In addition, the idea behind this method is to use machine learning to automatically replace manual extraction of rules to identify the expressions of opinions associated with features.

3 Dependency Relations for Feature-Opinion Mining

Dependency grammars represent sentence structures as a set of dependency relationships. A dependency relationship [Melcük, 87] is an asymmetric binary relationship between a word called head or governor, and another word called modifier or dependent. The dependency of words will form a dependency tree. The syntactic structure of a sentence consists of dependencies shown in Figure 1.

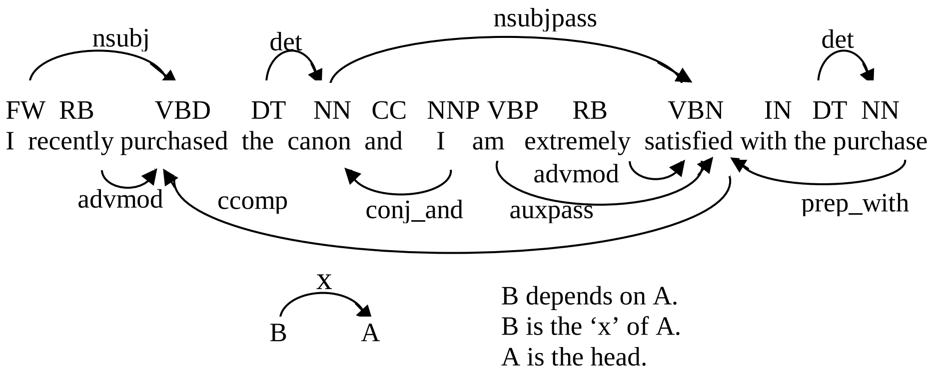


Figure 1: The syntactic structure of a sentence consists of dependencies

Each relationship has a word as head. The other is the dependent. A word has one head at most. However, a word may have several dependents.

With these relations defined by the dependency tree, we find there are five relations for mining product feature and opinion as following. PF refers to product features, O refers to opinion words and A refers to ancestors.

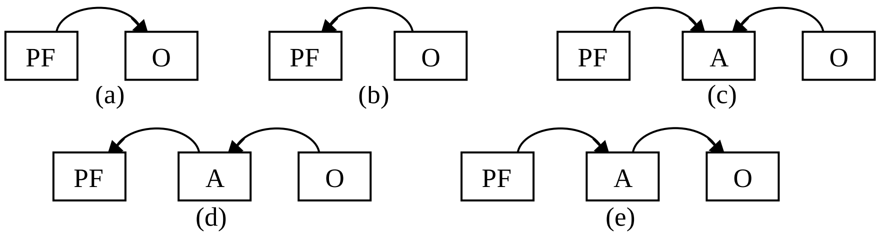


Figure 2: The relations of dependency sub-trees for product feature-opinion mining

1) **Child**: The product features are in the children as (a) in Figure 2. In such relation, the product feature is the subject or object of the verbs and the opinion word is a verb or a complement of a copular verb, for example

(1) "*I like this camera.*"

Dependency relation:

{nsubj(like-2, I-1), det(camera-4, this-3), dobj(like-2, camera-4)}

The dependencies are written abbreviated_relation_name(head, dependent) where the head and the dependent are words in the sentence to which the word number in the sentence is append. In the brackets, the first word is the parent and the second word is the child. In (1), the word "*camera*" is the product feature. The word "*like*" is the opinion. The "*camera*" is the direct object of the verb "*like*".

(2) "*The battery life is good.*"

Dependency relation:

{det(life-3, The-1), nn(life-3, battery-2), nsubj(good-5, life-3), cop(good-5, is-4)}

The phrase "*battery life*" is the product feature. The word "*good*" is the opinion. The "*battery life*" is a noun phrase which is the subject. The "*good*" is the complement of the copular verb.

2) **Parent**: The product features are in the parents as (b) in Figure 2. In such relation, the opinion words are in the modifiers of product features, which include adjectival modifier, relative clause modifier, etc., for example

(3) "*I have found that this camera take incredible pictures.*"

Dependency relation:

{nsubj(found-3, I-1), aux(found-3, have-2), complm(take-7, that-4),
det(camera-6, this-5), nsubj(take-7, camera-6), ccomp(found-3, take-7),
amod(pictures-9, incredible-8), dobj(take-7, pictures-9)}

The word "*picture*" is the product feature. The word "*incredible*" is the opinion which is the adjectival modifier of a word "*picture*".

3) **Sibling**: The product features and the opinion words are in the children of the same ancestor as (c) in Figure 2. In such relation, the opinion word may also be in an adverbial modifier, a complement of the verb, or a predicative, for example

(4) "*The pictures some time turn out blurry.*"
Dependency relation:
{det(picture-2, The-1), nsubj(turns-5, picture-2), det(time-4, some-3), dep(turns-5, time-4), dep(blurry-7, out-6), acomp(turns-5, blurry-7)}

The word "*picture*" is the product feature which is the subject. The word "*blurry*" is the opinion which is the adverbial modifier of a verb.

4) **Grand Parent**: The product features are in the parents of the words that are in the parents of the opinion words as (d) in Figure 2. In such relation, the opinion words are adjectival complement of modifiers of product features, for example

(5) "*It has movie mode that works good for a digital camera.*"
Dependency relation:
{nsubj(has-2, It-1), nn(mode-4, movie-3), dobj(has-2, mode-4), rel(works-6, that-5), rcmod(mode-4, works-6), acomp(works-6, good-7), det(camera-11, a-9), amod(camera-11, digital-10), prep_for(good-7,camera-11)}

The phrase "*movie mode*" is the product feature. The word "*good*" is the opinion which is the adverbial complement of relative clause modifier of noun phrase "*movie mode*".

5) **Grand Child**: The product features are in the children of the words that are in the children of the opinion words as (e) in Figure 2. In such relation, the product feature is the subject or object of the complements and the opinion word is a verb or a complement of a copular verb, for example

(6) "*It's great having the LCD display.*"
Dependency relation:
{nsubj(great-3, It-1), cop(great-3, 's-2), xcomp(great-3, having-4), det(display-7, the-5), nn(display-7, LCD-6), dobj(having-4, display-7)}

The phrase "*LCD display*" is the product feature. The word "*great*" is the opinion which is the complement of a copular verb. The "*LCD display*" is the object of a clausal complement.

4 Mining Product Feature-Opinion

In this section, we described our methods to mine product feature-opinion from online customer reviews. The product feature can be a brand name, a model name of a commodity, a property, a part, a feature of a product, a related concept, or a part of a related concept [Popescu, 05]. Section 4.1 explains some pre-processing steps. The core methods are described in Section 4.2 and Section 4.3. Figure 3 gives the architecture overview for our approach.

4.1 Pre-Processing

To start the pre-processing, reviews are submitted to a pipeline including parsing and dependency analysis. Firstly, we parse the review sentences by using the Stanford Parser. After that we exhaustively generate a dependency tree as shown in Figure 1.

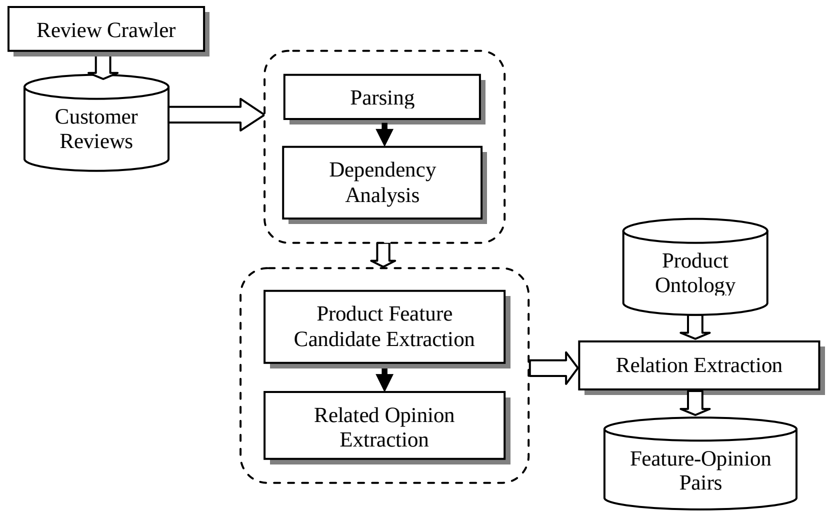


Figure 3: The architecture of our approach

4.2 Product Feature-opinion Candidate Extraction

When mining product feature-opinion, we first identifies product feature on which many customers have expressed their opinions. If a product feature appears, we will search for the related opinions and product features.

4.2.1 Product Feature Candidate Extraction

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 and General Inquirer Dictionary [Stone, 66] to extracting product feature candidate. We also discard stop words to reduce noise. A definite linguistic filtering pattern is a noun phrase as the following patterns:

-NN,
-NNNN, JJ NN
-NN NN NN, JJ NN NN, JJ JJ NN, NN IN NN
-NN IN DT NN

where NN, JJ, DT, and IN are the POS tags for noun, adjective, determiner, and preposition respectively defined by the Penn Treebank [Marcus 93]. Algorithm 1 demonstrates the process to extract all the product feature candidates in reviews.

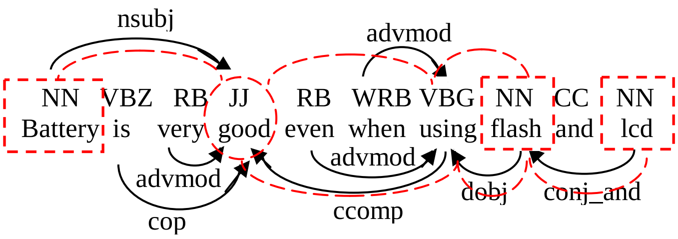
4.2.2 Related Opinion Extraction

This step is to identify product feature-opinion candidates. For each product feature candidate in every dependency parse tree, we search for the related opinion words. Some adjectives and verbs may be used for both favorable and unfavorable predictes. Thus, this paper uses adjectives and verbs as opinion words. The procedure of extracing opinions as following manner (Algoritm 2).

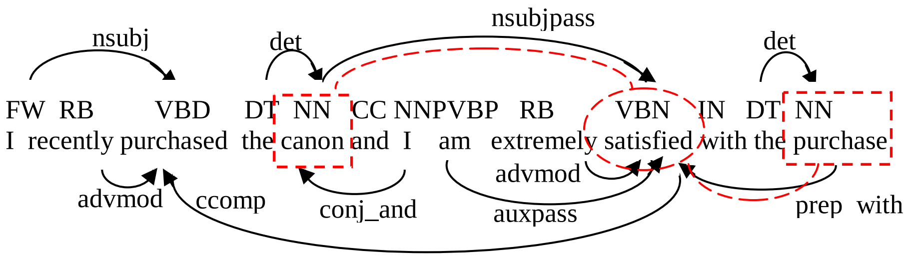
Algorithm 1. Pseudo-Code for extracting product feature candidates

```
//Input:    S  – Set of tagged sentences; s = s1,s2,…,sm
             P  – Set of noun phrase patterns
            GI  – Set of word in GI dictionary
//Output:   PS  – Set of product feature candidates
PS = ø
For each tagged sentence sn ∈ S
    PC = ø
          For i=1 to end of sentence sn
             If   i  < Length(sn) – 2  Then   x = 3
              Else If   i  = Length(sn) – 2  Then   x = 2
                   Else If   i  = Length(sn) – 1  Then   x = 1
                         Else x = 0
                          End
                   End
             End
              For j = x to 0
                   GT = Ti to Ti+j          /* POS Tag of wordi to wordi+j of sn */
                   GW = wordi to wordi+j
                   If   GT ∈ P and GW ∉ GI then
                     i = i+j
                       PC = PC ∪ GW
                       Break
                   End
               End
           End
     PS = PS ∪ PC
End
```

Consider the following examples: "*Battery is very good even when using flash and LCD.*" and "*I recently purchased the Canon and I am extremely satisfied with the purchase.*" Figure 4 shows the procedure of product feature-opinion candidate extraction. Firstly we find the product features, and then find the opinions through the dependency tree (in the manner as Algorithm 2). In these samples, the words in circle shape are the effective opinions of the product feature candidates in square shape. We can extract several pairs such as *(battery, good)*, *(flash, good)*, *(lcd, good)*, *(cannon, satisfied)*, and *(purchase, satisfied)*. Each of such pairs becomes a product feature-opinion candidate. After product feature-opinion candidate extraction, we predict the opinion-relevant product feature relation using the probabilistic based model.



(a) "*Battery is very good even when using flash and LCD.*"



(b) "*I recently purchased the Canon and I am extremely satisfied with the purchase.*"

Figure 4: Example product feature-opinion candidate extraction

Algorithm 2. Pseudo-Code for extracting the product feature-opinion pairs

```
//Input      DT  – Set of dependency trees
             PS  – Set of product feature candidates in each sentence
             GI  – Set of word in GI dictionary
//Output     FOS – Set of product feature-opinion pairs
PairExtract(dt_i, ps_i)  /* Return set product feature-opinion pairs of each dependency tree */
FO = ø
  For m=1 to end of product feature candidate ps_i
     Rnode  =  ps_i(m)                  /* Initial product feature candidate to root node */
     f = FirstVisit(dt_i,Rnode)         /* Find first neighbor, return -1 if no neighbor */
     While (f ~= -1)
        If   (neighbor is adjective) or (neighbor is adveb and ∈ GI)  then
           pair = [Rnode,neigbor]       /* product feature-opinion pair */
           FO = FO ∪ pair
           f = -1
        Else
           f = NextVisit(dt_i,Rnode)    /* Find next neighbor, return -1 if no neighbor */
        End
     End
  End
PairExtract(DT, PS, GI)                 /* Return set of product feature-opinion pairs */
  For each dependency tree dt_i ∈ DT
       FO = PairExtract(dt_i, ps_i)
  End
```

4.3 Predicting a Relation by Maximum Entropy Model

Maximum entropy model was first described by Jaynes [Jaynes, 57]. The maximum entropy model chooses the least biased distribution, which maximizes uncertainty in the distribution subject to given constraints [Tan, 07]. For our work, we use maximum entropy model to predict the opinion-relevant product feature relation. This task can be re-formulated as a classification problem, in which the task is to observe some linguistic context $x \in X$ and predict the correct linguistic class $y \in Y$. We can design classes such as opinion-relevant product feature and opinion-irrelevant product feature. We can implement classifier cl: $X \rightarrow Y$ with a conditional probability model by simply choosing the class y with the highest conditional probability p in the context x:

$$cl(x) = \arg\max_{y} p(y \mid x) \quad (1)$$

The conditional probability $p(y|x)$ is defined as follows [Ratnaparkhi, 98]:

$$p(y \mid x) = \frac{1}{Z(x)} \prod_{i=1}^{k} \alpha_i^{f_i(x,y)} \quad (2)$$

$$Z(x) = \sum_{y} \prod_{i} \alpha_i^{f_i(x,y)} \quad (3)$$

where y refers to the outcome, x is the history (or context), k is the number of features and $Z(x)$ is a normalization factor to ensure that $\sum_y p(y \mid x) = 1$. Each parameter α_i corresponds to one feature f_i and can be interpreted as a weight for that feature.

We use the Generalized Iterative Scaling (GIS) algorithm [Darroch, 72] to estimate parameters or weights of the selected features. Under the maximum entropy framework, the probability for a class y and object x depends solely on the features that are active for the pair (x, y), where a feature is defined here as a function f: $X \times Y \rightarrow \{0, 1\}$ that maps a pair (x, y) to either 0 or 1. The feature is defined as follows:

$$f_{cp,y'}(x, y) = \begin{cases} 1 & \text{if } y = y' \textit{ and } cp(x) = \textit{true} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where $cp(x)$ is contextual predication that returns true or false, corresponding to the presence or absence of useful information in some context, or history $x \in X$. For example, to predict which the class of product feature-opinion candidate belongs (as shown in Table 1). The classifier considers dependency relation of the target product feature-opinion candidate. Supposing the opinion word depends on the product feature. The relation of the target product feature-opinion pair is parent, a feature function can be set as follows:

$$f_i(x_j, y_j) = \begin{cases} 1 & \text{if } y_j = YES \text{ and } \mathrm{Re}\,l(PARENT) = true \\ 0 & \text{otherwise} \end{cases}$$

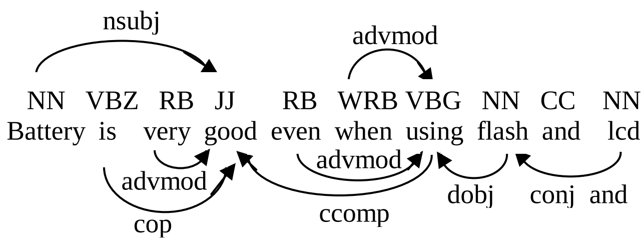
Class	**Description**
YES	Feature-opinion pair claimed to be opinion-relevant product feature
NO	Feature-opinion pair claimed to be opinion-irrelevant product feature

Table 1: Classes defined for the classification task

In order to use the maximum entropy to classify product feature-opinion candidates, we define important information in order to constrain the model. We use syntactic information to classifying product feature-opinion pair. One of the chalenges for this problem is dut to the wide variation of surface text. 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. Our idea is to learn such patterns from the dependency paths for each relationship. Furthermore, we attempt to capture relating product feature and opinion using dependency relations between them. For our work, we adopted a dependency relation consisting of six different relations as presented in Table 2.

Relation	**Description**
Parent	Opinion depends on the product feature.
Child	Product feature depends on the opinion.
Sibling	Both opinion and product feature depend on the same word.
Grandparent	Opinion depends on the word which depends on the product feature.
Grandchild	Product feature depends on the word which depends on the opinion.
Indirect	None of the above relations

Table 2: Dependency relations for product feature-opinion mining



Pair 1 (battery, good), path: NN→JJ, relation: Child
Pair 2 (flash, good), path: NN→VB→JJ, relation: GrandChild
Pair 3 (lcd, good), path: NN→NN→VB→JJ, relation: Indirect

Figure 5: Example of dependency paths and dependency relations

Let us consider the dependency tree of example "*Battery is very good even when using flash and LCD*" as shown in Figure 5. We can extract several product feature-opinion candidates such as "*battery, good*", "*flash-good*", and "*lcd-good*". Each such pair becomes a pair candidate. For effective relation extraction, we group product features by using product ontology that we will describe in next section. The maximum entropy model is used to predict opinion-relevant product feature. Firstly, for each pair, we compute several features automatically. We denote the features employed for learning as learning features, discriminative from the product features we discussed above. The features are opinions, grouped product features, dependency paths and dependency relations. We will simply choose the class with the highest conditional probability *p* according to Equation 1.

5 Product Ontology

In an abstract sense, an online customer review is a list of those product features or concepts that a customer likes or dislikes. Different customers will often refer to identical product features using inconsistent or incompatible terminology. Furthermore, customers might refer to a particular feature in different ways. For example, "*memory card*", "*compact flash*", "*compactflash*", "*CF card*", and "*memory stick*" are string for describing "*removable memory*". To solve this issue, we use sematic information encodsed in ontology. Figure 6 illustrates a part of our ontology.

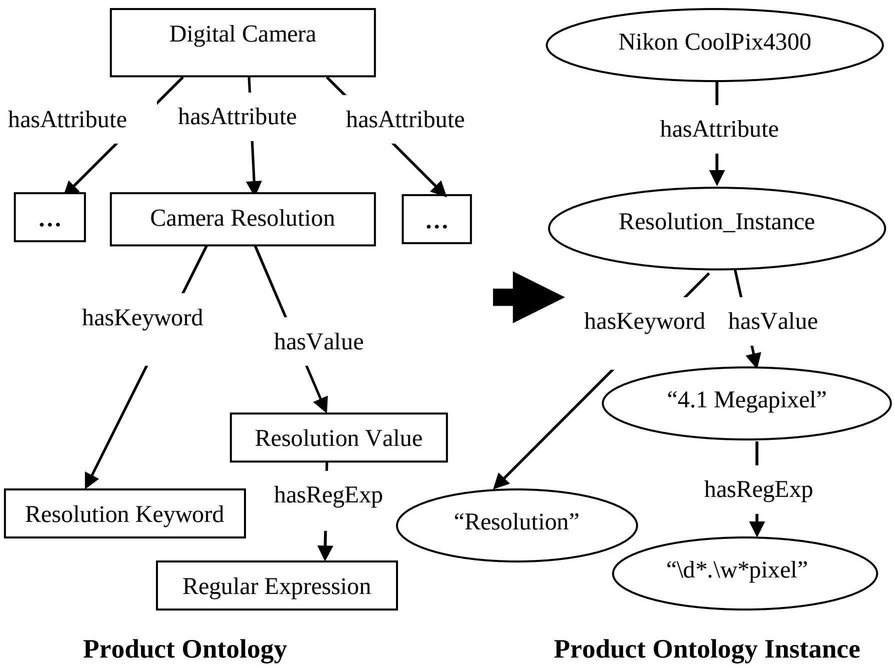


Figure 6: Fragments of product ontology and product ontology instance

Ontology plays a pivotal role here by providing a source of shared and precisely defined terms that can be used in such meta-data [Bhatt, 06]. Ontology can create an agreed-upon vocabulary for sharing knowledge, exchanging information, and eliminating ambiguity [Xue, 09]. In our work, we use ontology to normalize the language for distinguishing between different product features. In this paper, we design ontology by applying core ontology of Jannach et al [Jannach, 09]. Product ontology is expressed in a tree-hierarchy of concepts. We manually construct product ontology by integrating manufacturer product descriptions and terminologies in customer reviews. The root of the tree represents the product. Subsequent sub-trees represent attributes of the product.

According to the problem of describing a product feature in different ways, it is important to group terminologies with similar meaning together. Our work uses a simple method. The basic idea is to employ product ontology to group terminologies using simple regular expression patterns as showed on Figure 6. If a product feature candidate dose not matches any regular expression, using itself as a grouped product feature.

6 Experimental Settings

6.1 Data and Evaluation

The dataset used in our experiments included two sets on digital cameras from Hu's previous work [Hu, 04] and digital camera reviews from *Amazon.com*. The sentences in the dataset have manually generated tags indicating product features and opinions. We conducted 5-fold cross validation on that dataset. We employed the OpenNLP maximum entropy package as our classification tool.

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. Therefore, the definitions of precision, recall and F-score are as following.

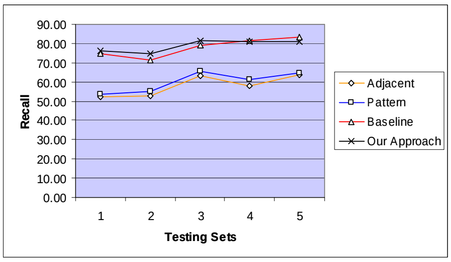
$$\text{Precision} = \frac{\text{PC}}{\text{PM}} \qquad \text{Recall} = \frac{\text{PC}}{\text{PT}} \qquad \text{F - score} = \frac{2 \text{ x Precision x Recall}}{\text{Precision} + \text{Recall}}$$

PC = number of correctly mined product feature-opinion pairs;
PM = number of all mined product feature-opinion pairs;
PT = number of all correct product feature-opinion pairs.

6.2 Experimental Results and Discussion

In order to evaluate our method on the task of mining product features and opinions, we used a dataset of 1250 sentences described in Section 6.1. We randomly divided the dataset into five equal-sized folds. We used four folds as the training data and one fold as the testing data. We conducted the experiments to compare with Hu's approach (adjacent based method). Beside, the patterns used by Popesecu's approach (pattern based method) are adopted to compare with our method. The result is compared with Hu's approach and Popesecu's approach because they are the opinion summarization most relevant to our work and they have evaluated their performance on product review datasets. The product feature candidates are extracted by the method described in Section 4.2.1. Baseline is our approach without using product ontology.

We use precision, recall and F-score to evaluate performances. Five-fold cross-validation results of extracting product features and opinions are shown in Figure 7, Figure 8, and Figure 9. Table 3 shows the average results of different methods. The results show that our method outperforms others in the precision, the recall and the F-score. It shows that our method is superior to adjacent based method and some pattern based method with two main reasons. One reason, in adjacent based method, for each product feature, its nearest opinion word is used to construct the product feature-opinion pair. It produces many invalid pairs due to the complexity of sentences in product reviews. A second important reason, the pattern based method could not discover the relations between product features and opinions from the complex sentences. Beside, our approach performs a little better than non-ontology in recall and F-score because the right words dose not include in the ontology at design time.



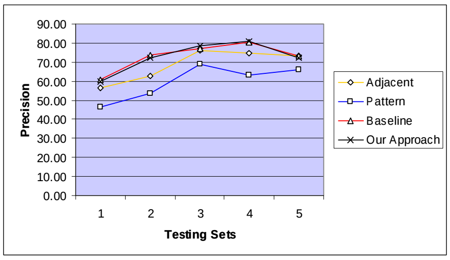


Figure 7: Recall of different methods *Figure 8: Precision of different methods*

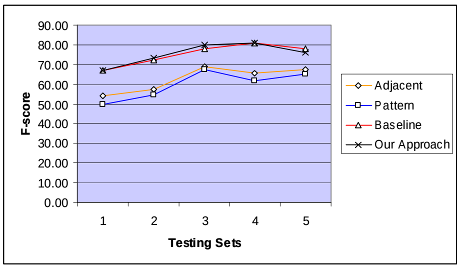


Figure 9: F-score of different methods

Methods	**Precision (%)**	**Recall (%)**	**F-score (%)**
Adjacent Based	68.65	57.93	62.69
Pattern Based	59.65	59.95	59.72
Baseline	73.12	77.98	75.34
Our Approach	72.65	78.77	75.45

Table 3: Average results for total performance with 5-fold cross validation on dataset

Table 4 shows the comparison of extracting product features and opinions of each method on simple and complex sentences. It is notable that the adjacent based and pattern based method can extract product features and opinions only from simple sentences. Interestingly, our method can extract product features and opinions form both type of sentences. Our method focus on opinion as adjective and verb exclude noun as in (3) because most of opinions as an adjective or a verb. In summary, we conclude that the approach is more flexible and effective than the adjacent based approach and opinion pattern based approach.

Structure	**Example**	**Methods**		
		Adjacent Based	Pattern Based	Our Approach
Simple	(1) There is a *great* camera.	Yes	Yes	Yes
	(2) The optical zoom works *great*.	Yes	Yes	Yes
	(3) Lens has *problems*.	No	Yes	No
	(4) I *like* this camera.	No	Yes	Yes
	(5) It is *great* having the LCD display.	No	No	Yes
Complex	(6) It has movie mode that works *good* for a digital camera.	No	No	Yes
	(7) I bought my canon g3 about a month ago and i have to say i am very *satisfied*.	No	No	Yes
	(8) The *nice* thing is that it uses the SD memory card.	No	No	Yes

Table 4: The comparison of extracting product features and opinions of difference methods

7 Conclusion and Future Work

In this paper, a dependency and semantic based approach is proposed for mining opinions from online customer reviews. We focused on extracting relations between product features and opinions. We have proposed a novel way to capture the actual relations of product features in sentences regardless the distance from them to opinions. Experimental results show the effectiveness of the proposed approaches.

As part of our future work, we would like to understand the reasons behind the unsatisfactory performance on the complex sentence. For example, a complex sentence such as "*With the automatic settings, i really haven't taken a bad picture yet.*" confuses our method, because the sentence that describes positive expression and it's not relevant to extract "*bad picture*". The possible improvements could consist of using more natural language processing techniques. Finally, we would also investigate self-learning methods for classification that may provide a mechanism for further reducing the amount of labeled data required to produce highly accurate results.

Acknowledgements

The authors would like to thank the reviewers for their insightful comments and the valuable suggestions.

References

[Bhatt, 06] Bhatt, M., Flahive, A., Wouters, C., Rahayu, W., Taniar, D.: MOVE: A Distributed Framework for Materialized Ontology View Extraction, Algorithmica, Vol. 45, No. 3 (2006), 457-481.

[Carenini, 05] Carenini, G., Ng, R. T., Zwart, E.: Extracting Knowledge from Evaluative Text, In Proc. 3rd Int. Conf. Knowledge Capture, 2005, 11–18.

[Carenini, 06] Carenini, G., Ng, R., Pauls, A.: Multi-document Summarization of Evaluative Text, In Proc. 11th Conf. European Chapter of the ACL, 2006.

[Daly, 04] Daly, O., Taniar, D.: Exception Rules Mining Based on Negative Association Rules, In Proc. Int. Conf. ICCSA, Lecture Notes in Computer Science, Vol. 3046 (2004), 543-555.

[Darroch, 72] Darroch, J., Ratcliff, D.: Generalized Iterative Scaling for Log-linear Model, The Annals of Mathematical Statistics, Vol. 43, No.5 (1972), 1470–1480.

[Gamon, 05] Gamon, M., Aue, A., Corston-Oliver, S., Ringger, E.: Pulse: Mining Customer Opinions from Free Text, In Proc. 6th Int. Symp. Advances in intelligent data analysis, 2005, 121–132.

[Hu, 04] Hu, M., Liu, B.: Mining and Summarizing Customer Reviews, In Proc. 10th Int. Conf. Knowledge Discovery and Data Mining, Seattle, WA, 2004, 168–177.

[Jannach, 09] Jannach, D., Shchekotykhin, K., Friedrich, G.: Automated Ontology Instantiation from Tabular Web Sources-The ALLRIGHT System, Web Semantics: Sci.Serv.Agents World Wide Web, 2009.

[Jaynes, 57] Jaynes, E T.: Information Theory and Statistical Mechanics, Physical Reviews, Vol. 106 (1957), 620-630.

[Liu, 05] 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, 342-351.

[Liu, 07] Liu, B.: Web Data Mining Exploring Hyperlinks, Contenrs, and Usage Data, Springer, New York, 2007.

[Melcük, 87] Melcük, I.: Dependency Syntax: Theory and Practice, State University of New York Press, 1987.

[Marcus, 93] Marcus, M. P., Santorini, B., Marcinkiewicz, M. A.: Building a Large Annotated Corpus of English: The Penn Treebank, Computational Linguistics, 1993.

[Pang, 02] Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment Classification Using Machine Learning Techniques, In Proc. Conf. Empirical Methods in Natural Language Processing, 2002, 79-86.

[Popescu, 05] Popescu, A. M., Etzioni, O.: Extracting Product Features and Opinions from Reviews, In Proc. Conf. Human Language Technology and Empirical Methods in Natural Language Processing, Vancouver, British Columbia, 2005, 339–346.

[Ratnaparkhi, 98] Ratnaparkhi, A.: Maximum Entropy Models for Natural Language Ambiguity Resolution, PhD thesis, University of Pennsylvania, 1998.

[Riloff, 03] Riloff, E., Janyce, W., Theresa, W.: Learning Subjective Nouns Using Extraction Pattern Bootstrapping, In Proc. 7th Conf. Natural Language Learning, 2003, 25-32.

[Stone, 66] Stone, P., Dunphy, D., Smith, M., Ogilvie, D., et al.: The General Inquirer: A Computer Approach to Content Analysis, Cambridge, MA: The MIT Press, 1966.

[Tan, 07] Tan, L., Taniar, D.: Adaptive Estimated Maximum-Entropy Distribution Model, Information Science, Vol. 177, No. 15 (2007), 3110-3128.

[Wilson, 05] Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing Contextual Polarity in Phrase-level Sentiment Analysis, In Proc. Conf. Human Language Technology and Empirical Methods in Natural Language Processing, Vancouver, British Columbia, 2005, 347-354.

[Xue, 09] Xue, Y., Wang, C., Ghenniwa, H. H., Shen, W.: A Tree Similarity Measuring Method and Its Application to Ontology Comparison, Journal of Universal Computer Science, Vol.15, No.9 (2009), 1766-1781.

[Yi, 05] Yi, J., Niblack, W.: Sentiment Mining in WebFountain, In Proc. 21st Int. Conf. on Data Engineering, 2005, 1073–1083.

[Zhuang, 06] Zhuang, L., Jing, F., Zhu, X. Y.: Movie Review Mining and Summarization, In Proc. 15th ACM Int. Conf. Information and knowledge management, 2006, 43-50.