Mining and Summarizing Customer Reviews

Minqing Hu and Bing Liu
Department of Computer Science
University of Illinois at Chicago
851 South Morgan Street
Chicago, IL 60607-7053
{mhu1, liub}@cs.uic.edu

ABSTRACT
Merchants selling products on the Web often ask their customers to review the products that they have purchased and the associated services. As e-commerce is becoming more and more popular, the number of customer reviews that a product receives grows rapidly. For a popular product, the number of reviews can be in hundreds or even thousands. This makes it difficult for a potential customer to read them to make an informed decision on whether to purchase the product. It also makes it difficult for the manufacturer of the product to keep track and to manage customer opinions. For the manufacturer, there are additional difficulties because many merchant sites may sell the same product and the manufacturer normally produces many kinds of products. In this research, we aim to mine and to summarize all the customer reviews of a product. This summarization task is different from traditional text summarization because we only mine the features of the product on which the customers have expressed their opinions and whether the opinions are positive or negative. We do not summarize the reviews by selecting a subset or rewrite some of the original sentences from the reviews to capture the main points as in the classic text summarization. Our task is performed in three steps: (1) mining product features that have been commented on by customers; (2) identifying opinion sentences in each review and deciding whether each opinion sentence is positive or negative; (3) summarizing the results. This paper proposes several novel techniques to perform these tasks. Our experimental results using reviews of a number of products sold online demonstrate the effectiveness of the techniques.

Categories and Subject Descriptors
H.2.8 [**Database Management**]: Database Applications – *data mining*. I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *text analysis*.

General Terms
Algorithms, Experimentation, Human Factors.

Keywords
Text mining, sentiment classification, summarization, reviews.

KDD'04, August 22–25, 2004, Seattle, Washington, USA.

1. INTRODUCTION
With the rapid expansion of e-commerce, more and more products are sold on the Web, and more and more people are also buying products online. In order to enhance customer satisfaction and shopping experience, it has become a common practice for online merchants to enable their customers to review or to express opinions on the products that they have purchased. With more and more common users becoming comfortable with the Web, an increasing number of people are writing reviews. As a result, the number of reviews that a product receives grows rapidly. Some popular products can get hundreds of reviews at some large merchant sites. Furthermore, many reviews are long and have only a few sentences containing opinions on the product. This makes it hard for a potential customer to read them to make an informed decision on whether to purchase the product. If he/she only reads a few reviews, he/she may get a biased view. The large number of reviews also makes it hard for product manufacturers to keep track of customer opinions of their products. For a product manufacturer, there are additional difficulties because many merchant sites may sell its products, and the manufacturer may (almost always) produce many kinds of products.

In this research, we study the problem of generating *feature-based summaries* of customer reviews of products sold online. Here, *features* broadly mean product features (or attributes) and functions. Given a set of customer reviews of a particular product, the task involves three subtasks: (1) identifying features of the product that customers have expressed their opinions on (called *product features*); (2) for each feature, identifying review sentences that give positive or negative opinions; and (3) producing a summary using the discovered information.

Let us use an example to illustrate a feature-based summary. Assume that we summarize the reviews of a particular digital camera, *digital_camera*_1. The summary looks like the following:

*Digital_camera*_1:
Feature: **picture quality**
Positive: 253
<individual review sentences>
Negative: 6
<individual review sentences>
Feature: **size**
Positive: 134
<individual review sentences>
Negative: 10
<individual review sentences>
…

Figure 1: An example summary

In Figure 1, *picture quality* and (camera) *size* are the product features. There are 253 customer reviews that express positive opinions about the picture quality, and only 6 that express negative opinions. The <individual review sentences> link points to the specific sentences and/or the whole reviews that give positive or negative comments about the feature.

With such a feature-based summary, a potential customer can easily see how the existing customers feel about the digital camera. If he/she is very interested in a particular feature, he/she can drill down by following the <individual review sentences> link to see why existing customers like it and/or what they complain about. For a manufacturer, it is possible to combine summaries from multiple merchant sites to produce a single report for each of its products.

Our task is different from traditional text summarization [15, 39, 36] in a number of ways. First of all, a summary in our case is *structured* rather than another (but shorter) free text document as produced by most text summarization systems. Second, we are only interested in features of the product that customers have opinions on and also whether the opinions are positive or negative. We do not summarize the reviews by selecting or rewriting a subset of the original sentences from the reviews to capture their main points as in traditional text summarization.

As indicated above, our task is performed in three main steps:

(1) Mining product features that have been commented on by customers. We make use of both data mining and natural language processing techniques to perform this task. This part of the study has been reported in [19]. However, for completeness, we will summarize its techniques in this paper and also present a comparative evaluation.

(2) Identifying opinion sentences in each review and deciding whether each opinion sentence is positive or negative. Note that these opinion sentences must contain one or more product features identified above. To decide the *opinion orientation* of each sentence (whether the opinion expressed in the sentence is positive or negative), we perform three subtasks. First, a set of adjective words (which are normally used to express opinions) is identified using a natural language processing method. These words are also called *opinion words* in this paper. Second, for each opinion word, we determine its semantic orientation, e.g., positive or negative. A bootstrapping technique is proposed to perform this task using WordNet [29, 12]. Finally, we decide the opinion orientation of each sentence. An effective algorithm is also given for this purpose.

(3) Summarizing the results. This step aggregates the results of previous steps and presents them in the format of Figure 1.

Section 3 presents the detailed techniques for performing these tasks. A system, called FBS (*F*eature-*B*ased *S*ummarization), has also been implemented. Our experimental results with a large number of customer reviews of 5 products sold online show that FBS and its techniques are highly effectiveness.

2. RELATED WORK

Our work is closely related to Dave, Lawrence and Pennock's work in [9] on semantic classification of reviews. Using available training corpus from some Web sites, where each review already has a class (e.g., thumbs-up and thumbs-downs, or some other quantitative or binary ratings), they designed and experimented a number of methods for building sentiment classifiers. They show that such classifiers perform quite well with test reviews. They also used their classifiers to classify sentences obtained from Web search results, which are obtained by a search engine using a product name as the search query. However, the performance was limited because a sentence contains much less information than a review. Our work differs from theirs in three main aspects: (1) Our focus is not on classifying each review as a whole but on classifying each sentence in a review. Within a review some sentences may express positive opinions about certain product features while some other sentences may express negative opinions about some other product features. (2) The work in [9] does not mine product features from reviews on which the reviewers have expressed their opinions. (3) Our method does not need a corpus to perform the task.

In [30], Morinaga *et al.* compare reviews of different products in one category to find the reputation of the target product. However, it does not summarize reviews, and it does not mine product features on which the reviewers have expressed their opinions. Although they do find some frequent phrases indicating reputations, these phrases may not be product features (e.g., "doesn't work", "benchmark result" and "no problem(s)"). In [5], Cardie *et al* discuss opinion-oriented information extraction. They aim to create summary representations of opinions to perform question answering. They propose to use opinion-oriented "scenario templates" to act as summary representations of the opinions expressed in a document, or a set of documents. Our task is different. We aim to identify product features and user opinions on these features to automatically produce a summary. Also, no template is used in our summary generation.

Our work is also related to but different from subjective genre classification, sentiment classification, text summarization and terminology finding. We discuss each of them below.

2.1 Subjective Genre Classification

Genre classification classifies texts into different styles, e.g., "editorial", "novel", "news", "poem" etc. Although some techniques for genre classification can recognize documents that express opinions [23, 24, 14], they do not tell whether the opinions are positive or negative. In our work, we need to determine whether an opinion is positive or negative and to perform opinion classification at the sentence level rather than at the document level.

A more closely related work is [17], in which the authors investigate sentence subjectivity classification and concludes that the presence and type of adjectives in a sentence is indicative of whether the sentence is subjective or objective. However, their work does not address our specific task of determining the semantic orientations of those subjective sentences. Neither do they find features on which opinions have been expressed.

2.2 Sentiment Classification

Works of Hearst [18] and Sack [35] on sentiment-based classification of entire documents use models inspired by cognitive linguistics. Das and Chen [8] use a manually crafted lexicon in conjunction with several scoring methods to classify stock postings on an investor bulletin. Huettner and Subasic [20]

also manually construct a discriminant-word lexicon and use fuzzy logic to classify sentiments. Tong [41] generates sentiment timelines. It tracks online discussions about movies and displays a plot of the number of positive and negative sentiment messages over time. Messages are classified by looking for specific phrases that indicate the author's sentiment towards the movie (e.g., "great acting", "wonderful visuals", "uneven editing"). Each phrase must be manually added to a special lexicon and manually tagged as indicating positive or negative sentiment. The lexicon is domain dependent (e.g., movies) and must be rebuilt for each new domain. In contrast, in our work, we only manually create a small list of seed adjectives tagged with positive or negative labels. Our seed adjective list is also domain independent. An effective technique is proposed to grow this list using WordNet.

Turney's work in [42] applies a specific unsupervised learning technique based on the mutual information between document phrases and the words "excellent" and "poor", where the mutual information is computed using statistics gathered by a search engine. Pang et al. [33] examine several supervised machine learning methods for sentiment classification of movie reviews and conclude that machine learning techniques outperform the method that is based on human-tagged features although none of existing methods could handle the sentiment classification with a reasonable accuracy. Our work differs from these works on sentiment classification in that we perform classification at the sentence level while they determine the sentiment of each document. They also do not find features on which opinions have been expressed, which is very important in practice.

2.3 Text Summarization

Existing text summarization techniques mainly fall in one of the two categories: template instantiation and passage extraction. Work in the former framework includes [10, 39]. They emphasize on identification and extraction of certain core entities and facts in a document, which are packaged in a template. This framework requires background knowledge in order to instantiate a template to a suitable level of detail. Therefore, it is not domain or genre independent [37, 38]. This is different from our work as our techniques do not fill any template and are domain independent.

The passage extraction framework [e.g., 32, 25, 36] identifies certain segments of the text (typically sentences) that are the most representative of the document's content. Our work is different in that we do not extract representative sentences, but identify and extract those specific product features and the opinions related to them.

Boguraev and Kennedy [2] propose to find a few very prominent expressions, objects or events in a document and use them to help summarize the document. Our work is again different as we find all product features in a set of customer reviews regardless whether they are prominent or not. Thus, our summary is not a traditional text summary.

Most existing works on text summarization focus on a single document. Some researchers also studied summarization of multiple documents covering similar information. Their main purpose is to summarize the similarities and differences in the information content among these documents [27]. Our work is related but quite different because we aim to find the key features that are talked about in multiple reviews. We do not summarize similarities and differences of reviews.

2.4 Terminology Finding

In terminology finding, there are basically two techniques for discovering terms in corpora: symbolic approaches that rely on syntactic description of terms, namely noun phrases, and statistical approaches that exploit the fact that the words composing a term tend to be found close to each other and reoccurring [21, 22, 7, 6]. However, using noun phrases tends to produce too many non-terms (low precision), while using reoccurring phrases misses many low frequency terms, terms with variations, and terms with only one word. Our association mining based technique does not have these problems, and we can also find infrequent features by exploiting the fact that we are only interested in features that the users have expressed opinions on.

3. THE PROPOSED TECHNIQUES

Figure 2 gives the architectural overview of our opinion summarization system.

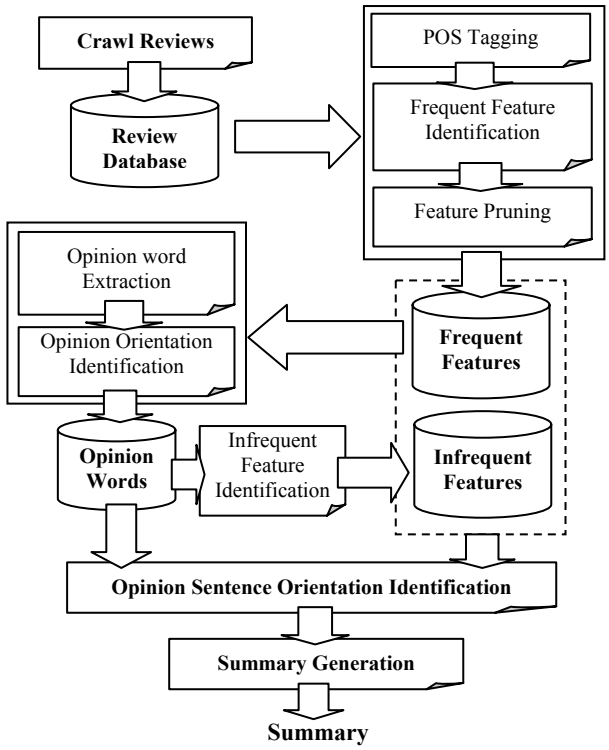

Figure 2: Feature-based opinion summarization

The inputs to the system are a product name and an entry Web page for all the reviews of the product. The output is the summary of the reviews as the one shown in the introduction section.

The system performs the summarization in three main steps (as discussed before): (1) mining product features that have been commented on by customers; (2) identifying opinion sentences in each review and deciding whether each opinion sentence is positive or negative; (3) summarizing the results. These steps are performed in multiple sub-steps.

Given the inputs, the system first downloads (or crawls) all the reviews, and put them in the review database. It then finds those "hot" (or frequent) features that many people have expressed their opinions on. After that, the opinion words are extracted using the

resulting frequent features, and semantic orientations of the opinion words are identified with the help of WordNet. Using the extracted opinion words, the system then finds those infrequent features. In the last two steps, the orientation of each opinion sentence is identified and a final summary is produced. Note that POS tagging is the part-of-speech tagging [28] from natural language processing, which helps us to find opinion features. Below, we discuss each of the sub-steps in turn.

3.1 Part-of-Speech Tagging (POS)

Product features are usually nouns or noun phrases in review sentences. Thus the part-of-speech tagging is crucial. We used the NLProcessor linguistic parser [31] to parse each review to split text into sentences and to produce the part-of-speech tag for each word (whether the word is a noun, verb, adjective, etc). The process also identifies simple noun and verb groups (syntactic chunking). The following shows a sentence with POS tags.

<S> <NG><W C='PRP' L='SS' T='w' S='Y'> ***I*** </W> </NG>
<VG> <W C='VBP'> ***am*** </W><W C='RB'> ***absolutely*** </W></VG> <W C='IN'> ***in*** </W> <NG> <W C='NN'> ***awe*** </W> </NG> <W C='IN'> ***of*** </W> <NG> <W C='DT'> ***this*** </W> <W C='NN'> ***camera*** </W></NG><W C='.'> **.** </W></S>

NLProcessor generates XML output. For instance, <W C='NN'> indicates a noun and <NG> indicates a noun group/noun phrase.

Each sentence is saved in the review database along with the POS tag information of each word in the sentence. A transaction file is then created for the generation of frequent features in the next step. In this file, each line contains words from one sentence, which includes only the identified nouns and noun phrases of the sentence. Other components of the sentence are unlikely to be product features. Some pre-processing of words is also performed, which includes removal of stopwords, stemming and fuzzy matching. Fuzzy matching is used to deal with word variants and misspellings [19].

3.2 Frequent Features Identification

This sub-step identifies product features on which many people have expressed their opinions. Before discussing frequent feature identification, we first give some example sentences from some reviews to describe what kinds of opinions that we will be handling. Since our system aims to find what people like and dislike about a given product, how to find the product features that people talk about is the crucial step. However, due to the difficulty of natural language understanding, some types of sentences are hard to deal with. Let us see an easy and a hard sentence from the reviews of a digital camera:

"The pictures are very clear."

In this sentence, the user is satisfied with the picture quality of the camera, *picture* is the feature that the user talks about. While the feature of this sentence is explicitly mentioned in the sentence, some features are implicit and hard to find. For example,

"While light, it will not easily fit in pockets."

This customer is talking about the *size* of the camera, but the word *size* does not appear in the sentence. In this work, we focus on finding features that appear explicitly as nouns or noun phrases in the reviews. We leave finding implicit features to our future work.

Here, we focus on finding frequent features, i.e., those features that are talked about by many customers (finding infrequent features will be discussed later). For this purpose, we use association mining [1] to find all frequent itemsets. In our context, an itemset is simply a set of words or a phrase that occurs together in some sentences.

The main reason for using association mining is because of the following observation. It is common that a customer review contains many things that are not directly related to product features. Different customers usually have different stories. However, when they comment on product features, the words that they use converge. Thus using association mining to find frequent itemsets is appropriate because those frequent itemsets are likely to be product features. Those noun/noun phrases that are infrequent are likely to be non-product features.

We run the association miner CBA [26], which is based on the Apriori algorithm in [1] on the transaction set of noun/noun phrases produced in the previous step. Each resulting frequent itemset is a possible feature. In our work, we define an itemset as frequent if it appears in more than 1% (minimum support) of the review sentences. The generated frequent itemsets are also called candidate *frequent features* in this paper.

However, not all candidate frequent features generated by association mining are genuine features. Two types of pruning are used to remove those unlikely features.

Compactness pruning: This method checks features that contain at least two words, which we call *feature phrases*, and remove those that are likely to be meaningless.

The association mining algorithm does not consider the position of an item (or word) in a sentence. However, in a sentence, words that appear together in a specific order are more likely to be meaningful phrases. Therefore, some of the frequent feature phrases generated by association mining may not be genuine features. Compactness pruning aims to prune those candidate features whose words do not appear together in a specific order. See [19] for the detailed definition of compactness and also the pruning procedure.

Redundancy pruning: In this step, we focus on removing redundant features that contain single words. To describe the meaning of redundant features, we use the concept of *p-support* (*pure support*). *p-support* of feature *ftr* is the number of sentences that *ftr* appears in as a noun or noun phrase, and these sentences must contain no feature phrase that is a superset of *ftr*.

We use a minimum p-support value to prune those redundant features. If a feature has a p-support lower than the minimum p-support (in our system, we set it to 3) and the feature is a subset of another feature phrase (which suggests that the feature alone may not be interesting), it is pruned. For instance, *life* by itself is not a useful feature while *battery life* is a meaningful feature phrase. See [19] for more explanations.

3.3 Opinion Words Extraction

We now identify opinion words. These are words that are primarily used to express subjective opinions. Clearly, this is related to existing work on distinguishing sentences used to

express subjective opinions from sentences used to objectively describe some factual information [43]. Previous work on subjectivity [44, 4] has established a positive statistically significant correlation with the presence of adjectives. Thus the presence of adjectives is useful for predicting whether a sentence is subjective, i.e., expressing an opinion. This paper uses adjectives as opinion words. We also limit the opinion words extraction to those sentences that contain one or more product features, as we are only interested in customers' opinions on these product features. Let us first define an opinion sentence.

Definition: *opinion sentence*

If a sentence contains one or more product features and one or more opinion words, then the sentence is called an *opinion sentence.*

We extract opinion words in the following manner (Figure 3):

```
for each sentence in the review database
  if (it contains a frequent feature, extract all the adjective
      words as opinion words)
    for each feature in the sentence
        the nearby adjective is recorded as its effective
        opinion. /* A nearby adjective refers to the adjacent
        adjective that modifies the noun/noun phrase that is a
        frequent feature. */
```

Figure 3: Opinion word extraction

For example, *horrible* is the effective opinion of *strap* in "*The strap is horrible and gets in the way of parts of the camera you need access to*." Effective opinions will be useful when we predict the orientation of opinion sentences.

3.4 Orientation Identification for Opinion Words

For each opinion word, we need to identify its semantic orientation, which will be used to predict the semantic orientation of each opinion sentence. The semantic orientation of a word indicates the direction that the word deviates from the norm for its semantic group. Words that encode a desirable state (e.g., beautiful, awesome) have a positive orientation, while words that represent undesirable states have a negative orientation (e.g., disappointing). While orientations apply to many adjectives, there are also those adjectives that have no orientation (e.g., external, digital) [17]. In this work, we are interested in only positive and negative orientations.

Unfortunately, dictionaries and similar sources, i.e., WordNet [29] do not include semantic orientation information for each word. Hatzivassiloglou and McKeown [16] use a supervised learning algorithm to infer the semantic orientation of adjectives from constraints on conjunctions. Although their method achieves high precision, it relies on a large corpus, and needs a large amount of manually tagged training data. In Turney's work [42], the semantic orientation of a phrase is calculated as the mutual information between the given phrase and the word "excellent" minus the mutual information between the given phrase and the word "poor". The mutual information is estimated by issuing queries to a search engine and noting the number of hits. The paper [42], however, does not report the results of semantic orientations of individual words/phrases. Instead it only gives the classification results of reviews. We do not use these techniques in this paper as both works rely on statistical information from a rather big corpus. Their methods are also inefficient. For example, in [42], for each word or phrase, a Web search and a substantial processing of the returned results are needed.

In this research, we propose a simple and yet effective method by utilizing the adjective synonym set and antonym set in WordNet [29] to predict the semantic orientations of adjectives.

In WordNet, adjectives are organized into bipolar clusters, as illustrated in Figure 4. The cluster for *fast/slow*, consists of two half clusters, one for senses of *fast* and one for senses of *slow*. Each half cluster is headed by a *head synset*, in this case *fast* and its antonym *slow*. Following the head synset is the *satellite synsets*, which represent senses that are similar to the sense of the head adjective. The other half cluster is headed by the reverse antonymous pair *slow/fast*, followed by satellite synsets for senses of *slow* [12].

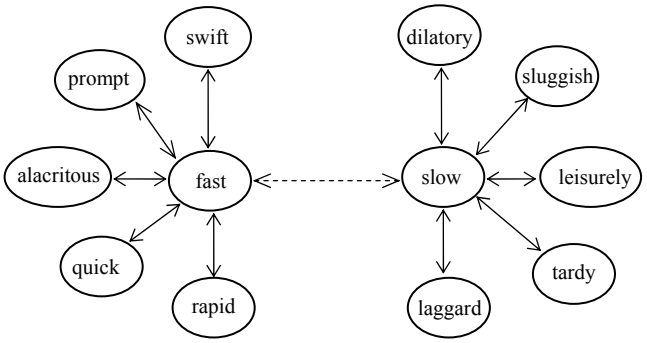

Figure 4: Bipolar adjective structure, (→ = similarity; ⇢ = antonymy)

In general, adjectives share the same orientation as their synonyms and opposite orientations as their antonyms. We use this idea to predict the orientation of an adjective. To do this, the synset of the given adjective and the antonym set are searched. If a synonym/antonym has known orientation, then the orientation of the given adjective could be set correspondingly. As the synset of an adjective always contains a sense that links to head synset, the search range is rather large. Given enough seed adjectives with known orientations, we can almost predict the orientations of all the adjective words in the review collection.

Thus, our strategy is to use a set of seed adjectives, which we know their orientations and then grow this set by searching in the WordNet. To have a reasonably broad range of adjectives, we first manually come up a set of very common adjectives (in our experiment, we used 30) as the seed list, e.g. positive adjectives: *great*, *fantastic*, *nice*, *cool* and negative adjectives: *bad*, *dull*. Then we resort to WordNet to predict the orientations of all the adjectives in the opinion word list. Once an adjective's orientation is predicted, it is added to the seed list. Therefore, the list grows in the process.

The complete procedure for predicting semantic orientations for all the adjectives in the opinion list is shown in Figure 5.

Procedure *OrientationPrediction* takes the adjective seed list and a set of opinion words whose orientations need to be determined.

It calls procedure *OrientationSearch* iteratively until no new opinion word is added to the seed list. Every time an adjective with its orientation is added to the seed list, the seed list is updated; therefore calling *OrientationSearch* repeatedly is necessary in order to exploit the newly added information.

```
1. Procedure OrientationPrediction(adjective_list, seed_list)
2. begin
3.    do {
4.       size1 = # of words in seed_list;
5.       OrientationSearch(adjective_list, seed_list);
6.       size2 = # of words in seed_list;
7.    } while (size1 ≠ size2);
8. end
```

```
1. Procedure OrientationSearch(adjective_list, seed_list)
2. begin
3.    for each adjective wi in adjective_list
4.    begin
5.       if (wi has synonym s in seed_list)
6.         { wi's orientation= s's orientation;
7.         add wi with orientation to seed_list; }
8.       else if (wi has antonym a in seed_list)
9.         { wi's orientation = opposite orientation of a's
                                  orientation;
10.        add wi with orientation to seed_list; }
11.   endfor;
12. end
```

Figure 5: Predicting the semantic orientations of opinion words

Procedure *OrientationSearch* searches WordNet and the seed list for each target adjective word to predict its orientation (line 3 to line 11). In line 5, it searches synset of the target adjective in WordNet and checks if any synonym has known orientation. If so, the target orientation is set to the same orientation as the synonym (line 6) and the target adjective along with the orientation is inserted into the seed list (line 7). Otherwise, the function continues to search antonym set of the target word in WordNet and checks if any antonym has known orientation (line 8). If so, the target orientation is set to the opposite of the antonym (line 9) and the target adjective with its orientation is inserted into the seed list (line 10). If neither synonyms nor antonyms of the target word have known orientation, the function just continues the same process for the next adjective since the word's orientation may be found in a later call of the procedure with an updated seed list.

For those adjectives that WordNet cannot recognize, they are discarded as they may not be valid words. For those that we cannot find orientations, they will also be removed from the opinion words list and the user will be notified for attention. If the user feels that the word is an opinion word and knows its sentiment, he/she can update the seed list. In our experiments, there is no user involvement (those removed opinion words are dropped). For the case that the synonyms/antonyms of an adjective have different known semantic orientations, we use the first found orientation as the orientation for the given adjective.

3.5 Infrequent Feature Identification

Frequent features are the "hot" features that people comment most about the given product. However, there are some features that only a small number of people talked about. These features can also be interesting to some potential customers and the manufacturer of the product. The question is how to extract these infrequent features (association mining is unable to identify such features)? Considering the following sentences:

"The pictures are absolutely amazing."

"The software that comes with it is amazing."

Sentences 1 and 2 share the same opinion word *amazing* yet describing different features: sentence 1 is about the *pictures*, and sentence 2 is about the *software*. Since one adjective word can be used to describe different objects, we could use the opinion words to look for features that cannot be found in the frequent feature generation step using association mining.

We extract infrequent features using the procedure in Figure 6:

```
for each sentence in the review database
   if (it contains no frequent feature but one or more opinion
       words)
      { find the nearest noun/noun phrase around the opinion
        word. The noun/noun phrase is stored in the feature
        set as an infrequent feature. }
```

Figure 6: Infrequent feature extraction

We use the *nearest* noun/noun phrase as the noun/noun phrase that the opinion word modifies because that is what happens most of the time. This simple heuristic seems to work well in practice.

A problem with the infrequent feature identification using opinion words is that it could also find nouns/noun phrases that are irrelevant to the given product. The reason for this is that people can use common adjectives to describe a lot of objects, including both interesting features that we want and irrelevant ones. This, however, is not a serious problem because the number of infrequent features, compared with the number of frequent features, is small. They account for around 15-20% of the total number of features as obtained in our experimental results. Infrequent features are generated for completeness. Moreover, frequent features are more important than infrequent ones. Since we rank features according to their p-supports, those wrong infrequent features will be ranked very low and thus will not affect most of the users.

3.6 Predicting the Orientations of Opinion Sentences

We now reach the step of predicting the orientation of an opinion sentence, i.e., positive or negative. In general, we use the dominant orientation of the opinion words in the sentence to determine the orientation of the sentence. That is, if positive/negative opinion prevails, the opinion sentence is regarded as a positive/negative one. In the case where there is the same number of positive and negative opinion words in the sentence, we predict the orientation using the average orientation of *effective opinions* or the orientation of the previous opinion sentence (recall that *effective opinion* is the closest opinion word for a feature in an opinion sentence). This is an effective method as our experimental results show. The detailed procedure is described in Figure 7.

Procedure *SentenceOrietation* deals with three situations in

```
1.      Procedure SentenceOrietation()
2.      begin
3.        for each opinion sentence s_i
4.        begin
5.          orientation = 0;
6.          for each opinion word op in s_i
7.            orientation += wordOrientation(op, s_i);
8.            /*Positive = 1, Negative = -1, Neutral = 0*/
9.          if (orientation > 0)  s_i's orientation = Positive;
10.         else  if (orientation < 0) s_i's orientation = Negative;
11.               else {
12.                 for each feature f in s_i
13.                   orientation +=
14.                   wordOrientation(f's effective opinion, s_i);
15.               if (orientation > 0)
16.                 s_i's orientation =  Positive;
17.               else if (orientation < 0)
18.                       s_i's orientation = Negative;
19.                    else s_i's orientation = s_i-1's orientation;
20.             }
21.       endfor;
22.     end
```

```
1.      Procedure wordOrientation(word, sentence)
2.      begin
3.        orientation = orientation of word in seed_list;
4.        If (there is NEGATION_WORD appears closely
              around word in sentence)
5.          orientation = Opposite(orientation);
6.      end
```

Figure 7: Predicting the orientations of opinion sentences

predicting the semantic orientation of an opinion sentence:

1. The user likes or dislikes most or all the features in one sentence. The opinion words are mostly either positive or negative, e.g., there are two positive opinion words, *good* and *exceptional* in "*overall this is a good camera with a really good picture clarity & an exceptional close-up shooting capability."*
2. The user likes or dislikes most of the features in one sentence, but there is an equal number of positive and negative opinion words, e.g., *"the auto and manual along with movie modes are very easy to use, but the software is not intuitive."* There is one positive opinion *easy* and one negative opinion *not intuitive*, although the user likes two features and dislikes one.
3. All the other cases.

For case 1, the dominant orientation can be easily identified (line 5 to 10 in the first procedure, *SentenceOrietation*). This is the most common case when people express their opinions. For case 2, we use the average orientation of effective opinions of features instead (line 12 to 18). Effective opinion is assumed to be the most related opinion for a feature. For case 3, we set the orientation of the opinion sentence to be the same as the orientation of previous opinion sentence (line 19). We use the context information to predict the sentence orientation because in most cases, people express their positive/negative opinions together in one text segment, i.e., a few consecutive sentences.

For a sentence that contains a *but* clause (sub-sentence that starts with *but*, *however*, etc.), which indicates sentimental change for the features in the clause, we first use the effective opinion in the clause to decide the orientation of the features. If no opinion appears in the clause, the opposite orientation of the sentence will be used.

Note that in the procedure *wordOrientation*, we do not simply take the semantic orientation of the opinion word from the set of opinion words as its orientation in the specific sentence. We also consider whether there is a negation word such as "no", "not", "yet", appearing *closely* around the opinion word. If so, the opinion orientation of the sentence is the opposite of its original orientation (lines 4 and 5). By *closely* we mean that the word distance between a negation word and the opinion word should not exceed a threshold (in our experiment, we set it to 5). This simple method deals with the sentences like "*the camera is not easy to use*", and "*it would be nicer not to see little zoom sign on the side*". This method is quite effective in most cases.

3.7 Summary Generation

After all the previous steps, we are ready to generate the final feature-based review summary, which is straightforward and consists of the following steps:

- For each discovered feature, related opinion sentences are put into positive and negative categories according to the opinion sentences' orientations. A count is computed to show how many reviews give positive/negative opinions to the feature.
- All features are ranked according to the frequency of their appearances in the reviews. Feature phrases appear before single word features as phrases normally are more interesting to users. Other types of rankings are also possible. For example, we can also rank features according the number of reviews that express positive or negative opinions.

The following shows an example summary for the feature "*picture*" of a digital camera. Note that the individual opinion sentences (and their corresponding reviews, which are not shown here) can be hidden using a hyperlink in order to enable the user to see a global view of the summary easily.

Feature: **picture**

Positive: 12

- Overall this is a good camera with a really good picture clarity.
- The pictures are absolutely amazing - the camera captures the minutest of details.
- After nearly 800 pictures I have found that this camera takes incredible pictures.

…

Negative: 2

- The pictures come out hazy if your hands shake even for a moment during the entire process of taking a picture.
- Focusing on a display rack about 20 feet away in a brightly lit room during day time, pictures produced by this camera were blurry and in a shade of orange.

Table 1: Recall and precision at each step of feature generation

Product name	No. of manual features	Frequent features (association mining)		Compactness pruning		P-support pruning		Infrequent feature identification	
		Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision
Digital camera1	79	0.671	0.552	0.658	0.634	0.658	0.825	0.822	0.747
Digital camera2	96	0.594	0.594	0.594	0.679	0.594	0.781	0.792	0.710
Cellular phone	67	0.731	0.563	0.716	0.676	0.716	0.828	0.761	0.718
Mp3 player	57	0.652	0.573	0.652	0.683	0.652	0.754	0.818	0.692
DVD player	49	0.754	0.531	0.754	0.634	0.754	0.765	0.797	0.743
Average	**69**	**0.68**	**0.56**	**0.67**	**0.66**	**0.67**	**0.79**	**0.80**	**0.72**

Table 2: Recall and precision of FASTR

	Recall	Precision	No. terms
Digital camera1	0.1898	0.0313	479
Digital camera2	0.1875	0.0442	407
Cellular phone	0.1493	0.0275	364
Mp3 player	0.1403	0.0214	374
DVD player	0.1633	0.0305	262
Average	**0.1660**	**0.0309**	**377.2**

4. EXPERIMENTAL EVALUATION

A system, called FBS (*F*eature-*B*ased *S*ummarization), based on the proposed techniques has been implemented in C++. We now evaluate FBS from three perspectives:

1. The effectiveness of feature extraction.
2. The effectiveness of opinion sentence extraction.
3. The accuracy of orientation prediction of opinion sentences.

We conducted our experiments using the customer reviews of five electronics products: 2 digital cameras, 1 DVD player, 1 mp3 player, and 1 cellular phone. The reviews were collected from Amazon.com and C|net.com. Products in these sites have a large number of reviews. Each of the reviews includes a text review and a title. Additional information available but not used in this project includes date, time, author name and location (for Amazon reviews), and ratings.

For each product, we first crawled and downloaded the first 100 reviews. These review documents were then cleaned to remove HTML tags. After that, NLProcessor [31] is used to generate part-of-speech tags. Our system is then applied to perform summarization.

For evaluation, we manually read all the reviews. For each sentence in a review, if it shows user's opinions, all the features on which the reviewer has expressed his/her opinion are tagged. Whether the opinion is positive or negative (i.e., the orientation) is also identified. If the user gives no opinion in a sentence, the sentence is not tagged as we are only interested in sentences with opinions in this work. For each product, we produced a manual feature list. Column "No. of manual features" in Table 1 shows the number of manual features for each product. All the results generated by our system are compared with the manually tagged results. Tagging is fairly straightforward for both product features and opinions. A minor complication regarding feature tagging is that features can be explicit or implicit in a sentence. Most features appear explicitly in opinion sentences, e.g., *pictures* in *"the pictures are absolutely amazing"*. Some features may not appear in sentences. We call such features implicit features, e.g., *size* in "*it fits in a pocket nicely*". Both explicit and implicit features are easy to identify by the human tagger.

Another issue is that judging opinions in reviews can be somewhat subjective. It is usually easy to judge whether an opinion is positive or negative if a sentence clearly expresses an opinion. However, deciding whether a sentence offers an opinion or not can be debatable. For those difficult cases, a consensus was reached between the primary human tagger (the first author of the paper) and the secondary tagger (the second author of the paper).

Table 1 gives the precision and recall results of the feature generation function of FBS. We evaluated the results at each step of our algorithm. In the table, column 1 lists each product. Columns 3 and 4 give the recall and precision of frequent feature generation for each product, which uses association mining. The results indicate that the frequent features contain a lot of errors. Using this step alone gives poor results, i.e., low precision. Columns 5 and 6 show the corresponding results after compactness pruning is performed. We can see that the precision is improved significantly by this pruning. The recall stays steady. Columns 7 and 8 give the results after pruning using p-support. There is another dramatic improvement in the precision. The recall level almost does not change. The results from Columns 4-8 clearly demonstrate the effectiveness of these two pruning techniques. Columns 9 and 10 give the results after infrequent feature identification is done. The recall is improved dramatically. The precision drops a few percents on average. However, this is not a major problem because the infrequent features are ranked rather low, and thus will not affect most users.

To further illustrate the effectiveness of our feature extraction

step, we compared the features generated using our method with terms found by the well known and publicly available term extraction and indexing system, FASTR [11] of Christian Jacquemin. Table 2 shows the recall and precision of FASTR.

We observe that both the average recall and precision of FASTR are significantly lower than those of our method. After a close inspection of the terms generated by FASTR, we see that there are two major reasons that lead to its poor results. First of all, FASTR generates a large number of terms, as shown in the fourth column "No. terms" of Table 2. The average number of terms found by FASTR is 377. Most of these terms are not product features at all (although many of them may be noun phrases). Secondly, FASTR does not find one-word terms, but only term phrases that consist of two or more words. Our feature extraction method finds both one-word terms and term phrases. Comparing the results in Table 1 and Table 2, we can clearly see that the proposed method is much more effective for our task.

Table 3 shows the evaluation results of the other two procedures: opinion sentence extraction and sentence orientation prediction. The average recall of opinion sentence extraction is nearly 70%. The average precision of opinion sentence extraction is 64%. Note that as indicated earlier determining whether a sentence expresses an opinion is subjective. Our result analysis indicates that people like to describe their "stories" with the product lively: they often mention the situation that they used the product, the detailed product features used, and also the results they got. While human taggers do not regard these sentences as opinion sentences as there is no indication of whether the user likes the features or not, our system labels these sentences as opinion sentences because they contain both product features and some opinion adjectives. This decreases precision. Although these sentences may not show strong user opinions towards the product features, they may still be beneficial and useful.

Our system has a good accuracy in predicting sentence orientations: the average accuracy for the five products is 84%. This shows that our method of using WordNet to predict adjective semantic orientations and orientations of opinion sentences are highly effective.

Table 3: Results of opinion sentence extraction and sentence orientation prediction

Product name	Opinion sentence extraction		Sentence orientation accuracy
	Recall	Precision	
Digital camera1	0.719	0.643	0.927
Digital camera2	0.634	0.554	0.946
Cellular phone	0.675	0.815	0.764
Mp3 player	0.784	0.589	0.842
DVD player	0.653	0.607	0.730
Average	**0.693**	**0.642**	**0.842**

In summary, we can see that our techniques are very promising, especially for sentence orientation prediction. We believe that they may be used in practical settings. We also note three main limitations of our system: (1) We have not dealt with opinion sentences that need pronoun resolution [40]. For instance, "*it is quiet but powerful*". To understand what *it* represents, pronoun resolution needs to be performed. Pronoun resolution is a complex and computational expensive problem in natural language processing (NLP). We plan to adapt some existing techniques from NLP to suit our needs. (2) We only used adjectives as indicators of opinion orientations of sentences. However, verbs and nouns can also be used for the purpose, e.g., "*I like the feeling of the camera*", "*I highly recommend the camera*". We plan to address this issue in the future. (3) It is also important to study the strength of opinions. Some opinions are very strong and some are quite mild. Highlighting strong opinions (strongly like or dislike) can be very useful for both individual shoppers and product manufacturers.

5. CONCLUSIONS

In this paper, we proposed a set of techniques for mining and summarizing product reviews based on data mining and natural language processing methods. The objective is to provide a feature-based summary of a large number of customer reviews of a product sold online. Our experimental results indicate that the proposed techniques are very promising in performing their tasks. We believe that this problem will become increasingly important as more people are buying and expressing their opinions on the Web. Summarizing the reviews is not only useful to common shoppers, but also crucial to product manufacturers.

In our future work, we plan to further improve and refine our techniques, and to deal with the outstanding problems identified above, i.e., pronoun resolution, determining the strength of opinions, and investigating opinions expressed with adverbs, verbs and nouns. Finally, we will also look into monitoring of customer reviews. We believe that monitoring will be particularly useful to product manufacturers because they want to know any new positive or negative comments on their products whenever they are available. The keyword here is *new*. Although a new review may be added, it may not contain any new information.

6. ACKNOWLEDGMENTS

The research in this paper was supported by the National Science Foundation under the grant NSF IIS-0307239.

7. REFERENCES

[1]. Agrawal, R. & Srikant, R. 1994. Fast algorithm for mining association rules. *VLDB'94*, 1994.

[2]. Boguraev, B., and Kennedy, C. 1997. Salience-Based Content Characterization of Text Documents. In *Proc. of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization.*

[3]. Bourigault, D. 1995. Lexter: A terminology extraction software for knowledge acquisition from texts. *KAW'95*.

[4]. Bruce, R., and Wiebe, J. 2000. Recognizing Subjectivity: A Case Study of Manual Tagging. *Natural Language Engineering*.

[5]. Cardie, C., Wiebe, J., Wilson, T. and Litman, D. 2003. Combining Low-Level and Summary Representations of Opinions for Multi-Perspective Question Answering. *2003 AAAI Spring Symposium on New Directions in Question Answering.*

[6]. Church, K.W. and Hanks, P. 1990. Word Association Norms, Mutual Information and Lexicography.

Computational Linguistics, 16(1):22-29.

[7]. Daille, B. 1996. Study and Implementation of Combined Techniques for Automatic Extraction of Terminology. *The Balancing Act: Combining Symbolic and Statistical Approaches to Language*. MIT Press, Cambridge

[8]. Das, S. and Chen, M., 2001. Yahoo! for Amazon: Extracting market sentiment from stock message boards. *APFA'01*.

[9]. Dave, K., Lawrence, S., and Pennock, D., 2003. Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews. *WWW'03*.

[10]. DeJong, G. 1982. An Overview of the FRUMP System. *Strategies for Natural Language Parsing*. 149-176.

[11]. FASTR. http://www.limsi.fr/Individu/jacquemi/FASTR/

[12]. Fellbaum, C. 1998. *WordNet: an Electronic Lexical Database*, MIT Press.

[13]. Finn, A. and Kushmerick, N. 2003. Learning to Classify Documents according to Genre. *IJCAI-03 Workshop on Computational Approaches to Style Analysis and Synthesis*.

[14]. Finn, A., Kushmerick, N., and Smyth, B. 2002. Genre Classification and Domain Transfer for Information Filtering. In *Proc. of European Colloquium on Information Retrieval Research*, pages 353-362.

[15]. Goldstein, J., Kantrowitz, M., Mittal, V., and Carbonell, J. 1999. Summarizing Text Documents: Sentence Selection and Evaluation Metrics. *SIGIR'99*.

[16]. Hatzivassiloglou, V. and Mckeown, K., 1997. Predicting the Semantic Orientation of Adjectives. In *Proc. of 35th ACL/8th EACL*.

[17]. Hatzivassiloglou, V. and Wiebe, 2000. J. Effects of Adjective Orientation and Gradability on Sentence Subjectivity. *COLING'00*.

[18]. Hearst, M, 1992. Direction-based Text Interpretation as an Information Access Refinement. In Paul Jacobs, editor, *Text-Based Intelligent Systems*. Lawrence Erlbaum Associates.

[19]. Hu, M., and Liu, B. 2004. Mining Opinion Features in Customer Reviews. To appear in *AAAI'04*, 2004.

[20]. Huettner, A. and Subasic, P., 2000. Fuzzy Typing for Document Management. In *ACL'00 Companion Volume: Tutorial Abstracts and Demonstration Notes*.

[21]. Jacquemin, C., and Bourigault, D. 2001. Term extraction and automatic indexing. In R. Mitkov, editor, *Handbook of Computational Linguistics*. Oxford University Press.

[22]. Justeson, J. S., and Katz, S.M. 1995. Technical Terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering* 1(1):9-27.

[23]. Karlgren, J. and Cutting, D. 1994. Recognizing Text Genres with Simple Metrics using Discriminant Analysis. *COLING'94*.

[24]. Kessler, B., Nunberg, G., and Schutze, H. 1997. Automatic Detection of Text Genre. In *Proc. of 35th ACL/8th EACL*.

[25]. Kupiec, J., Pedersen, J., and Chen, F. 1995. A Trainable Document Summarizer. *SIGIR'1995*

[26]. Liu, B., Hsu, W., Ma, Y. 1998. Integrating Classification and Association Rule Mining. *KDD'98*, 1998.

[27]. Mani, I., and Bloedorn, E., 1997. Multi-document Summarization by Graph Search and Matching. *AAAI'97*.

[28]. Manning, C. and Schütze, H. 1999. *Foundations of Statistical Natural Language Processing*, MIT Press. Cambridge, MA: May 1999.

[29]. Miller, G., Beckwith, R, Fellbaum, C., Gross, D., and Miller, K. 1990. Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography (special issue)*, 3(4):235-312.

[30]. Morinaga, S., Ya Yamanishi, K., Tateishi, K, and Fukushima, T. 2002. Mining Product Reputations on the Web. *KDD'02*.

[31]. NLProcessor – *Text Analysis Toolkit*. 2000. http://www.infogistics.com/textanalysis.html

[32]. Paice, C. D. 1990. Constructing Literature Abstracts by Computer: Techniques and Prospects. *Information Processing and Management* 26:171-186.

[33]. Pang, B., Lee, L., and Vaithyanathan, S., 2002. Thumbs up? Sentiment Classification Using Machine Learning Techniques. In *Proc. of EMNLP 2002*

[34]. Reimer, U. and Hahn, U. 1997. A Formal Model of Text Summarization based on Condensation Operators of a Terminological Logic. *In Proceedings of ACL'97 Workshop on Intelligent, Scalable Text Summarization*.

[35]. Sack, W., 1994. On the Computation of Point of View. *AAAI'94*, Student abstract.

[36]. Salton, G. Singhal, A. Buckley, C. and Mitra, M. 1996. Automatic Text Decomposition using Text Segments and Text Themes. *ACM Conference on Hypertext.*

[37]. Sparck J. 1993a. Discourse Modeling for Automatic Text Summarizing. *Technical Report 290,* University of Cambridge Computer Laboratory.

[38]. Sparck J. 1993b. What might be in a summary? *Information Retrieval* 93: 9-26.

[39]. Tait, J. 1983. *Automatic Summarizing of English Texts.* Ph.D. Dissertation, University of Cambridge.

[40]. Tetreault, J. 1999. Analysis of Syntax-Based Pronoun Resolution Methods. *ACL'99*.

[41]. Tong, R., 2001. An Operational System for Detecting and Tracking Opinions in on-line discussion. *SIGIR 2001 Workshop on Operational Text Classification*.

[42]. Turney, P. 2002. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. *ACL'02*.

[43]. Wiebe, J. 2000. Learning Subjective Adjectives from Corpora. *AAAI'00*.

[44]. Wiebe, J., Bruce, R., and O'Hara, T. 1999. Development and Use of a Gold Standard Data Set for Subjectivity Classifications. In *Proc. of ACL'99*.