A Holistic Approach to Product Review Summarization

Jung-Yeon Yang, Jaeseok Myung, Sang-goo Lee
School of Computer Science and Engineering
Seoul National University, Seoul, Korea
{jyyang, jsmyung, sglee}@europa.snu.ac.kr

Abstract

As online shopping is becoming commonplace, more and more product information and product reviews are posted on the Internet. Because customers cannot see and feel the products directly, product reviews are becoming an essential source of qualitative information. As a result, the volume of reviews is increasing drastically and review summarization (or opinion mining) is becoming an important tool for a practical utilization.

Presented in this paper is a holistic approach to product review summarization which improves on existing methods. In extracting product features and opinion words, we present a flexible extraction tool that can be customized according to the characteristics of source documents. For sentiment analysis and feature scoring, we utilize the rate scores, which are overall ratings of the product by users. Feature dependent sentiment polarities of opinion words are also utilized. Experiments show that these considerations improve the accuracy of product review summarization.

1. Introduction

As the product information is increasing in e-market place, many product reviews are also posted in the Internet. Since customers cannot see the products directly, product reviews are becoming an essential source of qualitative information. With growing needs of product reviews, the problem of the massive volume of reviews arises as a practical issue. In order to solve this problem, review summarization (or opinion mining) has become an important tool for the practical utilization of the reviews.

In order to find out the semantics of documents, Natural Language Processing (NLP) was applied[1][2]. However, as the massive data processing is needed, NLP approach met some problems; it takes a very long time to handle large volume of data. In order to solve these problems, computational statistics that can handle massive data in a more practical time were applied[7].

In this paper, we suggest a holistic approach to product review summarization. We summarize reviews as scoring of product features using user’s opinion, feature occurrences, and the rate of review. Especially, the rate, which represents user’s explicit evaluations to the product, is important information. So, we utilize rate scores to the review summarization.

There are 3 steps in a product review summarization: feature extraction, sentiment analysis, and feature scoring. When extracting product features and opinion words, we present a flexible extraction tool, which can be customized according to the characteristics of source documents. For sentiment analysis and feature scoring, we utilize rate scores, which are users’ overall ratings of the product. Feature dependant sentiment polarities of opinion words are also utilized. Lastly, we show that these considerations can improve the accuracy of product review summarization with experiments.

2. Related Work

The purpose of the traditional document retrieval methods is to extract important keywords. However, in case of feature extraction on the summarization of product reviews, we select and extract product features that customers are interested in. In order to extract product features and opinion words from product reviews, we use the part-of-speech (POS) tagging. The POS tag is a famous and useful approach to analyze documents. It is helpful to find out the semantics and role of words in sentences. Hu et al[1][4] and Popescu et al[3] use algorithms which apply the POS tagging to identify nouns. An association-mining, frequency-based approach, a Point-wise Mutual Information (PMI), Bayesian network, SVM(Support Vector Machine), and n-gram approach are also used to extract feature words and opinion words.

In order to understand the semantics of review sentences in the product review, many researches are studied, mainly, the NLP approaches[6][8]. In many algorithms, PMI and the association rules are used, and the WordNet is used widely as a word corpus which has sentimental information [9][10][11]. However these algorithms cannot handle massive amount of data.
in a short time; they derive a sentimental polarity of words from general usage and meaning of words, and depend on domain expert’s manual pre-definitions [9][10].

The evaluation of features can be shown as a score, which is derived from rates of product reviews. Scaffidi et al shows a score of product features using the scoring algorithms that use the rate of reviews and feature occurrences[7]. In other algorithms that use NLP methods, a set of vocabulary such as WordNet is used for scoring[1][5]. These algorithms use opinion words in order to score a product.

3. Product Review Summarization

Summarization of product reviews is partitioned into three processes. Figure 1 shows all processes of our summarization method. At first, we make feature-opinion pairs as basic information. We have been developed the extraction tool that handle feature words and opinion words from product reviews. In order to classify a sentimental polarity of opinions, we use a probability distribution of opinion words with rates. We classify an opinion word into a positive one or a negative one automatically. Lastly, we derive a score of feature -using rates of reviews, a distribution of sentimental polarities, and feature occurrences. In this section, we explain each process.

3.1. Feature Extraction

Although the reviews have tens of thousands of unique words, only a small number of words are significantly useful for the users. Thus, feature word extraction can be one of the most important issues for the review summary process, and we need to be careful in order to extract appropriate feature words. However, acquiring knowledge from a huge unstructured text is not that easy. Even identifying the important part of the review is difficult, because computer cannot understand the unstructured raw text.

There have been many approaches supporting feature extraction process, and most of the approaches based on a certain model, such as Bayesian network, SVM, and so on. Despite of many existing successful works, we did not use the model-based feature extraction method, as it considers each word just as a meaningless sequence of characters and it does not utilize the characteristics of textual data, such as a structure of a sentence, and part-of-speech information. Moreover, the model is used as fully automatic unsupervised learning methods that cannot be changed easily.

Table 1. Feature Extraction Methods in PicAChoo

<table>
<thead>
<tr>
<th>Category</th>
<th>Method</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primitive</td>
<td>Frequency-based</td>
<td>Frequently used(TF)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Widely used(DF)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Threshold</td>
</tr>
<tr>
<td></td>
<td>Co-occurrence</td>
<td>Fixed-size window</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(left, right, both)</td>
</tr>
<tr>
<td></td>
<td>Pattern-based</td>
<td>Sentence</td>
</tr>
<tr>
<td></td>
<td>Plug-in</td>
<td>Anything</td>
</tr>
<tr>
<td></td>
<td>Logical</td>
<td>And, Or</td>
</tr>
<tr>
<td></td>
<td>Arithmetical</td>
<td>+ - * / ^ %</td>
</tr>
</tbody>
</table>

For this reason, we developed a customizable feature extraction tool named PicAChoo, which stands for ‘Pick And Choose’. It concentrates on providing an environment in which you can apply several feature extraction methods without hard-coding. Applying this tool, we dealt with semi automatic supervised feature extraction model based on several heuristics. To achieve this objective, we prepared four types of primitive methods to make a feature set, and we suggested two types of composition methods to compose different feature sets. Overall methods and options are represented in Table 1.

Another aspect that we have to consider when we deal with textual data is the complex feature. Generally, a word cannot be meaningful enough without any other information. In the opinion mining case, the feature words are combined with the opinion words. PicAChoo provides a feature set that consists of a pair
of feature words and opinion words extracted by co-
ocurrence. By using the feature extraction tool, the feature set can be constructed dynamically, and the productivity of entire process can be increased.

3.2. Sentiment Analysis

After the word extraction, we classify the sentimental polarity of opinion words which describes the product feature. In typical existing sentiment classification methods, experts define seed word corpus, which have positive words and negative words, and they expand corpus using the WordNet or SentiWordNet [11][12]. However, making seed word corpus is done manually. In practical cases, we do not know what words will be used in a specific case with specific features; therefore, it is hard to define seed word corpus for every case of word usage.

In order to find out a sentimental polarity of opinion words, we have to consider the context of words. We made an algorithms to classify the sentimental polarity of a word in consideration of an own context. Our algorithm uses sentiment dictionaries in order to decide a sentimental polarity. These dictionaries are constructed automatically using the rate of reviews and review titles. Using these positive and negative sentiment dictionaries, we can get a sentimental polarity of the feature-opinion pair.

We use PMI values between an opinion word and a feature. Equation (1) shows how to get a sentimental orientation (or polarity) of the opinion-feature pair.

\[
SO_{\text{-PMI}}(o, f) = \text{posPMI}(o, f) - \text{negPMI}(o, f)
= \log \frac{\text{posP}(o|f)}{\text{posP}(o)\cdot \text{posP}(f)} - \log \frac{\text{negP}(o|f)}{\text{negP}(o)\cdot \text{negP}(f)}
\]  

(1)

In Equation (1), SO-PMI(o, f) is the sentimental orientation between o and f. o is opinion word which describes a feature f. The prefixes pos means a positive case and neg means a negative case. Each case uses a different dictionary. That is, \(pP(o)\) means the probability of the case that the opinion word is mentioned in a positive review. In same, \(pP(f)\) is the case of feature f. \(pP(o \cap f)\) is the probability of the case that an opinion word and a feature word are occurred together in the same review. SO-PMI(o, f) is calculated using a difference between two PMI values. Each PMI values mean a PMI between an opinion word and a feature word in a positive case or a negative case.

3.3. Scoring Product Features

In real review documents, there are several opinions and many features that are mentioned together. Each feature has its own valuation and rate of review that are estimated through the interaction among valuations of features. Rate cannot be the representative value of each product feature. In order to summarize reviews correctly, we have to find out their own valuation of each feature. In our method, we use the rate of review in order to classify a sentimental polarity of an opinion word and calculate the score of product features.

In our research, the evaluation of each product feature is expressed by showing the score of each product feature. In order to calculate the evaluation score, we use feature occurrences, rate of reviews, feature co-occurrence information and sentimental polarities of opinion words. At first, we classify the sentimental polarity of the opinion words which describes the product feature. We compute the strength of an opinion using the weight of the feature. And then, we use the distribution of the sentimental polarity and strength of the opinion.

In product reviews, users say their own opinion for each product feature and then rate a product. There are many reviews on the same product. In all product reviews, many features are mentioned, yet not every feature is mentioned in every product review. Figure 2 shows this situation.

![Figure 2](image.png)

Figure 2. Examples of an opinion composition in reviews

In the product reviews, the unit of the opinion is not a document or a review, but a product feature. We can express the distribution of opinions like Figure 2 using the sentiment classification. We assume that the rate of review is derived from the combination of several opinions about features. For example, in Figure 2, if the rate of \(R_f\) is 4, then it is derived from many positive features and negative features. \(f_4\) and \(f_5\) has a negative
influence upon the overall valuation of the product. On the other hand, \( f_1, f_2, \) and \( f_6 \) may have a higher score than the rate of \( R_1 \). That is, the positive opinion makes the rate of review high and the negative opinion makes the rate of review down. In order to reflect this phenomenon, we use Equation (2).

\[
\text{if } SP(f_i, R_j) \text{ is positive,} \\
\text{ownScore}(f_i, R_j) = \text{rate}(R_j) \times (1 + P_{neg}(R_j)) \\
\text{weight}(f_i)
\]

\[
\text{if } SP(f_i, R_j) \text{ is negative,} \\
\text{ownScore}(f_i, R_j) = \text{rate}(R_j) \times (1 - P_{pos}(R_j)) \\
\text{weight}(f_i)
\]

\( (1 \leq i \leq \text{Num of features}, 1 \leq j \leq \text{Num of reviews}) \) (2)

In Equation (2), \( SP(f_i, R_j) \) means the sentimental polarity of the \( i^{th} \) feature in the \( j^{th} \) review. \( P_{pos}(R_j) \) and \( P_{neg}(R_j) \) are the percentage of positive opinions or negative opinions in the \( j^{th} \) review. We use the percentage of the opposite sentiment in the equation for each sentimental polarity of opinion.

In the mean time, number of occurrence is one of the important factors. If a reviewer mentioned a specific product feature in the review, then it means that he thinks that the feature is more important than others. So, we have to reflect the occurrence of features as a weight when the score is calculated. The weight is calculated through the following equation: if the product feature is mentioned, the scope of a weight is from 0 to 2.

\[
\text{weight(feature)} = 2 - 2^{\text{occurrence(feature)}} \quad (3)
\]

After the computation of the score of product feature in all reviews, we estimate the representative score of the product feature using the following Equation (4).

\[
\text{score}(f_i) = \frac{\sum_{k=1}^{NR} \text{ownScore}(f_i, R_k)}{NR} \quad (4)
\]

In the above equation, \( NR \) means the number of reviews in which the \( i^{th} \) feature was mentioned and \( R \) is the review that has a comment about the \( i^{th} \) feature.

4. Experiments

In this chapter, we show the performance of each method through some experiments. We used practical review data from epinions.com for experiments and randomly selected 100 products and 6,000 reviews from the ‘Digital Camera’ category. We applied each method to those reviews in order to observe the performance of both methods.

We used the ‘Pros’ field and the ‘Cons’ field in order to make the evaluation set. Unlike general review systems, the review data from the epinions.com has two important fields; ‘Pros’ and ‘Cons’. In these fields, reviewers write good or bad points of the product explicitly in order to distinguish whether the product feature is good or not. Consequently, we make the true set with these fields.

Figure 3 shows a distribution of feature evaluation for each method. The volume of the bar means numbers of product which has each feature and an evaluation. As we can see from the results in the true set, there are several opinions for the products. But existing scoring method[7] derive biased evaluations which are presented in (b). Since they consider the rate of a review as a score of a feature, most of product features were estimated to have high scores. This phenomenon is caused from the facts that the percentages of the high-scored reviews are much bigger than the low-scored ones. Practically, in the review system, high-scored reviews are 5~10 times more than low-scored ones. So, through this method, it’s hard to find out negative features. As shown in the (c), we derived evaluations, which are closer to the true set than (b), in our proposed method case.

Figure 3. Distribution of feature evaluation
In order to improve the performance of existing method, we combine our method with the existing one. We estimate the score of product features with the mean-value of scores, which are derived from existing and our method. After computing the score using the mean-value, we compare the error summation of the existing method and our method. The performance shows the stability in comparison with other results. Figure 4 shows the percentage of improvement through our method. The volume of bars means reduction in errors in comparison with an existing method. The proposed method improves the performance about 20% in comparison with an existing method.

![Figure 4. Improvement of our proposed method](image)

5. Conclusion

In this paper, we suggest a method for a review summarization using users’ opinions and rates in practical cases. In order to extract features, we developed a customizable feature extraction tool. Many linguistic features and extraction methods are supported by this tool. For sentiment analysis and feature scoring, we utilize the rate scores which are overall ratings of the product by users. Feature dependant sentiment polarities of opinion words are also utilized. Finally, we derive a score of a product feature in order to show the quality of the feature to customers. As we utilize rates of reviews and apply sentimental polarities of opinion words to derivation of scores, we can overcome the shortcomings of prior methods and take good points of them. Through the experiment, we prove the performance of our method.

We are developing a tool that includes whole process for a summarization of product reviews. The modification of the sentiment analysis method and the scoring method is needed in order to improve a performance of our approach. We are finding out other factors which affect on the scoring of product features.

6. Acknowledgment

This research was supported by the Ministry of Knowledge Economy, Korea, under the Information Technology Research Center support program supervised by the Institute of Information Technology Advancement. (grant number IITA-2008-C1090-0801-0031)

7. References


