A Holistic Lexicon-Based Approach to Opinion Mining

Xiaowen Ding
Department of Computer Science
University of Illinois at Chicago
851 S. Morgan Street
Chicago, IL 60607-0753

xding@cs.uic.edu

Bing Liu
Department of Computer Science
University of Illinois at Chicago
851 S. Morgan Street
Chicago, IL 60607-0753

liub@cs.uic.edu

Philip S. Yu
Department of Computer Science
University of Illinois at Chicago
851 S. Morgan Street
Chicago, IL 60607-0753

psyu@cs.uic.edu

ABSTRACT

One of the important types of information on the Web is the opinions expressed in the user generated content, e.g., customer reviews of products, forum posts, and blogs. In this paper, we focus on customer reviews of products. In particular, we study the problem of determining the semantic orientations (positive, negative or neutral) of opinions expressed on product features in reviews. This problem has many applications, e.g., opinion mining, summarization and search. Most existing techniques utilize a list of *opinion* (*bearing*) *words* (also called *opinion lexicon*) for the purpose. Opinion words are words that express desirable (e.g., great, amazing, etc.) or undesirable (e.g., bad, poor, etc) states. These approaches, however, all have some major shortcomings. In this paper, we propose a *holistic lexicon-based approach* to solving the problem by exploiting external evidences and linguistic conventions of natural language expressions. This approach allows the system to handle opinion words that are context dependent, which cause major difficulties for existing algorithms. It also deals with many special words, phrases and language constructs which have impacts on opinions based on their linguistic patterns. It also has an effective function for aggregating multiple conflicting opinion words in a sentence. A system, called Opinion Observer, based on the proposed technique has been implemented. Experimental results using a benchmark product review data set and some additional reviews show that the proposed technique is highly effective. It outperforms existing methods significantly.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Information filtering*. I.2.7 [**Natural Language Processing**] – *Text analysis*

General Terms

Algorithms, Experimentation.

Keywords

Opinion mining, sentiment analysis, context dependent opinions.

1. INTRODUCTION

With the rapid expansion of e-commerce over the past 10 years, more and more products are sold on the Web, and more and more people are buying products online. In order to enhance customer shopping experience, it has become a common practice for online merchants to enable their customers to write reviews on products that they have purchased. With more and more 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 or more at some large merchant sites. Many reviews are also long, which 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 only gets a biased view. The large number of reviews also makes it hard for product manufacturers or businesses to keep track of customer opinions and sentiments on their products and services. It is thus highly desirable to produce a summary of reviews [13, 21] (see below and also Section 3).

In the past few years, many researchers studied the problem, which is called *opinion mining* or *sentiment analysis* [1, 3, 13, 15, 28, 37]. The main tasks are (1) to find product features that have been commented on by reviewers and (2) to decide whether the comments are positive or negative. Both tasks are very challenging. In this paper, we focus on task (2). That is, given a set of product features of a product, we want to accurately identify the semantic orientations of opinions expressed on each product feature by each reviewer. *Semantic orientation* means whether the opinion is positive, negative or neutral. We will formally define the problem in Section 3, where we will see that our task is realistic and has many applications. Although several works on opinion mining exist, there is still not a general framework or model that clearly articulates various aspects of the problem and their relationships. We make an attempt in this paper in Section 3.

In [13], a *lexicon-based* method is proposed to use *opinion bearing words* (or simply *opinion words*) to perform task (2). Opinion words are words that are commonly used to express positive or negative opinions (or sentiments), e.g., "amazing", "great", "poor" and "expensive". The method basically counts the number of positive and negative opinion words that are near the product feature in each review sentence. If there are more positive opinion words than negative opinion words, the final opinion on the feature is positive and otherwise negative. The *opinion lexicon* or the set of opinion words was obtained through a bootstrapping process using WordNet (http://wordnet.princeton.edu/) [8]. This method is simple and efficient, and gives reasonable results. However, this technique has some major shortcomings.

First of all, it does not have an effective mechanism for dealing with context dependent opinion words. There are many such words. For example, the word "small" can indicate a positive or a negative opinion on a product feature depending on the product feature and the context. There is probably no way to know the semantic orientation of a context dependent opinion word by looking at only the word and the product feature that it modifies without prior knowledge of the product or the product feature. Asking a domain expert or user to provide such knowledge is not scalable due to the huge number of products, product features and opinion words. Several researchers have attempted the problem [11, 16, 28]. However, their approaches still have some major limitations as we will see in the next section. In this paper, we propose a *holistic lexicon-based approach* to solving the problem, which improves the lexicon-based method in [13]. Instead of looking at the current sentence alone, this approach exploits external information and evidences in other sentences and other reviews, and some linguistic conventions in natural language expressions to infer orientations of opinion words. No prior domain knowledge or user inputs are needed. Based on our experiment results, we are fairly confident to say that context dependent opinion words no longer present a major problem.

Second, when there are multiple conflicting opinion words in a sentence, existing methods are unable to deal with them well. We propose a new method to aggregate orientations of such words by considering the distance between each opinion word and the product feature. This turns out to be highly effective.

To complete the proposed approach, a set of linguistic patterns are devised to handle special words, phrases and constructs based on their underlying meanings or usage patterns, which have not been handled satisfactorily so far by existing methods.

The proposed technique has been evaluated using the benchmark review data set used in [13, 28] which consists of a large number of reviews of five products, and also a new data set consisting of reviews of three products. The results show that the new method outperforms the existing methods significantly.

2. RELATED WORK

Opinion analysis has been studied by many researchers in recent years. Two main research directions are sentiment classification and feature-based opinion mining. Sentiment classification investigates ways to classify each review document as positive, negative, or neutral. Representative works on classification at the document level include [4, 5, 9, 12, 26, 27, 29, 32]. These works are different from ours as we are interested in opinions expressed on each product feature rather than the whole review.

Sentence level subjectivity classification is studied in [10], which determines whether a sentence is a subjective sentence (but may not express a positive or negative opinion) or a factual one. Sentence level sentiment or opinion classification is studied in [10, 13, 17, 23, 28, 33, etc]. Our work is different from the sentence level analysis as we identify opinions on each feature. A review sentence can contain multiple features, and the orientations of opinions expressed on the features can also be different, e.g., "the voice quality of this phone is great and so is the reception, but the battery life is short." "voice quality", "reception" and "battery life" are features. The opinion on "voice quality", "reception" are positive, and the opinion on "battery life" is negative. Other related works at both the document and sentence levels include those in [2, 10, 15, 16, 36].

Most sentence level and even document level classification methods are based on identification of opinion words or phrases. There are basically two types of approaches: (1) corpus-based approaches, and (2) dictionary-based. approaches. Corpus-based approaches find co-occurrence patterns of words to determine the sentiments of words or phrases, e.g., the works in [10, 32, 34]. Dictionary-based approaches use synonyms and antonyms in WordNet to determine word sentiments based on a set of seed opinion words. Such approaches are studied in [1, 8, 13, 17].

[13] proposes the idea of opinion mining and summarization. It uses a *lexicon-based* method to determine whether the opinion expressed on a product feature is positive or negative. A related method is used in [17]. These methods are improved in [28] by a more sophisticated method based on relaxation labeling. We will show in Section 5 that the proposed technique performs much better than both these methods. In [37], a system is reported for analyzing movie reviews in the same framework. However, the system is domain specific. Other recent work related to sentiment analysis includes [3, 15, 16, 18, 19, 20, 21, 22, 24, 30, 34]. [14] studies the extraction of comparative sentences and relations, which is different from this work as we do not deal with comparative sentences in this research.

Our holistic lexicon-based approach to identifying the orientations of context dependent opinion words is closely related to works that identify domain opinion words [11, 16]. Both [11] and [16] use *conjunction rules* to find such words from large domain corpora. The conjunction rule basically states that when two opinion words are linked by "and" in a sentence, their opinion orientations are the same. For example, in the sentence, "this room is beautiful and spacious", both "beautiful" and "spacious" are positive opinion words. Based on this rule or language convention, if we do not know whether "spacious" is positive or negative, but know that "beautiful" is positive, we can infer that "spacious" is also positive. Although our approach will also use this linguistic rule or convention, our method is different in two aspects. First, we argue that finding domain opinion words is still problematic because in the same domain the same word may indicate different opinions depending on what features it is applied to. For example, in the following review sentences in the camera domain, "the battery life is very *long*" and "it takes a *long* time to focus", "long" is positive in the first sentence, but negative in the second. Thus, we need to consider both the feature and the opinion word rather than only the opinion word as in [11, 16]. Second, our approach does not need to find opinion orientations of domain opinion words up front or offline. It makes the decision whenever needed or online. It is more flexible. [28] also uses similar rules to compute opinion orientations based on relaxation labeling. However, as we will see, [28] produces poorer results than the proposed method.

3. PROBLEM DEFINITION

This section first defines the general problem of semantic analysis of reviews and then highlights the specific instance of the problem that we aim to solve.

In general, opinions can be expressed on anything, e.g., a product, an individual, an organization, an event, a topic, etc. We use the general term *object* to denote the entity that has been commented on. The object has a set of *components* (or parts) and also a set of attributes (or properties). Thus the object can be hierarchically decomposed according to the *part-of* relationship, i.e., each component may also have its sub-components and so on. For

example, a product (e.g., a car, a digital camera) can have different components, an event can have sub-events, a topic can have sub-topics, etc. Formally, we have the following definition:

Definition (object): An *object O* is an entity which can be a product, person, event, organization, or topic. It is associated with a pair, *O*: (*T*, *A*), where *T* is a hierarchy or taxonomy of *components* (or *parts*), *sub-components*, and so on, and *A* is a set of *attributes* of *O*. Each component has its own set of sub-components and attributes.

Example 1: A particular brand of digital camera is an object. It has a set of components, e.g., lens, battery, etc., and also a set of attributes, e.g., picture quality, size, etc. The battery component also has its set of attributes, e.g., battery life, battery size, etc.

Essentially, an object is represented as a tree. The root is the object itself. Each non-root node is a component or sub-component of the object. Each link is a *part-of* relationship. Each node is also associated with a set of attributes. An opinion can be expressed on any node and any attribute of the node.

Example 2: Following Example 1, one can express an opinion on the camera (the root node), e.g., "I do not like this camera", or on one of its attributes, e.g., "the picture quality of this camera is poor". Likewise, one can also express an opinion on any one of the camera's components or the attribute of the component.

To simplify our discussion, we use the word "features" to represent both components and attributes, which allows us to omit the hierarchy. Using features for products is also quite common in practice. For an ordinary user, it is probably too complex to use a hierarchical representation of features and opinions. We note that in this framework the object itself is also treated as a feature.

Let the review be *r*. In the most general case, *r* consists of a sequence of sentences $r = \langle s_1, s_2, \ldots, s_m \rangle$.

Definition (explicit and implicit feature): If a feature *f* appears in review *r*, it is called an *explicit feature* in *r*. If *f* does not appear in *r* but is implied, it is called an *implicit feature* in *r*.

Example 3: "battery life" in the following sentence is an explicit feature:

"The battery life of this camera is too short".

"Size" is an implicit feature in the following sentence as it does not appear in the sentence but it is implied:

"This camera is too large".

Here, "large" is called a *feature indicator*.

Definition (opinion passage on a feature): The *opinion passage* on feature *f* of an object evaluated in *r* is a group of consecutive sentences in *r* that expresses a positive or negative opinion on *f*.

It is possible that a sequence of sentences (at least one) in a review together expresses an opinion on an object or a feature of the object. Also, it is possible that a single sentence expresses opinions on more than one feature:

"The picture quality is good, but the battery life is short".

Most current research focuses on sentences, i.e., each passage consisting of a single sentence. In our subsequent discussion, we use sentences and passages interchangeably as we work on sentences as well.

Definition (explicit and implicit opinion): An *explicit opinion* on feature *f* is a subjective sentence that directly expresses a

positive or negative opinion. An *implicit opinion* on feature *f* is an objective sentence that implies an opinion.

Example 4: The following sentence expresses an explicit positive opinion:

"The picture quality of this camera is amazing."

The following sentence expresses an implicit negative opinion:

"The earphone broke in two days."

Although this sentence states an objective fact (assume it is true), it implicitly expresses a negative opinion on the earphone.

Definition (opinion holder): The *holder* of a particular opinion is the person or the organization that holds the opinion.

In the case of product reviews, forum postings and blogs, opinion holders are usually the authors of the postings. Opinion holders are more important in news articles because they often explicitly state the person or organization that holds a particular view. For example, the opinion holder in the sentence "John expressed his disagreement on the treaty" is "John". In this work, we will not study opinion holders (see [17]).

Definition (semantic orientation of an opinion): The *semantic orientation* of an opinion on a feature *f* states whether the opinion is positive, negative or neutral.

We now put things together to define a *model* of an object and a set of opinions on the object. An object is represented with a finite set of features, $F = \{f_1, f_2, \ldots, f_n\}$. Each feature f_i in F can be expressed with a finite set of words or phrases W_i, which are *synonyms*. That is, we have a set of corresponding synonym sets $W = \{W_1, W_2, \ldots, W_n\}$ for the *n* features. Since each feature f_i in F has a name (denoted by f_i), then $f_i \in W_i$. Each author or *opinion holder j* comments on a subset of the features $S_j \subseteq F$. For each feature $f_k \in S_j$ that opinion holder *j* comments on, he/she chooses a word or phrase from W_k to describe the feature, and then expresses a positive, negative or neutral opinion on it.

This simple model covers most but not all cases. For example, it does not cover the situation described in the following sentence: "the view-finder and the lens of this camera are too close", which expresses a negative opinion on the distance of the two components. However, we will use this simplified model in the rest of this paper. The above cases are rare in product reviews.

This model introduces three main practical problems. Given a collection of reviews *D* as input, we have:

Problem 1: Both *F* and *W* are unknown. Then, in opinion analysis, we need to perform three tasks:

Task 1: Identifying and extracting object features that have been commented on in each review $d \in D$.

Task 2: Determining whether the opinions on the features are positive, negative or neutral.

Task 3: Grouping synonyms of features, as different people may use different words to express the same feature.

Problem 2: *F* is known but *W* is unknown. This is similar to Problem 1, but slightly easier. All the three tasks for Problem 1 still need to be performed, but Task 3 becomes the problem of matching discovered features with the set of given features *F*.

Problem 3: *W* is known (then *F* is also known). We only need to perform Task 2 above, namely, determining whether the

opinions on the known features are positive, negative or neutral after all the sentences that contain them are extracted.

Clearly, the first problem is the most difficult to solve. Problem 2 is slightly easier. Problem 3 is the easiest, but still realistic.

Example 5: A cellular phone company wants to analyze customer reviews on a few models of its phones. It is quite realistic to produce the feature set F that the company is interested in and also the set of synonyms of each feature W_i (although the set might not be complete). Then there is no need to perform Tasks 1 and 3 (which are very challenging problems).

Output: The final output for each evaluative text d is a set of pairs. Each pair is denoted by (f, SO), where f is a feature and SO is the semantic or opinion orientation (positive or negative) expressed in d on feature f. We ignore neutral opinions in the output as they are not usually useful.

This model covers most but not all cases. For example, it does not cover the situation described in the following sentence: "the viewfinder and the lens of this camera are too close", which expresses a negative opinion on the distance of the two components. However, such cases are rare in practice. We will use this simplified model in the rest of this paper. Note also that this model does not consider the strength of each opinion [33], i.e., whether the opinion is strongly negative (or positive) or weakly negative (or positive), but it can be added.

There are many ways to use the results. A simple way is to produce a feature-based summary of opinions on the object [13]. That is, for each feature, we can show how many reviewers expressed negative opinions and how many reviewers expressed positive opinions. What is important is that this is a structured summary produced from unstructured text. The summary can also be easily visualized to give a clear view of opinions on different object features from existing users [21].

The rest of the paper focuses on solving Problem 3. That is, we assume that all features are given, which is realistic for specific domains as Example 5 shows. Our task is to determine whether the opinion expressed by each reviewer on each product feature is positive, negative or neutral.

4. THE PROPOSED TECHNIQUE

We now present the proposed technique. The main idea is to use the opinion words around each product feature in a review sentence to determine the opinion orientation on the product feature. As we discussed earlier, the key difficulties are: (1) how to combine multiple opinion words (which may be conflicting) to arrive at the final decision, (2) how to deal with context or domain dependent opinion words without any prior knowledge from the user, and (3) how to deal with many important language constructs which can change the semantic orientations of opinion words. We propose several novel techniques which make use of the review and sentence context, and general natural language rules to deal with these problems.

4.1. Opinion Words, Phrases and Idioms

Opinion (or sentiment) words and phrases are words and phrases that express positive or negative sentiments. Words that encode a desirable state (e.g., great, awesome) have a positive orientation, while words that represent an undesirable state have a negative orientation (e.g., disappointing). While orientations apply to most adjectives, there are those adjectives that have no orientations

(e.g., external, digital). There are also many words whose semantic orientations depend on contexts in which they appear. For example, the word "long" in the following two sentences has completely different orientations, one positive and one negative:

> "The battery of this camera lasts very long"

> "This program takes a long time to run"

In the proposed method, we will deal with this problem. Although words that express positive or negative orientations are usually adjectives and adverbs, verbs and nouns can be used to express opinions as well, e.g., verbs such as "like" and "hate", and nouns such as "junk" and "rubbish".

Researchers have compiled sets of such words and phrases for adjectives, adverbs, verbs, and nouns respectively. Such lists are collectively called the *opinion lexicon*. Each set is usually obtained through a bootstrapping process [13] using the WordNet. In this work, we used the lists from [13]. However, the lists only have opinion words that are adjectives and adverbs. We added verb and noun lists identified in the same way. We also have lists of *context dependent* opinion words.

In order to make use of the different lists, we need to perform *part-of-speech* (POS) *tagging* as many words can have multiple POS tags depending on their usages. The part-of-speech of a word is a linguistic category that is defined by its syntactic or morphological behavior. Common POS categories in English are: noun, verb, adjective, adverb, pronoun, preposition, conjunction and interjection. In this project, we used the NLProcessor linguistic parser [25] for POS tagging.

Idioms: Apart from opinion words, there are also idioms. We also identified those positive, negative and dependent idioms. In fact, most idioms express strong opinions, e.g., "cost (somebody) an arm and a leg". We annotated more than 1000 idioms. Although this task is time consuming, it is only a one-time effort and the annotated idioms can be used by the community.

Non-opinion phrases containing opinion words: An important issue that needs to be handled is that some phrases have no opinion but contain opinion words, e.g., "pretty large", where "pretty" is a positive opinion word, but the whole phrase has no opinion or has context dependent opinion. Such phrases need to be identified and used to overwrite the opinion words in them.

4.2. Aggregating Opinions for a Feature

Using the final lists of positive, negative and dependent words, and idioms, the system identifies (positive, negative or neutral) opinion orientation expressed on each product feature in a review sentence as follows:

- Given a sentence s that contains a set of features, opinion words in the sentence are identified first. Note that a sentence may express opinions on multiple features. For each feature f in the sentence, we compute an orientation score for the feature. A positive word is assigned the semantic orientation score of $+1$, and a negative word is assigned the semantic orientation score of -1. All the scores are then summed up using the following score function:

$$score(f) = \sum_{w_i: w_i \in s \wedge w_i \in V} \frac{w_i.SO}{dis(w_i, f)}, \qquad (1)$$

where w_i is an opinion word, V is the set of all opinion words

(including idioms) and s is the sentence that contains the feature f, and $dis(w_i, f)$ is the distance between feature f and opinion word w_i in the sentence s. $w_i.SO$ is the semantic orientation of the word w_i. The multiplicative inverse in the formula is used to give low weights to opinion words that are far away from the feature f.

The reason that the new function works better than the simple summation of opinions in [13] is that far away opinion words may not modify the current feature. However, setting a distance range/limit within which the opinion words are considered does not perform well either because in some cases, the opinion words may be far away. The proposed new function deals with both problems nicely.

Note that the feature itself can be an opinion word as it may be an adjective representing a *feature indicator*, e.g., "reliable" in the sentence "This camera is very reliable". In this case, score(f) is +1 or –1 depending on whether f (e.g., "reliable") is positive or negative (in this case, Equation (1) will not be used).

- If the final score is positive, then the opinion on the feature in the sentence s is positive. If the final score is negative, then the opinion on the feature is negative. It is neutral otherwise.

The algorithm is given in Figure 2, where the variable *orientation* in OpinionOrietation holds the total score. Several constructs need special handling, for which a set of linguistic rules is used:

Negation Rules: The negation word or phrase usually reverses the opinion expressed in a sentence. Negation words include traditional words such as "no", "not", and "never", and also pattern-based negations such as "stop" + "vb-ing", "quit" + "vb-ing" and "cease" + "to vb". Here, vb is the POS tag for verb and "vb-ing" is vb in its -ing form. The following rules are applied for negations:

Negation Negative → Positive   //e.g., "no problem"

Negation Positive → Negative   // e.g., "not good"

Negation Neutral → Negative   // e.g., "does not work", where "work" is a neutral verb.

For pattern-based negations, the system detects the patterns and then applies the rules above. For example, the sentence, "the camera stopped working after 3 days", conforms to the pattern "stop"+"vb-ing", and is assigned the negative orientation by applying the last rule as "working" is neutral.

Note that "Negative" and "Positive" above represent negative and positive opinion words respectively.

Non-negation containing negation words: There are also non-negation phrases that contain negation words, e.g., "not" in "I like this camera *not just* because it is beautiful" does not mean negation because of the phrase "not just". Again, such phrases need be identified to overwrite the negation words in them.

"But" Clause Rules: A sentence containing "but" also needs special treatment. The opinion before "but" and after "but" are usually the opposite to each other. Phrases such as "with the exception of", "except that", and "except for" behaves similarly to "but" and are handled in the same way as "but". "but" clauses are handled as follows:

```
1.   Algorithm OpinionOrietation()
2.      for each sentence s_i that contains a set of features do
3.         features = features contained in s_i;

4.         for each feature f_j in features do
5.            orientation = 0;
6.            if feature f_j is in the "but" clause then
7.               orientation = apply the "but" clause rule
8.            else  remove "but" clause from s_i if it exists;
9.               for each unmarked opinion word ow in s_i do
10.                    // ow can be a TOO word or Negation word as well
11.                  orientation += wordOrientation(ow, f_j, s_i);
12.            endfor
13.            endif

14.            if orientation > 0 then
15.               f_j's orientation in s_i = 1
16.            else   if orientation < 0 then
17.                  f_j's orientation in s_i = -1
18.               else
19.                  f_j's orientation in s_i = 0
20.               endif
21.            endif

22.            if f_j is an adjective then
23.               (f_j).orientation += f_j's orientation in s_i;
24.            else   let o_ij is the nearest adjective word to f_j, in s_i;
25.               (f_j, o_ij).orientation += f_j's orientation in s_i;
26.            endif

27.         endfor
28.      endfor;
         // Context dependent opinion words handling
29.      for each f_j with orientation = 0 in sentence s_i do
30.         if f_j is an adjective then
31.            f_j's orientation in s_i = (f_j).orientation
32.         else    // synonym and antonym rule should be applied too
33.            let o_ij is the nearest opinion word to f_j, in s_i;
34.            if (f_j, o_ij) exists then
35.               f_j's orientation in s_i = (f_j, o_ij).orientation
36.            endif
37.         endif
38.         if f_j's orientation in s_i = 0 then
39.            f_j's orientation in s_i = apply inter-sentence
                     conjunction rule
40.         endif
41.      endfor
```

```
1.   Procedure wordOrientation(word, feature, sentence)
2.      if word is a Negation word then
3.         orientation = apply Negation Rules;
4.         mark words in sentence used by Negation rules
5.      elseif word is a TOO word then
6.         orientation = apply TOO Rules;
7.         mark words in sentence used by TOO rules
8.      else
9.         orientation = orientation of word in opinionWord_list
10.     endif
```

11. $orientation = \dfrac{orientation}{dis(word, feature)},$

Figure 2: Predicting the orientations of opinions on product features

If the product feature f_j appears in the "but" clause then
    for each unmarked opinion word *ow* in the "but" clause of
      the sentence s_i do
        // *ow* can be a TOO word (see below) or Negation word
        orientation += **wordOrientation**(*ow*, f_j, s_i);
    endfor
    If *orientation* ≠ 0 then
      return *orientation*
    else *orientation* = orientation of the clause before "but"
      If *orientation* ≠ 0 then
        return −1 * *orientation*
      else   return 0
  endif

The algorithm above basically says that we follow the semantic orientation of the "but" clause first. If we cannot get an orientation there, we will look at the clause before "but" and negate its orientation.

Non-but clauses containing but-like words: Similar to negations and opinion words, a sentence containing "but" does not necessarily change the opinion orientation. For example, "but" in "I not only like the picture quality of this camera, but also its size" does not change opinion after "but" due to the phrase "but also".

4.3. Handling Context Dependent Opinions

We now deal with context dependent opinion words. Three rules (or *linguistic conventions*) are proposed, which use the contextual information in other reviews of the same product, sentences in the same review and even clauses of the same sentence to infer the orientation of the opinion word in question. Since this method makes use of the global information rather than only the local information, we thus call this approach the *holistic* approach.

1. **Intra-sentence conjunction rule**: For example, we have the sentence, "the battery life is very long". It is not clearly whether "long" means a positive or a negative opinion on the product feature "battery life". Our algorithm tries to see whether any other reviewer said that "long" is positive (or negative). For example, another reviewer wrote "This camera takes *great* pictures and has a *long* battery life". From this sentence, we can discover that "long" is *positive* for "battery life" because it is conjoined with the positive opinion word "great". We call this the *intra-sentence conjunction rule*, which means that a sentence only expresses one opinion orientation unless there is a "but" word which changes the direction. The following sentence is unlikely: "This camera takes *great* pictures and has a *short* battery life." It is much more natural to say: "This camera takes *great* pictures, but has a *short* battery life."

2. **Pseudo intra-sentence conjunction rule:** Sometimes, one may not use an explicit conjunction "and". Let us use the example sentence "the battery life is long" again. We have no idea whether "long" is positive or negative for "battery life". A similar strategy can be applied. For instance, another reviewer might have written the following: "The camera has a long battery life, which is great". The sentence indicates that the semantic orientation of "long" for "battery life" is positive due to "great", although no explicit "and" is used.

Using these two rules, we consider two cases:

Adjectives as feature indicators: In this case, an adjective is a feature indicator. For example, "small" is a feature indicator

that indicates feature "size" in the sentence, "this camera is very small". It is not clearly from this sentence whether "small" means positive or negative. The above two rules can be applied to determine the semantic orientation of "small" for "camera".

Explicit features that are not adjectives: In this case, we use opinion words near the feature words to determine the opinion orientations on the feature words. For example, in the sentence "the battery life of this camera is long", "battery life" is the given feature and "long" is a nearby opinion word. Again we can apply the above two rules to find the semantic orientation of "long" for "battery life".

3. **Inter-sentence conjunction rule:** If the above two rules could not decide the opinion orientation, we use the context of previous or next sentence (or clauses) to decide. That is, we extend the intra-sentence conjunction rule to neighboring sentences. The idea is that people usually express the same opinion (positive or negative) across sentences unless there is an indication of opinion change using words such as "but" and "however". For example, the following sentences are natural:

    "The picture quality is amazing. The battery life is long"

However, the following sentences are not natural:

    "The picture quality is amazing. The battery life is short"

It is much more natural to say: "The picture quality is amazing. However, the battery life is short"

Below, we give the algorithm. The variable *orientation* is the opinion score on the current feature. Note that the algorithm only uses neighboring sentences. Neighboring clauses in the same sentence can be used in a similar way too.

  if the previous sentence exists and has an opinion then
    if there is not a "However" or "But" word to change the
      direction of the current sentence, then
      orientation = the orientation of the last clause of the
        previous sentence
    else *orientation* = opposite orientation of the last clause of
      the previous sentence
  elseif the next sentence exists and has an opinion then
    if there is a not "However" or "But" word to change the
      direction of the next sentence, then
      orientation = the orientation of the first clause of the
        next sentence
    else *orientation* = opposite orientation of the last clause
      of the next sentence
  else *orientation* = 0
  endif

For rule 1 and rule 2, we should also note the following. It is possible that in the reviews of a product the same opinion word for the same feature has conflicting orientations. For example, another reviewer may say that "small" is negative for camera size:

    "This camera is very small, which I don't like"

In this case, our algorithm takes the majority view. That is, if more people indicate that "small" is positive for camera size, we will treat it as positive and vice versa. Note that if the above reviewer instead says:

"This camera is too small"

"small" is not given an orientation because "too" here indicates an negative opinion in any case (see the above TOO rules).

Synonym and Antonym Rule: If a word is found to be positive (or negative) in a context for a feature, its synonyms are also considered positive (or negative), and its antonyms are considered negative (or positive). For example, in the above sentence, we know that "long" is positive for "battery life". Then we also know that "short" is negative for "battery life".

The whole algorithm is given in Figure 2. Lines 22 – 26 and lines 29 – 41 need some additional explanation. Lines 29 – 41 deal with product features that the first iteration (lines 2 – 28) did not identify their opinion orientations because there were no opinion words or the opinion words have context dependent orientations. Thus, lines 29 – 41 basically use the three strategies above to handle the context dependent (or undecided) cases. Line 30 says that if the feature f_j is an adjective (i.e., a feature indicator), then its orientation simply takes the majority orientation in other reviews (line 31). If the feature f_j is not a feature indicator, we find the nearest opinion word o_{ij} and use the dominant orientation in other reviews on the pair (f_j, o_{ij}) (line 35), which is stored in (f_j, o_{ij}).*orientation* and is computed in line 25 (see below). If (f_j, o_{ij}) does not exist, we will see if o_{ij}'s synonym or antonym exists in the (f, o) pair list. If it exists, we apply the synonym and antonym rule. If we still cannot find a match in the (f, o) list, the orientation of feature f_j remains to be neutral. Note that the application of the synonym and antonym rule is not included in the algorithm in Figure 2 just to simply it, but can be added easily.

Lines 22 – 26 record those opinions identified in other sentences or reviews, which are used in lines 29 – 41. Line 22 states that if feature f_j is an adjective (i.e., a feature indicator), we aggregate its orientations in different reviews (line 23). If the feature f_j is not a feature indicator (line 24), we find the nearest opinion word o_{ij} (line 24) and again sum up its orientation in different reviews (line 25). The orientation is stored in (f_j, o_{ij}).*orientation*. A pair is used because we want to ensure that the opinion word o_{ij} is for the specific feature f_j since an opinion word can modify multiple features with different orientations.

5. EMPIRICAL EVALUATION

A system, called Opinion Observer, based on the proposed technique has been implemented in C++. This section evaluates Opinion Observer to assess its accuracy for predicting the semantic orientations of opinions on product features.

We carried out the experiments using customer reviews of 8 products: two digital cameras, one DVD player, one MP3 player, two cellular phones, one router and one anti-virus software. The characteristics of each review data set are given in Table 1. The reviews of the first five products are the benchmark data set from [13] (http://www.cs.uic.edu/~liub/FBS/FBS.html). The reviews of the last three products are annotated by us following the same scheme as that in [13]. All the reviews are from amazon.com.

An issue in judging opinions in reviews is that the decisions can be 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 for some fuzzy cases can be difficult. For the difficult sentences, a consensus was reached between the primary human annotators (the first author of the paper and two MS students) and the secondary annotator (the second author of the paper).

Note that the features here are considerably more than those used in [13] because [13] only considers explicit noun features. Here, we include both explicit and implicit features of all POS tags. There are a large number of features that are verbs and adjectives, which often indicate implicit features. Duplicate features that appear in different sentences or reviews are also counted to reflect opinions from different reviewers on the same feature. Note also that there are many features that are synonyms. Identifying synonyms is a challenging problem and is beyond the scope of this paper (we plan to address it in our future work).

Table 1: Characteristics of the review data

	Product name	No. of reviews	No. of features
1	Digital camera 1	45	263
2	Digital camera 2	34	191
3	Cellular phone 1	49	374
4	MP3 player	95	720
5	DVD player	99	356
6	Cellular phone 2	41	306
7	Router	31	227
8	Anti-virus software	51	179
	Total	**445**	**2616**

The NLProcessor system [25] is used to generate POS tags. After POS tagging, our system Opinion Observer is applied to find orientations of opinions expressed on product features.

Table 2 gives the experimental results. The performances are measured using the standard evaluation measures of precision (p), recall (r) and F-score (F), $F = 2pr/(p+r)$.

In this table, we compare three techniques: (1) the proposed new technique used in Opinion Observer, (2) the proposed technique without handling context dependency of opinion words, (3) the existing technique FBS in [13]. In Table 3, we will also compare with the Opine system in [28], which improved FBS.

From Table 2, we observe that the new algorithm Opinion Observer has a much higher F-score than the existing FBS method. The main loss of FBS is in the recall. The precision is slight higher because it is only able to find those obvious cases. The new method in Opinion Observer is able to improve the recall dramatically with almost no loss in precision.

Note that the original FBS system [13] only deals with explicit noun features. It has been extended to consider all types of features. The results of FBS reported here are from the improved system. It still uses the same technique as that in [13], but with the same number of features as Opinion Observer.

We also observe from Table 2 that handling context dependent opinion words help significantly. Without it (**Opinion Observer – No context dependency handling**), the average F-score dropped to 87% (Column 7) due to poor recall (Column 6) because many features are assigned the neutral orientation.

Similarly, we can see that the score function of Equation (1) is highly influential as well. Using the simple summation of semantic orientations without considering the distance between opinion words and product features as in FBS produces a worse average F-score (0.87 in Column 10) (**Opinion Observer – Without using Equation (1)**). Thus, we conclude that both the

Table 2: Results of opinion sentence extraction and sentence orientation prediction

Product name	Opinion Observer			Opinion Observer – No context dependency handling			Opinion Observer – Without using Equation (1)			FBS		
	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
Digital camera 1	0.93	0.92	**0.93**	0.95	0.88	**0.91**	0.91	0.88	**0.89**	0.93	0.80	**0.86**
Digital camera 2	0.96	0.96	**0.96**	0.97	0.92	**0.95**	0.97	0.92	**0.95**	0.98	0.87	**0.92**
Cellular phone 1	0.93	0.90	**0.91**	0.93	0.79	**0.86**	0.91	0.86	**0.88**	0.94	0.70	**0.80**
MP3 player	0.87	0.86	**0.87**	0.89	0.79	**0.83**	0.86	0.82	**0.84**	0.91	0.69	**0.78**
DVD player	0.89	0.88	**0.89**	0.92	0.83	**0.87**	0.9	0.85	**0.87**	0.91	0.72	**0.80**
Cellular phone 2	0.95	0.95	**0.95**	0.95	0.89	**0.92**	0.95	0.92	**0.93**	0.95	0.82	**0.88**
Router	0.84	0.82	**0.83**	0.84	0.78	**0.81**	0.82	0.78	**0.80**	0.83	0.67	**0.74**
Antivirus software	0.90	0.87	**0.88**	0.93	0.72	**0.81**	0.89	0.81	**0.85**	0.94	0.64	**0.76**
Average	0.91	0.90	**0.90**	0.92	0.83	**0.87**	0.90	0.85	**0.87**	0.92	0.74	**0.82**

score function and the handling of context dependent opinion words are very useful.

Table 3 compares the results of the Opine system reported in [28] based on the same benchmark data set (reviews of the first 5 products in Table 1). It was shown in [28] that Opine outperforms FBS. Here, we are only able to compare the average results as individual results for each product are not reported in [28]. It can be observed that Opinion Observer outperforms Opine on both precision and recall. Furthermore, the new algorithm is much simpler than the relaxation labeling method used in [28]. In the table, we also include the results of the FBS method on the reviews of the first 5 products. Again, Opinion Observer is dramatically better in recall and F-score with almost no loss in precision.

In summary, we conclude that the new technique is highly effective and is markedly better than the existing methods.

Table 3: Comparison of FBS, OPINE and Opinion Observer based on the benchmark data set in [13], which consists of all reviews of the first 5 products in Table 2.

	Precision	Recall	F-Score
FBS	0.93	0.76	0.83
OPINE	0.86	0.89	0.87
Opinion Observer	0.92	0.91	0.91

6. CONCLUSION

This paper proposed an effective method for identifying semantic orientations of opinions expressed by reviewers on product features. It is able to deal with two major problems with the existing methods, (1) opinion words whose semantic orientations are context dependent, and (2) aggregating multiple opinion words in the same sentence. For (1), a holistic approach is proposed that can accurately infer the semantic orientation of an opinion word based on the review context. For (2), a new function to combine multiple opinion words in the same sentence is proposed. Furthermore, previous research only considers explicit opinions expressed by adjectives and adverbs. In this work, both explicit and implicit opinions are considered. Our method also handles implicit features represented by feature indicators. These make the proposed technique more complete. Experimental results show that the proposed technique performs markedly better than the state-of-the-art existing methods.

7. REFERENCES

[1]. A. Andreevskaia and S. Bergler. Mining WordNet for Fuzzy Sentiment: Sentiment Tag Extraction from WordNet Glosses. In *EACL'06*, pp. 209–216, 2006.

[2]. P. Beineke, T. Hastie, C. Manning, and S. Vaithyanathan. An Exploration of Sentiment Summarization. In *Proc. of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, 2003.

[3]. G. Carenini, R. Ng, and A. Pauls. Interactive Multimedia Summaries of Evaluative Text. *IUI'06*, 2006.

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

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

[6]. X. Ding and B. Liu. The Utility of Linguistic Rules in Opinion Mining." *SIGIR-2007* (poster paper).

[7]. A. Esuli and F. Sebastiani, EACL-06, 2006. Determining Term Subjectivity and Term Orientation for Opinion Mining, EACL-06, 2006.

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

[9]. M. Gamon, A. Aue, S. Corston-Oliver, and E. K. Ringger. Pulse: Mining customer opinions from free text. *IDA'2005*.

[10]. V. Hatzivassiloglou and J. Wiebe. Effects of adjective orientation and gradability on sentence subjectivity. *COLING'00*, 2000.

[11]. V. Hatzivassiloglou and K. McKeown. Predicting the Semantic Orientation of Adjectives. *ACL-EACL'97*, 1997.

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

[13]. M. Hu and B. Liu. Mining and summarizing customer reviews. *KDD'04*, 2004.

[14]. N. Jindal, and B. Liu. Mining Comparative Sentences and Relations. In *AAAI'06*, 2006.

[15]. N. Kaji and M. Kitsuregawa. Automatic Construction of Polarity-Tagged Corpus from HTML Documents. *COLING/ACL'06*, 2006.

[16]. H. Kanayama and T. Nasukawa. Fully Automatic Lexicon Expansion for Domain-Oriented Sentiment Analysis. *EMNLP'06*, 2006.

[17]. S. Kim and E. Hovy. Determining the Sentiment of Opinions. *COLING'04*, 2004.

[18]. S. Kim and E. Hovy. Automatic Identification of Pro and Con Reasons in Online Reviews. *COLING/ACL 2006*.

[19]. N. Kobayashi, R. Iida, K. Inui and Y. Matsumoto. Opinion Mining on the Web by Extracting Subject-Attribute-Value Relations. In *Proc. of AAAI-CAAW'06*, 2006.

[20]. L.-W. Ku, Y.-T. Liang and H.-H. Chen. Opinion Extraction, Summarization and Tracking in News and Blog Corpora. In *Proc. of the AAAI-CAAW'06*, 2006.

[21]. B. Liu, M. Hu, M. and J. Cheng. Opinion Observer: Analyzing and comparing opinions on the Web. *WWW-05*, 2005.

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

[23]. T. Nasukawa and J. Yi. Sentiment analysis: Capturing favorability using natural language processing. *K-CA*-2003.

[24]. V. Ng, S. Dasgupta and S. M. Niaz Arifin. Examining the Role of Linguistic Knowledge Sources in the Automatic Identification and Classification of Reviews. *ACL'06*, 2006.

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

[26]. B. Pang and L. Lee, Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. *ACL'05*, 2005.

[27]. B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment Classification Using Machine Learning Techniques. *EMNLP'2002*, 2002.

[28]. A-M. Popescu and O. Etzioni. Extracting Product Features and Opinions from Reviews. *EMNLP-05*, 2005.

[29]. E. Riloff and J. Wiebe. 2003. Learning extraction patterns for subjective expressions. *EMNLP'2003*, 2003.

[30]. V. Stoyanov and C. Cardie. Toward opinion summarization: Linking the sources. In *Proc. of the Workshop on Sentiment and Subjectivity in Text*, 2006.

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

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

[33]. T. Wilson, J. Wiebe, and R. Hwa. Just how mad are you? Finding strong and weak opinion clauses. *AAAI'04*, 2004.

[34]. J. Wiebe, and R. Mihalcea. Word Sense and Subjectivity. In *ACL'06*, 2006.

[35]. J. Wiebe, and E. Riloff: Creating Subjective and Objective sentence classifiers from unannotated texts. *CICLing*, 2005.

[36]. H. Yu, V. Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. *EMNLP'2003*.

[37]. L. Zhuang, F. Jing, X.-Yan Zhu, and L. Zhang. Movie Review Mining and Summarization. *CIKM-06*, 2006.