FEROM: Feature Extraction and Refinement for Opinion Mining

Hana Jeong, Dongwook Shin, and Joongmin Choi

Opinion mining involves the analysis of customer opinions using product reviews and provides meaningful information including the polarity of the opinions. In opinion mining, feature extraction is important since the customers do not normally express their product opinions holistically but separately according to its individual features. However, previous research on feature-based opinion mining has not had good results due to drawbacks, such as selecting a feature considering only syntactical grammar information or treating features with similar meanings as different. To solve these problems, this paper proposes an enhanced feature extraction and refinement method called FEROM that effectively extracts correct features from review data by exploiting both grammatical properties and semantic characteristics of feature words and refines the features by recognizing and merging similar ones. A series of experiments performed on actual online review data demonstrated that FEROM is highly effective at extracting and refining features for analyzing customer review data and eventually contributes to accurate and functional opinion mining.

Keywords: Customer review analysis, feature-based opinion mining, feature extraction, feature refinement.

Manuscript received Oct. 23, 2010; revised Mar. 24, 2011; accepted Apr. 7, 2011.

This work was supported by the Industrial Strategic Technology Development Program (10035348, Development of a Cognitive Planning and Learning Model for Mobile Platforms) funded by the Ministry of Knowledge Economy (MKE), Rep. of Korea.

Hana Jeong (email: hnjeong@islab.hanyang.ac.kr) was with the Department of IT Research, IBK Systems Corp., and is now with Daum Communications Corp., Rep. of Korea.

Dongwook Shin (email: foremostdw@gmail.com) and Joongmin Choi (corresponding author, email: jmchoi@hanyang.ac.kr) are with the Department of Computer Science and Engineering, Hanyang University, Ansan, Rep. of Korea.

http://dx.doi.org/10.4218/etrij.11.0110.0627

I. Introduction

As E-commerce has proliferated, with more people buying and selling more products online, customer reviews that describe experiences with product and service use are becoming more important [1]. Potential customers want to know the opinions of existing customers to garner information about the products they plan to buy, and businesses want to find and analyze public or customer opinions of their products to establish future directions for improvement [2].

Customer reviews generally contain the product opinions of many customers expressed in various forms including natural language sentences. A common phenomenon in natural sentence-based customer reviews is that people generally do not express their opinions in a simple way such as "this camera is good," but present them using features of the product such as "the battery life of this camera is too short." Our overall goal is to search for opinions about features of a target product from a collection of customer review data, analyze the opinion sentences, determine the orientations of the opinions, and provide a summary to the user.

This process is generally known as opinion mining, a branch of data mining that analyzes individual subjective opinions such as product reviews and extracts meaningful information from these reviews including the orientations of the opinions using natural language processing (NLP) methods or probabilistic inference models [3]. Specifically, we focus on feature-based opinion mining, in which the task applies to the sentence level to discover details about which aspects of a product people felt good or bad [4], [5]. Here, the term *feature* denotes an attribute or a component of a product, such as the "size" or "battery life" of a camera [6], [7].

In feature-based opinion mining, two tasks must be

accomplished. First, features of the product about which the reviewers have expressed their opinions must be identified and extracted. Second, the orientations or the polarities of the opinions must be determined [8]. In the final stage, opinion mining summarizes the extracted features and opinions.

Previous studies on feature-based opinion mining have applied various methods for feature extraction and refinement, including NLP and statistical methods. However, these analyses revealed two main problems. First, most systems select the feature from a sentence by considering only information about the term itself, for example, *term frequency*, not bothering to consider the relationship between the term and the related opinion phrases in the sentence. As a result, there is a high probability that the wrong terms will be chosen as features. Second, words like 'photo,' 'picture,' and 'image' that have the same or similar meanings are treated as different features since most methods only employ surface or grammatical analysis for feature differentiation. This results in the extraction of too many features from the review data, often causing incorrect opinion analysis and providing an inappropriate summary of the review analysis.

To resolve these problems, this paper proposes an enhanced method called, feature extraction and refinement for opinion mining (FEROM). The overall process of FEROM consists of three phases: preprocessing, feature extraction, and feature refinement. In preprocessing, FEROM conducts a morphological analysis including part-of-speech (POS) tagging of the review data and sentence splitting of a compound sentence into multiple sentences. In feature extraction, FEROM selects candidate features from noun phrases of the sentences and extracts related opinion information. In feature refinement, FEROM reduces the number of candidate features by merging candidates that have semantic similarity. During this process, the opinion information expressed by some opinion phrases is exploited to measure the similarities among candidate features.

To evaluate the functionality of FEROM, a series of experiments was performed on real online review data, the results of which showed that FEROM is highly effective at extracting and refining features for analyzing customer review data, greatly contributing to accurate and functional opinion mining.

II. Related Work

Studies on feature-based opinion mining have exploited various methods for feature extraction and refinement, including NLP and rule-based methods [9], [10], statistical methods [11], [12], and ontology-based methods [13].

Liu [10] proposed a system to extract features from review data using association rule mining. The system selects frequent terms and then extracts features by measuring the similarities between selected terms. The main problem of this method is that the system only considers the information from the term itself, for example, *term frequency*, which does not reflect the relationship between a feature and its related opinion information. As an example, we picked three sentences as shown in Table 1, in which 'digital camera' was extracted as a feature in Liu's system [5]. We examined those three sentences to determine the suitability of the feature being selected from each sentence.

Table 1. Suitability of 'digital camera' being selected as a feature in Liu's system.

Sentences	Suitability	Comments
I had searched for a **digital camera** for 3 months.	No	Does not contain opinion information for 'digital camera'
This is the best **digital camera** on the market.	Yes	Contains opinion information for 'digital camera' with the opinion word 'best'
The **camera** does not have a **digital** zoom.	No	Contains opinion information for 'zoom,' not for 'digital camera'

The first and the third sentences do not contain opinion information about the feature 'digital camera.' In particular, it would be appropriate to extract 'zoom' as a feature in the third sentence instead of 'digital camera.' Only the second sentence contains suitable opinion information (with opinion word 'best') for selecting 'digital camera' as a feature. Although 'digital camera' does not represent the characteristics of the product in the other two sentences, it is still a candidate feature because it is a noun phrase in the sentence. However, without opinion information, it is difficult to determine using Liu's system whether a word should be selected as a feature. Our proposed method solves this problem by referencing the relationship between a feature and its related opinion information during feature extraction.

Ding [10] proposed a feature extraction method using a rule-based approach. This method extracts a relatively large number of features compared with the amount of review data. For example, it generates 263 features from 45 reviews for digital cameras. The main reason for the extraction of so many features is that terms that have the same or similar meanings are not considered as the same features. For example, 'photo,' 'picture,' and 'image' all have the same meaning; however, they are considered as different features simply because they are different words. Consequently, this system could not

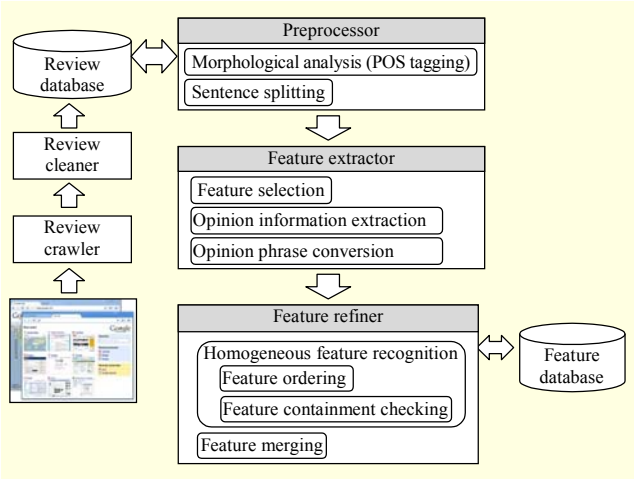

Fig. 1. FEROM system architecture.

provide proper summary information for the product. In FEROM, we solve this problem by reducing the number of features by merging words that have similar meanings using the semantic similarity between features and then providing reliable summary information for the product based on the merged features.

Aciar [13] proposed a feature extraction method for opinion mining that uses an ontology. Although this method worked well semantically, the main problem is the maintenance of the ontology to address the constant expansion of the review data. In this system, the ontology is manually constructed and must be updated when new features are added. In addition, a concept that is not defined in the ontology is not able to be classified. Thus, it is necessary to construct an automatic system to avoid continued intervention.

In summary, previous studies on feature-based opinion mining do not consider the relationship between a term and its related opinion information and also do not merge words with the same or similar meanings. We propose FEROM to solve these problems.

III. System Architecture

The system architecture of FEROM is shown in Fig. 1. The review crawler collects customer review data from online stores, and the review cleaner removes unnecessary content such as HTML tags and then stores the review data to the review database.

The preprocessor conducts morphological analysis of the review data including POS tagging, splits a compound sentence into multiple sentences, and performs stopword removal and stemming.

The feature extractor extracts product features from preprocessed review data. Feature extraction proceeds in three phases: feature selection selects a candidate feature in a sentence by looking for a noun phrase, opinion information extraction finds an opinion phrase that is associated with the candidate feature, and opinion phrase conversion replaces an opinion phrase expressed using a negative term with its antonym.

The feature refiner reduces the number of features by merging candidate features with the same or similar meanings, defined as homogeneous features. The feature refiner recognizes homogenous features by exploiting the feature ordering process that synchronizes the word orders of the features to detect synonymous feature candidates and the feature containment checking process that examines the subset-superset relationship between the features to check for similarity between them. Finally, the feature merging process merges homogeneous features into a representative feature and also prunes the feature candidates that have significantly low frequencies and very small amounts of related opinion information.

IV. Preprocessing

1. Morphological Analysis

In the initial step of preprocessing, FEROM eliminates the unnecessary content, such as tags, dates, and reviewer names, from the collected review data. Then, to extract noun phrases from the review data as feature candidates, NLProcessor [14] is used to perform morphological analysis, including POS tagging.

In general, morphological analysis is an essential component of natural language processing, dealing with the componential nature of words which are composed of morphemes. Morphological analysis recognizes the words that the text is made up of and identifies their part of POSs.

For example, the result of the morphological analysis with POS tagging for the sentence "Pictures show bright and clear" is as follows:

<NG>*Pictures*_NNS</NG> <VG>*show*_VBP</VG> *bright*_JJ *and*_CC *clear*_JJ .

Here, <NG> and <VG> represent a noun group (that is, a noun phrase) and a verb group (that is, a verb phrase), respectively. Also, POS tagging is accomplished by assigning and attaching a tag label to each word with an underline. Examples of POS tagging labels used include NNS for a plural noun, VBP for a present tense verb in non-3rd person, JJ for an adjective, and CC for a coordinating conjunction. More POS tagging labels can be found in [14].

After POS tagging, stopword removal and stemming are

conducted to increase the accuracy of the search information and the overall effectiveness of the system. Stopwords generally relate to function words, including determiners (for example, 'the,' 'a,' 'an') and prepositions (for example, 'in,' 'on,' 'of"), and these stopwords are removed from the sentence since they have little meaning on their own. Stemming is a process of converting variant forms of a word into a common base form called a stem to reduce the morphological variation [15]. For example, 'automatic,' 'automate,' and 'automation' are each converted into the stem 'automat.' Stemming is useful to search for one of these words to obtain opinion information that contain another word in the same stem group.

2. Sentence Splitting

Sentence splitting is a process for segmenting a compound sentence containing conjunctions into several simple sentences. Sentence splitting is necessary because compound sentences may contain several features, each of which may represent different opinion information.

Our method of splitting a compound sentence is carried out by recognizing a complete clause which is comprised of a noun phrase and a verb phrase. When several complete clauses exist in a sentence, the connective words ('and', 'or', 'but,' and so on) and the comma (',') are used to segregate them. Hence, the first step of sentence splitting is to divide the input sentence into several candidate complete clauses by simply separating the sentence when a conjunctive word or the comma is encountered. The next step is to examine each candidate clause to see if it is complete, namely, containing both a noun phrase and a verb phrase. A candidate clause that meets this condition is recognized as a complete clause. On the other hand, a candidate clause that does not satisfy this requirement is not complete, and hence is regarded as a component of the previous complete clause.

As an example, consider the following sentence from the collected review data:

"The screen is big enough, colors are vivid and the pictures show bright and clear."

Three features, 'screen,' 'colors,' and 'pictures,' exist in this sentence, and each feature involves separate opinion information expressed in a clause in the sentence. Sentence splitting is necessary in this case to recognize the three different opinions.

To assure correct splitting, FEROM identifies complete clauses, comprised of a noun phrase and a verb phrase appearing between two conjunctions or the commas. In the case of the above example sentence, "colors are vivid" is recognized as a complete clause since it contains a noun phrase

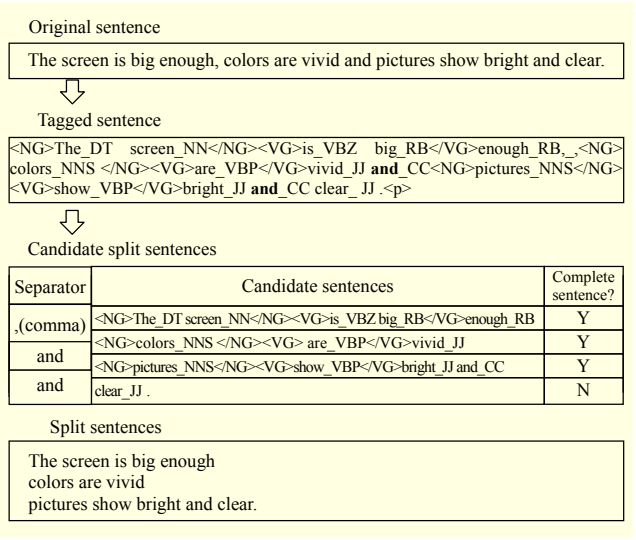

Fig. 2. Process of applying sentence-splitting algorithm to example sentence.

'colors' and a verb phrase 'are,' which are located between a comma and a conjunction 'and.' Each complete clause is then separated from the original sentence to form a new sentence.

The process of applying the sentence-splitting algorithm to the example sentence is shown in Fig. 2. The original sentence contains three separators including one comma and two conjunctions ('and'). Therefore, this compound sentence is initially split into four candidate sentences. However, the last candidate "clear," which is isolated since it is located between the conjunction 'and' and a period, cannot be used alone because it does not have a noun phrase and a verb phrase. As a result, it is attached to the previous candidate sentence "the pictures show bright" to form a new sentence "the pictures show bright and clear." Consequently, the original sentence is split into three sentences.

V. Feature Extraction

After sentence splitting, we can assume that each sentence contains opinion information about a single feature. In general, a feature in a sentence is in the form of a noun phrase [16], so feature selection normally proceeds by selecting noun phrases. For example, in the sentence "colors are vivid," the noun 'colors' is easily identified as an important feature. However, this simple process does not always work correctly. As we have already seen in Liu's system explained in section II, the noun phrase 'digital camera' can be extracted as a feature, although the sentence "I had searched for a digital camera for three months" does not contain any opinion information about the qualities of a digital camera.

Opinion information is expressed through an opinion phrase

that normally is in the form of an adjective such as 'vivid' in the example sentence mentioned above. On the other hand, the sentence "I had searched for a digital camera for three months" does not contain an adjective phrase for 'digital camera,' so it must be discarded. Based on these observations, feature extraction in FEROM proceeds in two phases: feature selection and opinion information extraction. The feature selection process simply selects a noun phrase in the sentence and assigns it as a candidate feature. The opinion information extraction process then identifies an opinion phrase in the form of an adjective. If such a phrase is found, FEROM regards the candidate as a proper feature and stores it along with the opinion phrase. Otherwise, the candidate is discarded since it is not associated with any opinion information.

Often, the opinion information in a sentence is expressed with negative terms such as 'not,' 'no,' and 'hardly.' In this case, the orientation of the opinion about the feature is the opposite of the meaning of the corresponding opinion phrase. Hence, for correct analysis, FEROM employs the opinion phrase conversion process that replaces an opinion phrase expressed using a negative adjective phrase with its antonym using WordNet [17]. For example, the sentence "the picture quality is not good" is changed into "the picture quality is bad."

The algorithm for opinion phrase conversion first determines whether a negative term exists in a sentence, and then calculates the distance between the opinion phrase and the negative term by counting the number of words between them. If the distance is smaller than the threshold α, the opinion phrase is replaced by its antonym. In this study, α is set to 3 by analyzing 50 sentences that include the negative terms.

VI. Feature Refinement

Feature refinement aims at reducing the number of features obtained as a result of the feature extraction process in order to simplify product review searches and eventually to provide a correct opinion summary of the customer reviews. To achieve these goals, the feature refinement in FEROM proceeds in two stages. In the first stage, the homogenous feature recognition process analyzes the similarities between the features and groups those features with the same or similar properties. In the second stage, the feature merging process determines the representative feature out of the candidate features in the same homogeneous group and prunes other redundant features.

1. Homogeneous Feature Recognition

In FEROM, homogeneous features are defined as features with the same or similar meanings. FEROM determines whether the extracted features are homogeneous using the synonym relation process in WordNet. In other words, if there is a synonym relation between two features, they are regarded as similar and homogeneous, as shown in Fig. 3.

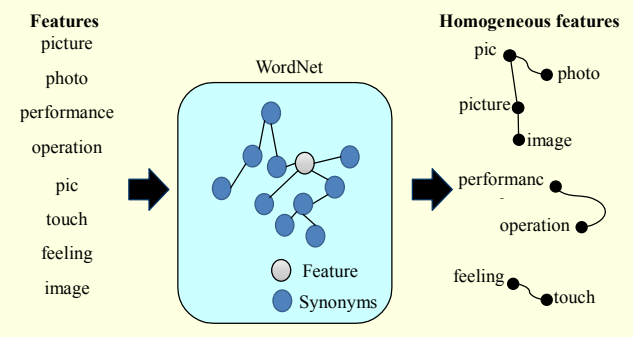

Fig. 3. Detecting homogeneous features using synonym relation in WordNet.

Table 2. Examples of compound nouns with respect to word configuration and word order.

Word configuration \ Word order	Same word order	Different word order
Same words	-	'picture quality' vs. 'quality picture'
Synonymous words	'photo quality' vs. 'picture quality'	'picture quality' vs. 'quality photo'

However, in the case of a compound noun, the process of recognizing homogeneous features is less obvious. In linguistics, two compound nouns that are comprised of the same nouns but with different word order might possess different meanings. Hence, to determine the homogeneity of 'compound noun'-based features, we consider three ways to determine whether or not the features consisting of *n*-words represent the same meaning, as shown in Table 2.

In fact, FEROM describes all the above features, that is, 'picture quality,' 'quality picture,' 'photo quality,' and 'quality photo,' as homogeneous. This implies that, for compound nouns, FEROM emphasizes the synonymity of individual words and ignores the word orders. Based on this principle, FEROM employs the processes of feature ordering and feature containment checking to recognize homogeneous features. Feature ordering synchronizes two compound features by changing the word order considering synonymous words. For example, features like 'the size of the camera' and 'the camera volume' are converted to 'camera size' and 'camera volume' by changing the word order of the first feature and by recognizing the synonymy of 'size' and 'volume.'

Feature containment checking detects homogeneous features by investigating the subset-superset relationship between

Table 3. Example subset-superset relations between features.

Review sentences	Feature	Feature after feature ordering	Feature in a set notation	Opinion phrase	Meaning of the feature
The camera is big.	camera	camera	{camera}	big	the size of camera
The camera size is big.	camera size	camera size	{camera, size}	big	the size of camera
The volume of the camera is huge.	volume of camera	camera volume	{camera, volume}	huge	the size of camera

features. This method maps each feature into a feature set consisting of simple nouns and determines if there is a subset-superset relationship between any two features and if those features are associated with synonymous opinion phrases. If these two conditions are satisfied, the two features are determined to be homogeneous.

Table 3 shows some example sentences for determining a subset-superset relationship between features.

Each of all three sentences contains a review about 'the size of camera.' Since the synonyms 'big' and 'huge' are used as opinion phrases, those sentences are believed to refer to the same feature. However, they use different noun phrases, that is, 'camera,' 'camera size,' and 'the volume of camera' to represent the feature. In a set notation, the features are {camera}, {camera, size}, and {camera, volume}, respectively. Since 'size' and 'volume' are treated as synonyms in WordNet, there is a subset-superset relationship for any combination of those two sets. Also, since all of the sentences use synonymous opinion phrases, it was concluded that 'camera,' 'camera size,' and 'the volume of the camera' are homogeneous features.

The task of recognizing homogeneous features contributes to a correct summary of review opinion information. For example, a user who is curious about other customers' opinions regarding the size of a particular camera would search for review information using the query 'camera size.' Without the feature refinement, a large portion of the review sentences would not be extracted or included in the analysis, for example, the first and the third sentences in Table 3. Consequently, the opinions expressed in those non-extracted sentences would not be reflected in the decision about opinion orientation, which might result in an inaccurate summary of the review opinions.

2. Feature Merging

Once several features are determined to be homogeneous, the next step is to merge them into a single feature. Figure 4 shows the feature merging process in FEROM. In this figure,

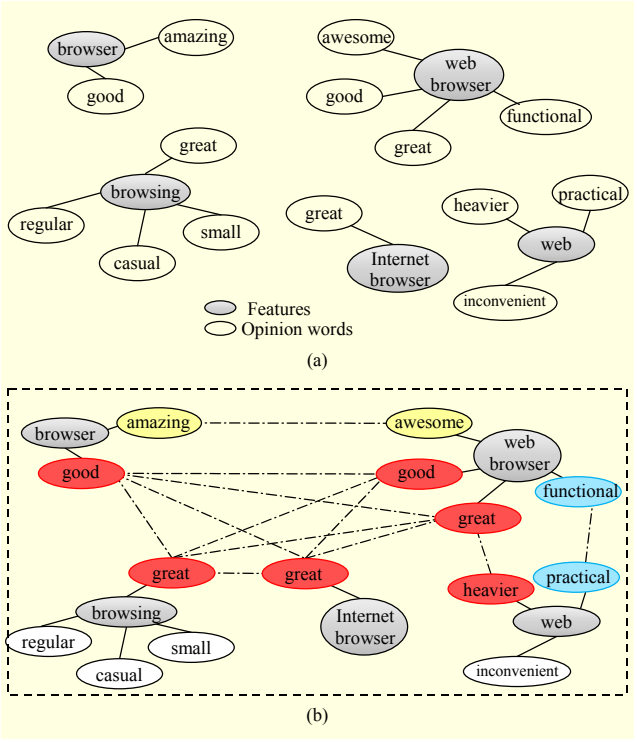

Fig. 4. Example of feature merging: (a) features and their opinion words prior to feature merging and (b) recognizing synonymous opinion words for feature merging.

two examples are used; one group of homogeneous features includes 'browser,' 'browsing,' 'internet browser,' and 'web browser' that are merged into the representative feature 'brow' (a stem word), and the other group includes 'web browser' and 'web' that are merged into 'web.'

In FEROM, there are two conditions required for features to be merged. First, two features must have a subset-superset relationship, and second, the opinion phrases of the features must be synonymous. Once the first condition is satisfied according to homogeneous feature recognition, the second condition is checked using the WordNet synonym function and the Jaccard coefficient method to measure the similarity between features:

$$sim(f_i, f_j) = \frac{n(SO)}{n(O_i) \cup n(O_j)}. \quad (1)$$

Here, f_i and f_j denote the features with a subset-superset relationship, O_i and O_j denote opinion phrases for f_i and f_j, respectively, and SO denotes synonymous opinion phrases contained in f_i and f_j (Note that n denotes 'the number of' as usual).

If this rate is greater than the threshold β, the corresponding features are merged. The threshold β is determined via experiment, as described in subsection VII.1. Next, the feature frequency considering the number of syllables (FFCNS) value

of each feature in the merged feature group is considered in order to select the representative feature using

$$FFCNS = n\_syllables \times feature\_freq, \quad (2)$$

where *n_syllables* denotes the number of syllables (*n-gram*) in the feature word, and *feature_freq* denotes the frequency of the feature. In other words, FEROM calculates the feature frequency by considering the number of syllables in the feature word along with the frequency of the feature and then selects the feature with the largest value as the main representative feature.

Although we could reduce the number of features through feature refinement and merging, irrelevant features may still exist. We assume that the features in the review data which have a low frequency and few opinion phrases are irrelevant to the product representation; therefore, in FEROM, features that have a lower than average frequency of all features and contain fewer than the average opinion phrases are further removed.

VII. Experimental Results

1. Determining the Threshold for Feature Merging

We used the 'precision' measure described in (3) to determine the threshold β for feature merging. This precision measure evaluates the ratio of correctly extracted features by the system. In (3), 'correct feature' indicates the feature that coincides with the manually tagged feature under human supervision for the experiments. Although both 'precision' and 'recall' are important evaluating criteria, we thought that 'precision' is more appropriate measure based on the observation that, in summarization, it is more important to provide the correct and exact feature information to the user than to provide complete information without missing features.

$$\text{Precision} = \frac{\text{No. of correct features extracted by the system}}{\text{No. of features extracted by the system}},$$
$$\text{Recall} = \frac{\text{No. of correct features extracted by system}}{\text{No. of correct features}}. \quad (3)$$

We analyzed 300 customer review sentences for 'camera,' 'cell phone,' and 'speaker' products to determine the threshold for feature merging.

An experiment is performed to obtain the precision measures of feature extraction by varying the threshold values. As shown in Fig. 5, the precision is maximized when the threshold is 0.1. Therefore we set the threshold for feature merging at 0.1, which implies that features with similarity values greater than 0.1 are merged.

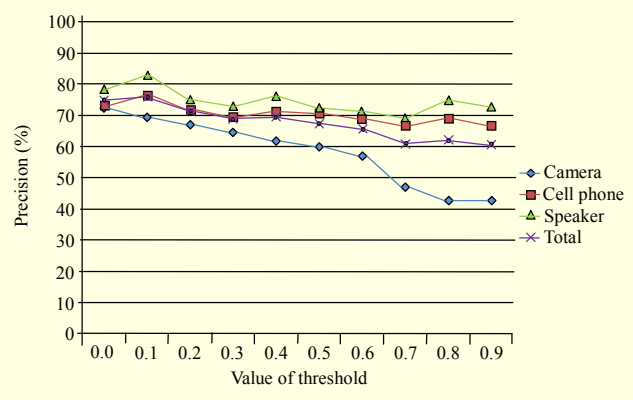

Fig. 5. Precision measures of feature extraction to determine the threshold for feature merging.

Table 4. Performance values measured after each feature extraction and refinement process (P: precison, R: recall).

Process / Product	Feature extraction		Feature refinement					
			Feature ordering		Feature containment checking		Feature merging	
	P	R	P	R	P	R	P	R
Camera	.91	.96	.87	.92	.85	.87	.85	.87
Cell phone	.83	.93	.85	.93	.84	.87	.85	.87
MP3	.83	.97	.83	.95	.83	.86	.84	.85
Navigation	.78	.97	.81	.97	.80	.94	.86	.93
Speaker	.84	.92	.84	.91	.84	.90	.84	.89
Average	.83	.95	.84	.94	.83	.89	.85	.88

2. Performance Evaluation of FEROM

We measured the effectiveness of FEROM using real online review data. A total of 500 review data points were collected from Amazon (amazon.com) and CNet (reviews.cnet.com), consisting of 50 reviews for each of five products (camera, mobile phone, MP3, navigation, and speaker) from each site. A total of 5,655 sentences were selected from the 500 reviews, and we manually read all the selected sentences and tagged the words that are considered as appropriate for the features in each sentence.

We evaluated the performance of FEROM by applying the precision and recall measures as described in (3). Note that feature extraction is performed on a sentence-by-sentence basis. For example, assume we evaluate the performance of feature extraction using the following sentence:

Very attractive compact camera, but has a very terrible battery life…only shoots 110 shots or less.

In this sentence, 'camera' and 'battery life' are correct features. If FEROM extracts three features, 'camera,' 'battery life' and 'shot,' the precision is 2/3=0.67 (67%) and the recall is 2/2=1.0 (100%). The performance evaluation proceeds such that the precision and recall values are measured separately after each process, including feature extraction, feature ordering, feature containment checking, and feature merging. This enables one to detect the variation in performance as the processes are performed in succession.

The experimental result for feature extraction and refinement is summarized in Table 4. Note here that, in the course of calculating precision and recall measures, the number of extracted features means the total number of features extracted from the sentences. Therefore in this case, if the same feature is found three times in three different sentences, the number of features extracted is increased by 3 counts. All categories of products showed similar performances overall, but 'camera' showed a relatively higher performance compared to those of the other products, with a precision of 91% and a recall of 96%. The reason for this result is believed to be that cameras are popular products in online shopping malls, and the customer reviews for this product are mostly well-written. On the other hand, 'navigation' has a relatively low precision value in the feature extraction stage, since proper nouns such as region names were extracted as features, even though they tend to occur infrequently and are hard to associate with opinion phrases. However, for the same reason, the precision is increased substantially in the next processes.

Note in Table 4 that, while passing through each stage, although the overall recall measures are decreasing, the overall precision measures are increasing slightly.

To evaluate further the performance of feature merging, we measured the reduced number of features and the error rates in the course of merging. Note that the concept of 'the number of features' is slightly different from the previous results. Here, the number of extracted features means the total number of different features extracted from the sentences. So in this case, even when the same feature is found three times in three different sentences, the number of extracted features is increased by only 1 count, not in 3 counts.

Table 5 shows how the decreasing numbers of features change after each process. A total of 2,182 features were extracted in the feature extraction stage. Then, the reduction in the number of features progressed after each process of the feature refinement stage to a final reduction of 74%.

To measure the error rates in merging, we defined over-merging as the ratio of merged features that should not have been merged, and under-merging as the ratio of unmerged features that should have been merged. Table 6 shows the resulting error rates in feature merging after each feature refinement process. As shown in this table, FEROM's feature merging performance is reasonably satisfactory by limiting the error rates to about 10% in average.

Table 5. Decreasing number of features.

Process / Product	Feature extraction	Feature refinement		
		Feature ordering	Feature containment checking	Feature merging
Camera	451	407	267	125
Cell phone	490	461	332	141
MP3	352	316	243	91
Navigation	474	452	351	117
Speaker	415	391	267	97
Total no.	2,182	2,027	1,460	572
Decreasing feature rate		7% ↓	33% ↓	74% ↓

Table 6. Error rates in feature merging measured after each feature refinement process (Ov: over-merging, Un: under-merging).

Process / Product	Feature refinement					
	Feature ordering		Feature containment checking		Feature merging	
	Ov	Un	Ov	Un	Ov	Un
Camera	.08	.05	.14	.05	.12	.03
Cell phone	.05	.19	.09	.06	.04	.05
MP3	.11	.09	.16	.08	.13	.04
Navigation	.04	.29	.05	.15	.04	.09
Speaker	.03	.21	.11	.09	.11	.05
Average	.06	.16	.11	.09	.09	.05

In summary, the FEROM system showed satisfactory results with an overall precision of 85% and an overall recall of 88%. In addition, the FEROM system drastically reduced the number of relevant features during feature refinement, qualitatively affecting the system performance by providing a more correct summary of the product opinions.

3. Performance Evaluation Using a Virtual Opinion Mining Framework

Since the FEROM system showed satisfactory performance results by maintaining relatively high precision and recall measures during correct feature extraction and by reducing the number of features through feature refinement and merging, we attempted to apply the FEROM system to a virtual opinion mining framework in order to validate its effectiveness.

Table 7. Ten fixed features for each product category.

Product category	Features
Camera	battery life, shutter speed, light, startup, LCD screen, macro mode, steel construction, feel/touch, shoot/shot, control
Cell phone	camera, email, media player, qwerty keyboard, ringtone, screen, speaker phone, text message, voice quality, web browser
MP3	bass, battery life, color, control, headphone, itunes, price, screen, size, weight
Navigation	battery life, bluetooth, GPS, manual, map, MP3 player, road, satellite, user interface, windshield mount
Speaker	balance, bass, clarity, computer speaker, frequency response, Klipsch, speaker cable, speaker option, volume level, woofer

We randomly collected customer reviews for five product categories (camera, cell phone, MP3, navigation, and speaker) as before, and a total of 1,601 sentences (278 for 'camera,' 333 for 'cell phone,' 244 for 'MP3,' 305 for 'navigation,' and 441 for 'speaker') were selected. At this time, we fixed ten features for each product and measured how well the system behaved to correctly extract those features from the review sentences. Table 7 lists the ten features for each product category.

Performances are compared among three systems: FEROM, Liu's system [9], and the traditional 'keyword-based querying' (KBQ) method. As mentioned in section II, Liu's system only considers the information from the term itself, such as term frequency, without reflecting the relationship between a feature and its related opinion information. The KBQ method is a general method that searches for the results (in various forms such as documents) that include the keywords in the query in their exact forms [18]. Thus, in KBQ, two features are considered different unless they are exactly matched, even if they are synonymous.

The comparison of performance results based on precision and recall is shown in Table 8 and Fig. 6. In addition, we also measured the number of feature instances recognized during this experiment to evaluate the effectiveness of feature merging. These comparison results clearly demonstrate that FEROM's feature extraction and refinement strategy outperforms the other two methods in all three measures. Note that the precision of the KBQ method is low because features with no associated opinion phrases were extracted, and the recall of KBQ matching is also low because some homogeneous features with syntactically different forms were not extracted.

Table 8. Performance comparison among KBQ, Liu's system, and FEROM.

Method / Product	KBQ		
	Precision	Recall	No. of features recognized
Camera	.656	.456	128
Cell phone	.691	.568	183
MP3	.644	.901	218
Navigation	.483	.694	221
Speaker	.768	.675	175
Average	.648	.659	185
Method / Product	**Liu's system**		
	Precision	Recall	No. of features recognized
Camera	.746	.744	144
Cell phone	.700	.775	198
MP3	.702	.831	184
Navigation	.539	.666	179
Speaker	.815	.669	156
Average	.700	.737	172.2
Method / Product	**FEROM**		
	Precision	Recall	No. of features recognized
Camera	.900	.818	126
Cell phone	.904	.764	149
MP3	.918	.853	136
Navigation	.898	.789	129
Speaker	.926	.743	116
Average	.909	.793	131.2

VIII. Conclusion

We proposed FEROM, an enhanced method of feature extraction and refinement for opinion mining, to analyze product review data. FEROM extracts candidate features considering the syntactic and semantic similarities between them and reduces the number of features by merging words with similar meanings.

FEROM showed satisfactory performance results through a series of experiments conducted on real product review data. Furthermore, the FEROM system showed good performance in a virtual opinion mining framework. Based on these observations, we claim that FEROM is a proper method for opinion mining by employing an enhanced scheme of feature extraction and refinement to analyze customer review data.

One of the weak points in FEROM is that, in the

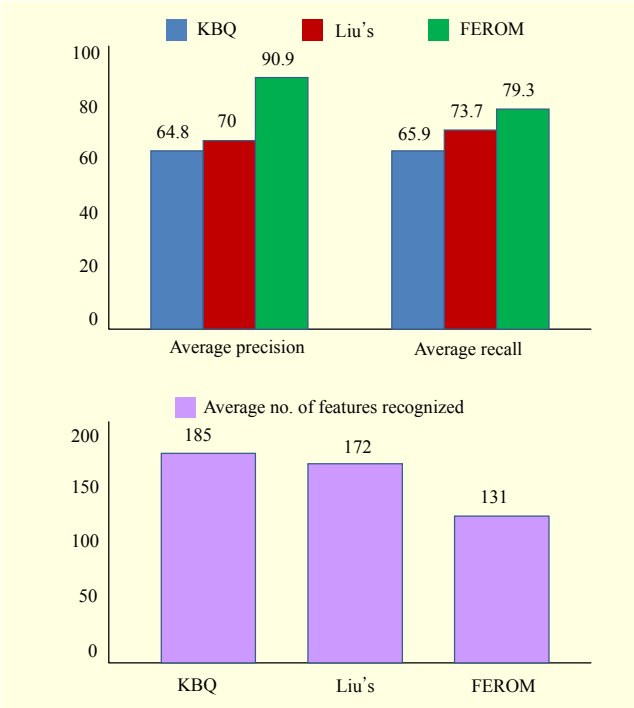

Fig. 6. Performance comparison among KBQ, Liu's method, and FEROM depicted in a graphical chart form.

homogeneous feature recognition process, only synonymous opinion words are considered to determine similar features. In a strict sense, however, antonyms can also express opinion information for homogeneous features. We intentionally excluded the case of antonym opinion phrases since our preliminary experience showed that it generated a huge number of features, which deteriorates the performance of the system. Nonetheless, this situation should be resolved, and we plan to revise and enhance the homogeneous feature recognition process to examine the case of antonym opinion phrases.

References

[1] J. Willis, "What Impact Will E-Commerce Have on the U.S. Economy?" *Economic Review*, Federal Reserve Bank of Kansas City, vol. 89, no. 2, 2004, pp. 53-71.

[2] N. Li and Z. Ping, "Consumer Online Shopping Attitudes and Behavior: An Assessment of Research," *Proc. 8th Americas Conf. Inf. Syst.*, 2002, pp. 508-517.

[3] B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis," *Foundations and Trends in Info. Retrieval*, vol. 2, no. 1-2, 2008, pp. 1-135.

[4] A. Popescu and O. Etzioni, "Extracting Product Features and Opinions from Reviews," *Proc. Conf. Human Language Technol. Empirical Methods Natural Language Process.*, 2005, pp. 339-346.

[5] B. Liu, M. Hu, and J. Cheng, "Opinion Observer: Analyzing and Comparing Opinions on the Web," *Proc. 14th Int. Conf. World Wide Web*, 2005, pp. 342-351.

[6] Y. Kim et al., "Feature Selection in Data Mining," *Data Mining: Opportunities and Challenges*, Idea Group Publishing, 2003, pp. 80-105.

[7] A. Kotcz, P. Vidya, and K. Jugal, "Summarization as Feature Selection for Text Categorization," *Proc. 10th Intl. Conf. Inf. Knowl. Manag.*, 2001, pp. 365-370.

[8] B. Liu, "Opinion Mining," *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*, Springer, 2007, pp. 411-448.

[9] B. Liu and M. Hu, "Mining Opinion Features in Customer Reviews," *Proc. 19th Nat. Conf. Artificial Int.*, 2004, pp. 755-760.

[10] X. Ding, B. Liu, and Y. Philip, "A Holistic Lexicon-Based Approach to Opinion Mining," *Proc. Int. Conf. Web Search Web Data Mining*, 2008, pp. 231-240.

[11] A. Abbasi, H. Chen, and A. Salem, "Sentiment Analysis in Multiple Languages: Feature Selection for Opinion Classification," *ACM Trans. Inf. Syst.*, vol. 26, no. 3, 2008, pp 1-34.

[12] S. Das and M. Chen, "Yahoo! for Amazon: Sentiment Extraction from Small Talk on the Web," *Manag. Sci.*, vol. 53, no. 9, 2001, pp. 1375-1388.

[13] S. Aciar et al., "Informed Recommender: Basing Recommendations on Consumer Product Reviews," *IEEE Intell. Syst.*, vol. 22, no. 3, 2007, pp. 39-47.

[14] NLProcessor–Text Analysis Toolkit, 2000. http://www.infogistics.com/textanalysis

[15] Porter's Stemming Algorithm. http://tartarus.org/~martin/PorterStemmer/

[16] O. Schiller and A. Caramazza, "Grammatical Feature Selection in Noun Phrase Production: Evidence from German and Dutch," *J. Memory and Language*, vol. 48, no. 1, 2003, pp. 169-194.

[17] G. Miller et al., "Introduction to WordNet: An On-line Lexical Database," *Int. J. Lexicography*, vol. 3, no. 4, 1990, pp. 235-244.

[18] R. Beaza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, Addison-Wesley, 1999.

Hana Jeong received her BS in internet media engineering from Kyungwon University, Rep. of Korea, in 2008, and MS in computer science and engineering from Hanyang University, Rep. of Korea, in 2010, respectively. She is currently a research staff member at Daum Corp., Rep. of Korea. Her research interests include data mining, opinion mining, information retrieval, and ubiquitous computing.

Dongwook Shin received his BS in internet media engineering from Kyungwon University, Rep. of Korea, in 2007, and his MS in computer science and engineering from Hanyang University, Rep. of Korea, in 2009, respectively. He is currently a PhD candidate in the Department of Computer Science and Engineering at Hanyang University, Ansan, Rep. of Korea. His research interests include social networks, data mining, information retrieval, and artificial intelligence.

Joongmin Choi received his BS and MS in computer engineering from Seoul National University, Seoul, Rep. of Korea, in 1984 and 1986, respectively, and the PhD in computer science from the State University of New York at Buffalo, USA, in 1993. He is currently a professor in the Department of Computer Science and Engineering at Hanyang University, Ansan, Rep. of Korea. His research interests include web intelligence, information retrieval and extraction, data mining, semantic web, and artificial intelligence.