Opinion Observer: Analyzing and Comparing Opinions on the Web

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

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

Junsheng Cheng
Department of Computer Science
University of Illinois at Chicago
851 South Morgan Street,
Chicago, IL 60607-7053
Jcheng1@cs.uic.edu

ABSTRACT

The Web has become an excellent source for gathering consumer opinions. There are now numerous Web sites containing such opinions, e.g., customer reviews of products, forums, discussion groups, and blogs. This paper focuses on online customer reviews of products. It makes two contributions. First, it proposes a novel framework for analyzing and comparing consumer opinions of competing products. A prototype system called *Opinion Observer* is also implemented. The system is such that with a single glance of its visualization, the user is able to clearly see the strengths and weaknesses of each product in the minds of consumers in terms of various product features. This comparison is useful to both potential customers and product manufacturers. For a potential customer, he/she can see a visual side-by-side and feature-by-feature comparison of consumer opinions on these products, which helps him/her to decide which product to buy. For a product manufacturer, the comparison enables it to easily gather marketing intelligence and product benchmarking information. Second, a new technique based on language pattern mining is proposed to extract product features from Pros and Cons in a particular type of reviews. Such features form the basis for the above comparison. Experimental results show that the technique is highly effective and outperform existing methods significantly.

Categories and Subject Descriptors

I.2.7 [Natural Language Processing]: Text analysis.

General Terms: Algorithms, Human Factors.

Keywords: Opinion analysis, sentiment analysis, information extraction, visualization.

1. INTRODUCTION

The Web has dramatically changed the way that consumers express their opinions. They can now post reviews of products at merchant sites and express their views on almost anything in Internet forums, discussion groups, and blogs. This online word-of-mouth behavior represents new and measurable sources of information for marketing intelligence. Techniques are now being developed to exploit these sources to help companies and individuals to gain such information effectively and easily.

This paper focuses on online customer reviews of products. It is a common practice for online merchants (e.g., amazon.com) to ask their customers to review the products that they have purchased. There are also dedicated review sites, e.g., epininons.com. With more and more people using the Web to express opinions, the number of reviews that a product receives grows rapidly. For some popular products, the number of reviews can be in hundreds or more. These reviews provide excellent sources of consumer opinions on products, which are very useful to both potential customers and product manufacturers.

In this paper, we propose an analysis system with a visual component to compare consumer opinions of different products. The system is called *Opinion Observer*. With a single glance of its visualization, the user can clearly see the strengths and weaknesses of each product in the minds of consumers. We use Figure 1 to illustrate the idea. It compares customer opinions of two digital cameras along different feature dimensions, i.e., picture, battery, zoom, size, and weight.

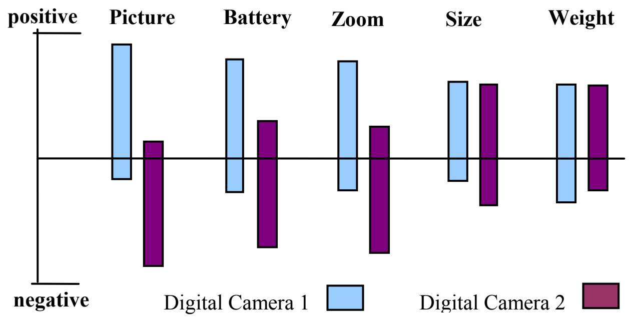


Figure 1: Visual comparison of consumer opinions on two products.

Each bar in Figure 1 shows the percents of reviews that express positive (above x-axis) and negative (below x-axis) opinions on a feature of a camera. One can easily see that digital camera 1 is a superior camera. Specifically, most customers have negative opinions about the picture quality, battery and zoom of digital camera 2. However, on the same three features, customers are mostly positive about digital camera 1. Regarding size and weight, customers have similar opinions on both cameras. The visualization enables the user to clearly see how the cameras compare with each other along each feature dimension.

This opinion comparison is useful to both potential customers (buyers) and product manufacturers.

- For a potential customer, although he/she can read all reviews of different products at merchant sites to mentally compare and assess the strengths and weaknesses of each product in order to decide which one to buy, it is much more convenient and less time consuming to see a visual feature-by-feature comparison of customer opinions in the reviews.

A system like ours can be installed at a merchant site that has reviews so that potential buyers can compare not only prices and product specifications (which can already be done at some sites), but also opinions from existing customers.

- For a product manufacturer, comparing consumer opinions of its products and those of its competitors to find their strengths and weaknesses is crucial for marketing intelligence and for product benchmarking. This is typically done manually now, which is very labor intensive and time consuming. Our system comes to help naturally in this case.

To enable the above visualization, two challenging technical tasks need to be performed:

1. Identifying product features that customers have expressed their (positive or negative) opinions on.
2. For each feature, identifying whether the opinion from each reviewer is positive or negative, if any. Negative opinions typically represent complains/problems about some features.

There are three main review formats on the Web. Different review formats may need different techniques to perform the above tasks.

Format (1) - Pros and Cons: The reviewer is asked to describe Pros and Cons separately. C|net.com uses this format.

Format (2) - Pros, Cons and detailed review: The reviewer is asked to describe Pros and Cons separately and also write a detailed review. Epinions.com uses this format.

Format (3) - free format: The reviewer can write freely, i.e., no separation of Pros and Cons. Amazon.com uses this format.

For formats (1) and (2), opinion orientations (positive or negative) of features are known because Pros and Cons are separated and thus there is no need to identify them. Only product features that have been commented on by customers need to be identified. For format (3), we need to identify both product features and opinion orientations. In [17], we proposed several techniques to perform these tasks for format (3), which are also useful for format (1). In both formats (1) and (3), reviewers typically use full sentences. However, for format (2), Pros and Cons tend to be very brief. For example, under Cons, one may write: "heavy, bad picture quality, battery life too short", which are elaborated in the detailed review.

In this paper, we propose a new technique to identify product features from Pros and Cons in format (2). The method is based on natural language processing and supervised pattern discovery. We show that the techniques in [17] are not suitable for format (2) because of short phrases or incomplete sentences (we call them *sentence segments*) in Pros and Cons rather than full sentences. We do not analyze detailed reviews of format (2) as they are elaborations of Pros and Cons. Analyzing short sentence segments in Pros and Cons produce more accurate results. Note that our visualization system is applicable to all three formats.

Our work is related but quite different from sentiment classification [e.g., 8, 9, 15, 29, 30, 32, 33, 35]. Its purpose is to classify reviews as positive or negative. It does not identify product features that have been commented on by consumers. We will discuss this and other related work in Section 2.

Given a set of products (which may be from the same brand or different brands) and a set of URLs of Web pages that contain customer reviews, Opinion Observer works in two stages:

Stage 1: Extracting and analyzing customer reviews in two steps:

Step 1: This step automatically connects to and downloads all customer reviews from the given pages. Subsequently, the system monitors these pages to periodically download new reviews if any. All raw reviews are stored in a database.

Note that this step is not needed if an online merchant or a dedicated review site that has reviews wants to provide the opinion comparison service.

Step 2: In this step, all the new reviews (which were not analyzed before) of every product are analyzed. Two tasks are performed, identifying product features and opinion orientations from each review. This can be done automatically or semi-automatically. Details of this step will be discussed in Sections 3.3 and 3.4.

Stage 2: In this stage, based on the analysis results, different users can visualize and compare opinions of different products using a user interface. The user simply chooses the products that he/she wishes to compare and the system then retrieves the analyzed results of these products and display them in the interface (see Section 3.2). Note that Stage 1 tasks are performed by the system or together with human analysts. Stage 2 is for anyone who is authorized to view the results.

This paper makes the following contributions:

1. To the best of our knowledge, Opinion Observer is the first system that allows comparison of consumer opinions of multiple (competing) products (it can be one). The system is useful to both potential customers and product manufacturers.
2. A new technique is proposed to identify product features from Pros and Cons of review format (2). Existing techniques in [17] are not suitable for this case. Our experimental results show that the proposed technique is highly effective.

2. RELATED WORK

Gathering and comparing consumer opinions of competing products from the Web for marketing intelligence and for product benchmarking is an important problem. To our knowledge, no existing system is able to perform visual comparison of consumer opinions as proposed in this paper. Below, we mainly discuss prior work related to analysis of customer reviews or opinions.

In [17], we propose several methods to analyze customer reviews of format (3). They perform the same tasks of identifying product features on which customers have expressed their opinions and determining whether the opinions are positive or negative. However, the techniques in [17], which are primarily based on unsupervised itemset mining, are only suitable for reviews of formats (3) and (1). Reviews of these formats usually consist of full sentences. The techniques are not suitable for Pros and Cons of format (2), which are very brief. Instead, we use supervised rule mining in this work to generate language patterns to identify product features. This new method is much more effective than the old methods (see Section 5). Currently we do not use detailed reviews of format (2). Although the methods in [17] can be applied to detailed reviews of format (2), analyzing short sentence segments in Pros and Cons produce more accurate results.

In [23], Morinaga *et al.* compare information of different products in a category through search to find the reputation of the products. It does not analyze reviews, and does not identify product features. Below, we present some other related research.

Terminology finding and entity extraction

There are basically two techniques for terminology finding: symbolic approaches that rely on noun phrases, and statistical approaches that exploit the fact that words composing a term tend to be found close to each other and reoccurring [e.g., 4, 7, 18, 19]. However, using noun phrases tends to produce too many non-terms, while using reoccurring phrases misses many low frequency terms, terms with variations, and terms with only one word. As shown in [17] using the existing terminology finding system FASTR [11] produces very poor results. Furthermore, using noun phrases are not sufficient for finding product features. We also need to consider other language components (e.g., verbs and adjectives) as we will see in Section 3.3.

Recently, information extraction from texts was studied by several researchers. Their focus is on using machine learning and NLP methods to extract/classify named entities and relations [5, 10, 14, 20, 31]. Our task involves identifying product features which are usually not named entities and can be expressed as nouns, noun phrases, verbs, and adjectives. Also, our extraction work uses short sentence segments rather than full sentences.

Sentiment classification

Sentiment classification classifies opinion texts or sentences as positive or negative. Work of Hearst [16] on classification of entire documents uses models inspired by cognitive linguistics. Das and Chen [8] use a manually crafted lexicon in conjunction with several scoring methods to classify stock postings. Tong [32] generates sentiment (positive and negative) timelines by tracking online discussions about movies over time.

[33] applies a unsupervised learning technique based on mutual information between document phrases and the words "excellent" and "poor" to find indicative words of opinions for classification. [29] examines several supervised machine learning methods for sentiment classification of movie reviews. [9] also experiments a number of learning methods for review classification. They show that the classifiers perform well on whole reviews, but poorly on sentences because a sentence contains much less information. [1] finds that supervised sentiment classification is inaccurate. They proposed a method based on social network for the purpose. However, social networks are not applicable to customer reviews.

[15] investigates sentence subjectivity classification. A method is proposed to find adjectives that are indicative of positive or negative opinions. [34] proposes a similar method for nouns. Other related works on sentiment classification and opinions discovery include [26, 27, 30, 35, 36].

Our work differs from sentiment and subjectivity classification as they do not identify features commented by customers or what customers praise or complain about. Thus, we solve a related but different problem. They also do not perform opinion comparisons.

3. OPINION OBSERVER

We now present the proposed system, Opinion Observer. We first describe the problem statement, and then introduce the main visualization interface of Opinion Observer for comparing consumer opinions of different products. After that, we discuss the new automatic technique for identifying product features from Pros and Cons in reviews of format (2), which is followed by the interactive method with a convenient user interface. Finally, we discuss how to automatically collect customer reviews from Web pages using a Web data record extraction technique.

3.1 Problem Statement

Let $P = \{P_1, P_2, \ldots, P_n\}$ be a set of products (which may be from the same brand or different brands) that the user is interested in. Each product P_i has a set of reviews $R_i = \{r_1, r_2, \ldots, r_k\}$. Each review r_j is a sequence of sentences $r_j = <s_{j1}, s_{j2}, \ldots, s_{jm}>$ (this is a simplification as Pros and Cons in a review may be separated). The reviews may be from one site or multiple sites as more than one site may have reviews of a particular product.

Definition (product feature): A *product feature* f in r_j is an attribute/component of the product that has been commented on in r_j. If f appears in r_j, it is called an *explicit feature* in r_j. If f does not appear in r_j but is implied, it is called an *implicit feature* in r_j.

In a similar way, we can define an explicit feature and an implicit feature in a sentence. For example, "battery life" in the following two opinion sentences/segments is an explicit feature:

"The battery life of this camera is too short"

"Battery life too short"

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

"This camera is too large"

"Too big"

Definition (opinion segment of a feature): The *opinion segment Os* of feature f in review r_i is a set of consecutive sentences that expresses a positive or negative opinion on f.

We note that it is common that a sequence of sentences (at least one) in a review together express an opinion on a feature. Also, one sentence may be used to express opinions of more than one feature as the following two sentences show:

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

"Good picture, long battery life"

Definition (positive opinion set of a feature): The *positive opinion set* (denoted by *Pset*) of feature f of product P_i is the set of opinion segments of f that expresses positive opinions about f from all the reviews R_i of product P_i.

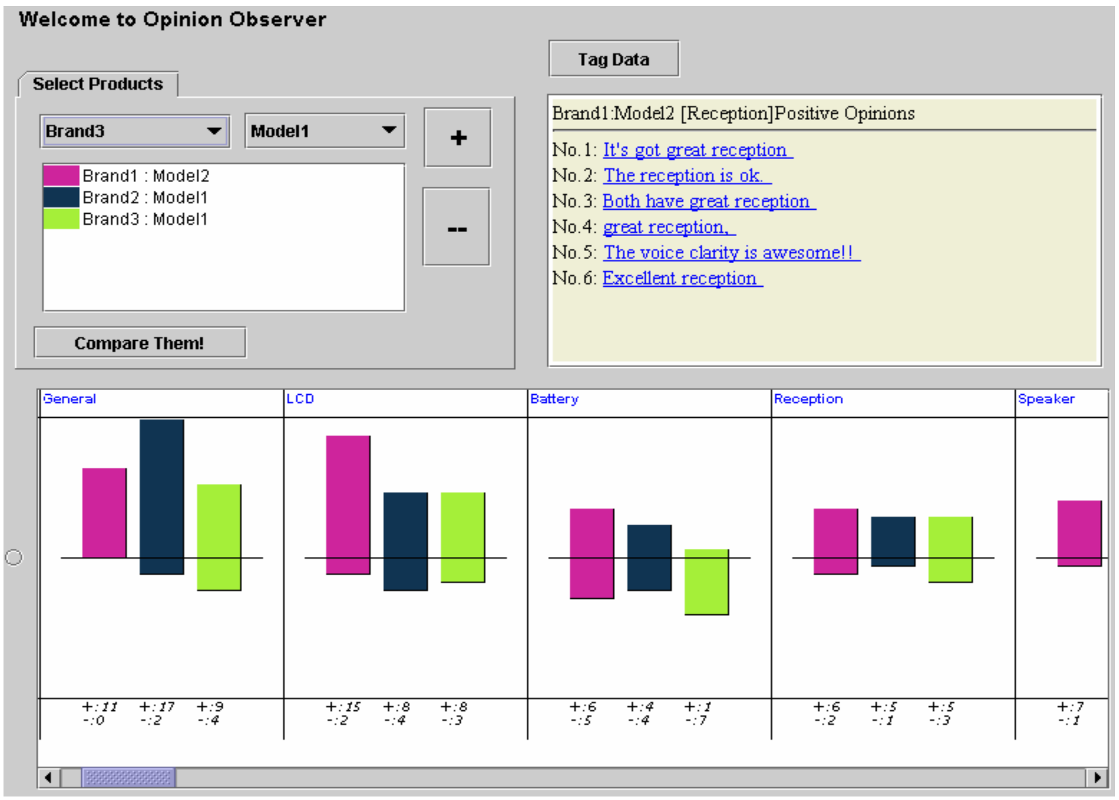
We can define the negative opinion set (*Nset*) in the same way.

Our task: In order to visually compare consumer opinions on a set of products, we need to analyze the reviews in R_i of each product P_i (1) to find all the explicit and implicit product features on which reviewers have expressed their (positive or negative) opinions, and (2) to produce the positive opinion set and the negative opinion set for each feature.

It should be noted that reviews can be analyzed and visualized at different levels of detail. For example, in analyzing the reviews of a digital camera, at the highest level (level 1) we can aggregate *Pset*s and *Nset*s of all features of the camera to show an overall customer opinion on the product. At level 2, we can focus on each main feature or component of the product, e.g., "battery", "zoom" and "picture", and generate its *Pset* and *Nset*. In visualization, we simply use the size of *Pset* or *Nset* of each feature to show the number of positive or negative opinions on the feature (see Figure 1). At level 3, we can study specific problems of each feature, e.g., "the picture is blurry" and "the picture is dark". At the moment, our system aims to work at level 1 and level 2, which are often sufficient. Details at level 3 and beyond are too specific and are studied by human analysts.

3.2 Visualizing Opinion Comparison

We now discuss visualization of opinion comparison. We assume that every product feature and its positive and negative opinion



Brand1:Model2 [Reception]Positive Opinions

No.1: It's got great reception
No.2: The reception is ok.
No.3: Both have great reception
No.4: great reception,
No.5: The voice clarity is awesome!!
No.6: Excellent reception

Figure 2: Opinion Observer's main comparison screen.

sets (*Pset* and *Nset*) have been generated (see Sections 3.3 and 3.4) for a set of products *P*. The main visualization screen is shown in Figure 2, which compares opinions on three cell phones from three different brands. Due to confidentiality, we do not show the actual brand and model of each product.

In the interface, products in *P* are organized into brands and models, which are below "Select Products". To start comparison, the user first selects a few products that he/she wants to compare. To select a product, he/she chooses a brand first and then a model using the drop-down lists. He/she then clicks on "+" to add the selected product model to the box below. The user can also delete a product from comparison by marking the product in the box and clicking on "--". In Figure 2, we see that three products are selected for comparison. To compare, the user clicks on the button "Compare Them!". A bar chart will appear below, which is similar to that in Figure 1. The bars above the x-axis in the middle show positive opinions and the bars below x-axis show negative opinions. The opinion bars of each product are shown in a different color. We clearly see how consumers view different features of each product. If the user is interested in the positive or negative review sentences of a particular feature of a product, he/she can click on the corresponding positive or negative portion of the bar. All the review sentences will then be retrieved and displayed in the box above. The user can click on each sentence to see the entire review from which the sentence is extracted.

Let the set of products selected for a particular comparison be *S* ($\subseteq P$). The set of features used in the visualization is the union of features of all the products in *S*. Each bar above or below x-axis can be displayed in two scales:

1. Actual number of positive or negative opinions (the size of *Pset* or *Nset*) normalized with the maximal number of opinions on any feature of any product. This is to ensure that the tallest bar fits the limited space. The height of the bar representing the size of *Pset* or *Nset* of a feature *j* of product *i*, denoted by $L^+_{i,j}$ (or $L^-_{i,j}$), is computed with

$$L^+_{i,j} = \frac{N^+_{i,j}}{\max(M^+, M^-)}, \qquad L^-_{i,j} = \frac{N^-_{i,j}}{\max(M^+, M^-)}$$

$$M^+ = \max(M^+_1, M^+_2, ..., M^+_k), \quad M^- = \max(M^-_1, M^-_2, ..., M^-_k)$$

$$M^+_i = \max(N^+_{i,1}, N^+_{i,2}, ..., N^+_{i,n}), \quad M^-_i = \max(N^-_{i,1}, N^-_{i,2}, ..., N^-_{i,n})$$

where $N^+_{i,j}$ (or $N^-_{i,j}$) is the size of *Pset* (or *Nset*) of feature *j* of product *i*. M^+ (or M^-) is the maximal size of all *Pset*s (or *Nset*s) of all features of the products in *S*. M_i^+ (or M_i^-) is the maximal size of *Pset*s (or *Nset*s) of all features of product *i*. In the display, $max(M^+, M^-)$ is made equal to the height of space below and above x-axis. In Figure 2, the actual number of positive or negative reviews is also listed below each bar.

2. Percent of positive or negative opinions. We can also show the comparison in term of percentages of positive and negative reviews. A similar method as above can be used to produce a suitable visualization.

To support the visualization, we need to identify product features and opinions on them, which is the topic of Sections 3.3 and 3.4.

3.3 Automated Opinion Analysis

As discussed in the introduction, there are three common review formats. For formats (3) and (1), our existing methods in [17] can be used to extract product features and decide opinion orientations (positive or negative). Note that due to the separation of Pros and Cons, there is no need to decide opinion orientations for reviews of format (1). In this section, we focus on reviews of format (2).

Figure 3 shows a review of format (2). Pros and Cons are separated and very brief. We propose a supervised pattern mining method to find language patterns to identify product features from Pros and Cons. We do not need to determine opinion orientations as they are already indicated by "Pros" and "Cons" (we do not analyze full reviews, which elaborate on Pros and Cons).

Our approach is based on the following important observation:

Each sentence segment contains at most one product feature. Sentence segments are separated by ',', '.', 'and', and 'but'.

For example, Pros in Figure 3 can be separated into 5 segments.

great photos	<photo>
easy to use	<use>
good manual	<manual>
many options	<option>
takes videos	<video>

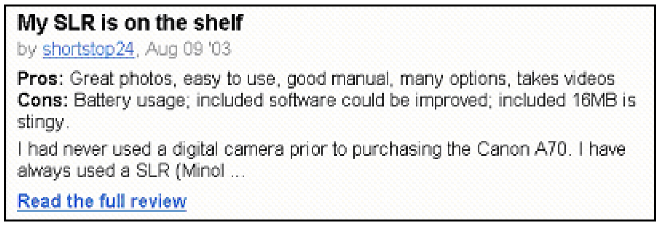
Cons in Figure 3 can be separated into 3 segments:

battery usage	<battery>
included software could be improved	<software>
included 16MB is stingy	<16MB> $\Rightarrow$ <memory>

We can see that each segment describes a product feature on which the reviewer has expressed an opinion (the last two can be seen as full sentences). The product feature for each segment is listed within <>. From the list of features, we note the following:

1. Explicit features and implicit features: Some features are genuine features, i.e., <photo>, <use>, <manual>, <option>, <video>, <battery>, and <software>. We call them explicit features as they appear in sentence segments. However, <16MB> is a value of feature <memory>, which we call an implicit feature as it does not appear in the sentence segment. We need to identify both types of product features.
2. Synonyms: Different reviewers may use different words to mean the same produce feature. For example, one reviewer may use "photo", but another may use "picture". Synonym of features should be grouped together.
3. Granularity of features: In sentence segment "great photos", it is easy to decide that "photo" is the feature. However, in "battery usage", we can use either "battery usage" or "battery" as the feature. As indicated in Section 3.1, we do not use "battery usage" as it is too specific and can fragment the comparison. For example, other reviewers may complain "battery size", "battery weight", "battery color", etc. This results in a large number of features and each feature is only commented on by a few customers. Then, visualization becomes ineffective. Note that in semi-automatic tagging, more detailed analysis is possible (see Sections 3.4).

Another important point to note is that a feature may not be a noun or noun phrase, which is used in [17]. Verbs may be features



My SLR is on the shelf
by shortstop24, Aug 09 '03

Pros: Great photos, easy to use, good manual, many options, takes videos
Cons: Battery usage; included software could be improved; included 16MB is stingy.

I had never used a digital camera prior to purchasing the Canon A70. I have always used a SLR (Minol ...

Read the full review

Figure 3: An example review of format (2).

as well, e.g., "use" in "easy to use". Of course, we can also use its corresponding noun as the feature, e.g., "usage" or simply "use"

3.3.1 Extracting Product Features

We use supervised rule discovery to perform this task. We first prepare a training dataset by manually labeling (or tagging) a large number of reviews. The steps are as follows:

1. Perform Part-Of-Speech (POS) tagging and remove digits: We use the NLProcessor linguistic parser [28] to generate the POS tag of each word (whether the word is a noun, verb, adjective, etc). POS tagging is important as it allows us to generate general language patterns.

   We also remove digits in sentences, e.g., changing "16MB" to "MB". Digits often represent concepts that are too specific to be used in rule discovery, which aims to generalize. We use two examples from above to illustrate the results of this step:

   "<N> Battery <N> usage"
   "<V> included <N> MB <V>is <Adj> stingy"

   <N> indicates a noun, <V> a verb, and <Adj> an adjective. Each POS tag appears right before the corresponding word(s).

2. Replace the actual feature words in a sentence with [feature]: This replacement is necessary because different products have different features. The replacement ensures that we can find general language patterns which can be used for any product. After replacement, the above two examples become:

   "<N> [feature] <N> usage"
   "<V> included <N> [feature] <V> is <Adj> stingy"

   For implicit features, we replace the words that indicate such features with [feature]. For example, "MB" above is replaced with [feature] as it indicates implicit feature <memory>.

   It is possible that some feature contains more than one word, e.g., "auto mode stinks", which will be changed to

   "<NP> [feature] <V> stinks"   // <NP>: noun phrase

3. Use n-gram to produce shorter segments from long ones: For example, "<V> included <N> [feature] <V> is <Adj> stingy" will generate 2 smaller segments:

   "<V> included <N> [feature] <V> is"
   "<N> [feature] <V> is <Adj> stingy"

   We only use 3-grams (3 words with their POS tags) here, which works well. The reason for using n-gram rather than full sentences is because most product features can be found based on local information and POS tagging. Using long sentences tend to generate a large number of spurious rules.

4. Distinguish duplicate tags: When there are duplicate tags in a segment, we distinguish them with a sequence number, e.g.:

   "<N1> [feature] <N2> usage"

5. Perform word stemming: This is commonly performed in information retrieval tasks to reduce a word to its stem.

After the five-step pre-processing and labeling (tagging), the resulting sentence (3-gram) segments are saved in a file (called a *transaction file*) for the generation of rules. In this file, each line contains one processed (labeled) sentence segment. We then use association rule mining [2] to find all rules.

Rule generation: Association rule mining is one of the main data mining models. It is commonly stated as follows: Let $I = \{i_1, \ldots, i_n\}$ be a set of items, and D be a set of transactions. Each transaction consists of a subset of items in *I*. An *association rule* is an implication of the form $X \rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$. The rule $X \rightarrow Y$ holds in *D* with confidence *c* if *c*% of transactions in *D* that support *X* also support *Y*. The rule has support *s* in *D* if *s*% of transactions in *D* contain $X \cup Y$. The problem of mining association rules is to generate all association rules in *D* that have support and confidence greater than the user-specified minimum support and minimum confidence.

We use the association mining system CBA [21] to mine rules. We use 1% as the minimum support, but do not use minimum confidence here, which will be used later. Some example rules are given below (we omit supports and confidences of the rules):

(a) <N1>, <N2> → [feature]
(b) <V>, easy, to → [feature]
(c) <N1> → [feature], <N2>
(d) <N1>, [feature] → <N2>

We observe that both POS tags and words may appear in rules. Note that although association rule mining is commonly applied as an unsupervised method, here we use it in a supervised case because features are manually tagged or labeled.

Post-processing: Not all generated rules are useful. Some post-processing is needed due to a few reasons:

1. We only need rules that have [feature] on the right-hand-side of "→" as our objective is to predict [feature] and extract the feature. Thus, the above rules (c) and (d) should be removed.
2. We need to consider the sequence of items in the conditional part (the left-hand-side) of each rule. In association rule mining, the algorithm does not consider the position of an item in a transaction. However, in natural language sentences, ordering of words is significant. For example, if we consider word sequence, rule (b) above should be:

   easy, to, <V> → [Feature]

   where <V> represents a verb, e.g., "use", which may also be the feature to be extracted. Thus, we need to check each rule against the transaction file to find the possible sequences. This may split the original rule into a few rules according to different sequences. This process is not complex, and thus will not be discussed further due to space limitations. Here, we need minimum confidence (we use 50%) to remove those derived rules that are not sufficiently predictive.
3. Finally, we generate language patterns: Rules still cannot be used to extract features. They need to be transformed into patterns to be used to match test reviews. For example, rules

   <N1>, <N2> → [feature]
   easy, to, <V> → [feature]

   are changed to the language patterns according to the ordering of the items in the rules from step 2 and the feature location:

   <N1> [feature] <N2>
   easy to <V> [feature]

   Note that step 2 and 3 can be computed together. We present them separately for clarity.

Extraction of product features: The resulting patterns are used to match and identify *candidate features* from new reviews after POS tagging. There are a few situations that need to be handled.

1. A generated pattern does not need to match a part of a sentence segment with the same length as the pattern. In other words, we allow gaps for pattern matching. For example, pattern "<NN1> [feature] <NN2>" can match the segment "size of printout".

   Note that our system allows the user to set a value for the maximum length that a pattern could expand. It also allows the user to set the maximum length of a review segment that a pattern should be applied to. These two values enable the user to refine the patterns for better extraction. Note also, the user can add new patterns as well. However, in our experiments reported in Section 5, we did not manually set any of these values or add any pattern (no manual involvement).
2. If a sentence segment satisfies multiple patterns, we normally use the pattern that gives the highest confidence as higher confidence indicates higher predictive accuracy (see feature refinement below as well).
3. For those sentence segments that no pattern applies, we use nouns or noun phrases produced by NLProcessor as features if such nouns or noun phrases exist.

Note that our rule mining method is not applicable to cases that a sentence segment has only a single word, e.g., "heavy" and "big". In such cases, we treat these single words as candidate features.

Feature refinement via frequent terms: In this final step, we want to correct some mistakes made during extraction. Two main cases are handled:

(1) There is a feature conflict, two or more candidate features in one sentence segment, i.e., point 2 and 3 above.

(2) There is a more likely feature in the sentence segment but not extracted by any pattern. For example, "hum" is found to be the feature in the following review segment for a speaker.

   "slight hum from subwoofer when not in use."

   However, the more suitable product feature is "subwoofer". The question is: how does the system know this?

In the above example, if we know that in a number of reviews of the product, "subwoofer" was found as candidate features, e.g.,

"subwoofer annoys people."
"Subwoofer is bulky."

However, "hum" was never found in any other review or never identified as a feature. We can conclude that "subwoofer" is more likely to the genuine feature. Based on this observation, we assume that if a candidate feature X appears more frequently than a candidate feature Y, then X is more likely to be a genuine feature. This assumption is reasonable because a more frequent feature is less likely to be wrong. Our experiment results in Section 5 also confirm this. We can then perform feature refinement for each review segment based on the assumption. In the above example, "subwoofer" is more frequent than "hum", thus "subwoofer" replaces "hum" as the feature for the segment.

We tested two strategies, *frequent-noun* and *frequent-term*. The frequent-noun strategy, which is more restrictive, only allows a noun to replace another noun, e.g., the "subwoofer" and "hum" case above. The detailed procedure is as follows:

1. The generated product features together with their frequency counts are saved in a candidate feature list.
2. We iterate over the review sentences. For each sentence segment, if there are two or more nouns, we choose the noun which is in the candidate feature list and is more frequent.

The frequent-term strategy allows replacement of any type of words. Again, for each sentence segment, we simply choose the word/phrase (it does not need to be a noun) with the highest frequency in the candidate feature list. This strategy comes to help in cases where the POS tagger makes mistakes and/or the product feature is not of type noun. Our experiments results show the frequent-term strategy gives better results than the frequent-noun strategy. It improves the recall and precision values of the product feature extraction significantly.

Mapping to implicit features: We noted earlier that some candidate features represent specific values of the actual features. For example, "heavy" and "big" are not features themselves but are values of <weight> and <size> respectively. Thus, we need to map them to implicit features, <weight> and <size>, respectively.

A similar rule mining technique as above can be used here. In labeling or tagging the training data for mining rules, we also tag the mapping of candidate features to their actual features. For example, when we tag "heavy" in the sentence segment below as a feature word we also record a mapping of "heavy" to <weight>.

"too heavy"

Rule mining can be used to generate mapping rules, which is simple, and thus will not be discussed further.

Computing *Pset* and *Nset*: *Pset* and *Nset* for each feature of every product is easily computed (for visualization) as we know whether the feature is from Pros or Cons of a review.

3.3.2 Grouping Synonyms

It is common that people use different words to describe a feature, i.e., "photo", "picture" and "image" all refers to the same feature in digital camera reviews. For effective visualization, it is important to group features with similar meaning together. Our current system uses a simple method. The basic idea is to employ WordNet [12] to check if any synonym groups/sets exist among the features. For a given word, it may have more than one sense, i.e., different synonyms for different senses. However, we cannot use all the synonyms as they will result in many errors. For example, movie and picture are considered as synonyms in a sense, or in a synset (defined in WordNet). This is true when we talk about Hollywood movies. However, in the case of a digital camera review, it is not suitable to regard picture and movie in one synset, as picture is more related to photo while movie refers to video. To reduce the occurrence of such situations, we choose only the top two frequent senses of a word for finding its synonyms. That is, word A and word B will be regarded as synonyms only if there is a synset containing A and B that appear in the top two senses of both words.

3.4 Semi-Automated Tagging of Reviews

It is very hard, if not impossible, for any automatic technique to achieve perfect accuracy due to the difficulty of natural language understanding. The techniques presented in Section 3.3 and those in [17] alone are useful in situations where a fast and approximate solution is sufficient. For applications that need near-perfect solutions, human analysts have to be involved to correct errors made by automatic techniques (which generate the first-cut solution). Opinion Observer enables analysts to correct errors using a convenient user interface, which also displays the results of automatic techniques for each review sentence. This is called semi-automatic tagging in this paper.

A tagging interface is given in Figure 4. On the top left corner, the analyst can choose a product by selecting a brand and a model. After that, he/she can click on "Retrieve Reviews" to retrieve all the reviews of the product from the database. The ID and the title (if any) of the reviews are displayed in the window on the left. The analyst can click on a title to display the full review in the window in the middle.

The analyst then can read the review. For reviews of format (2), he/she simply clicks on "Pros" or "Cons", which will then be highlighted in red (Figure 4) and all the features produced by the automated technique will be displayed on the right. The window in the middle (on the right) lists all the features identified by the automatic techniques. Positive and negative opinions are indicated by thumbs-up and thumbs-down.

For reviews of format (3) or (1), if the analyst finds that sentence *i* contains product features on which the reviewer has expressed an opinion, he/she selects the sentence by clicking on the sentence. Due to space limitations, this interface is not given here. The product features and opinions found by the automatic techniques in [17] are also displayed in the window on the right.

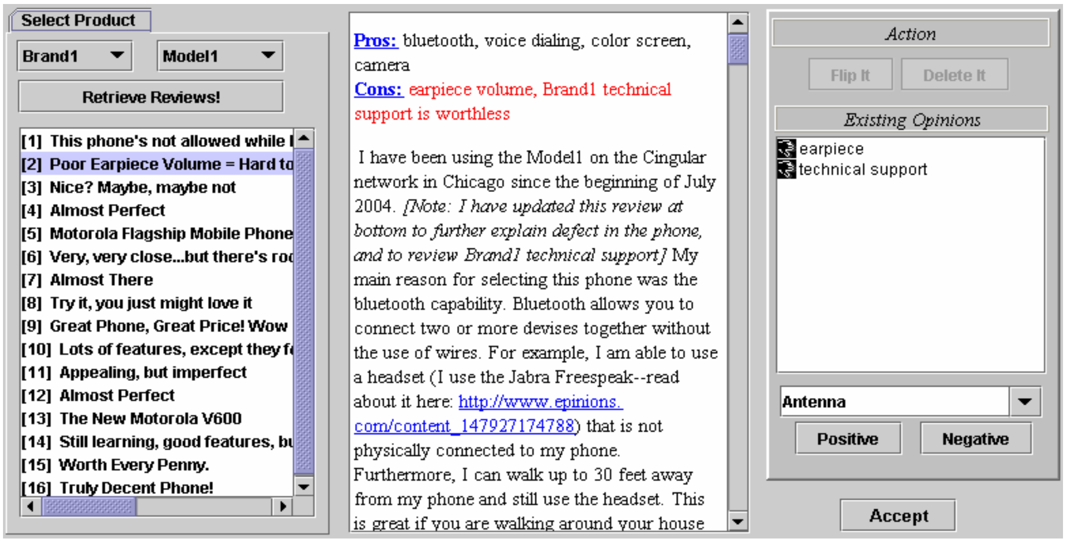
If the results generated by automatic techniques are correct, the analyst simply clicks on "Accept". If a feature is wrong, he/she can delete the feature. If a feature is missing, he/she can select an existing feature from the drop-down list or add a new feature by typing in (or cut-and-paste to) the feature slot. If the opinion orientation on a feature is not correct, he/she can also change it. This semi-automatic tagging is much more efficient than manual tagging with no help as we will see in the experiment section.

Note that it is possible that a company analyst can supply a list of product features. Then, the system only needs to map those identified features from reviews to the supplied features. We did not study this issue in this work, but plan to study it in the future. It should also be noted that in many cases using only the supplied features is insufficient because customers may mention something that the analyst has never thought of, i.e., unexpected features.

3.5 Extracting Reviews from Web Pages

In order to analyze reviews, we need to first extract them from Web pages. Note that this step is not needed if a merchant who already has reviews at its site (e.g., amazon.com) or a dedicated review site (e.g., epinions.com) wants to provide the service.

To perform the extraction task automatically is a non-trivial task. Manually browsing the Web and doing cut-and-paste is clearly not acceptable. It is also too time consuming to write a site specific extraction program for each site. Fortunately, there are existing technologies for this purpose. One approach is wrapper induction [24]. A wrapper induction system allows the user to manually label a set of reviews from each site and the system learns extraction rules from them. These rules are then used to automatically extract reviews from other pages at the same site. Another approach is to automatically find patterns from a page that contains several reviews. These patterns are then employed to extract reviews from other pages of the site. Both these approaches are based on the fact that reviews at each site are displayed according to some fixed layout templates. We use the second approach which is provided by our system MDR-2 [37], which is improvement of MDR [22]. MDR-2 is able to extract individual data fields in data records. Due to space limitations, we will not discuss it further (see [22][37] for more details).



[1] This phone's not allowed while I
[2] Poor Earpiece Volume = Hard to
[3] Nice? Maybe, maybe not
[4] Almost Perfect
[5] Motorola Flagship Mobile Phone
[6] Very, very close...but there's roo
[7] Almost There
[8] Try it, you just might love it
[9] Great Phone, Great Price! Wow
[10] Lots of features, except they fi
[11] Appealing, but imperfect
[12] Almost Perfect
[13] The New Motorola V600
[14] Still learning, good features, bu
[15] Worth Every Penny.
[16] Truly Decent Phone!

Pros: bluetooth, voice dialing, color screen, camera
Cons: earpiece volume, Brand1 technical support is worthless

I have been using the Model1 on the Cingular network in Chicago since the beginning of July 2004. *[Note: I have updated this review at bottom to further explain defect in the phone, and to review Brand1 technical support]* My main reason for selecting this phone was the bluetooth capability. Bluetooth allows you to connect two or more devises together without the use of wires. For example, I am able to use a headset (I use the Jabra Freespeak--read about it here: http://www.epinions.com/content_147927174788) that is not physically connected to my phone. Furthermore, I can walk up to 30 feet away from my phone and still use the headset. This is great if you are walking around your house

Figure 4: Tagging interface: An example review of format (2).

4. SYSTEM ARCHITECTURE

Opinion Observer is designed for use by product manufacturers. A manufacturer can compare consumer opinions from various sources to benchmark its products against those of its competitors. Note that a simpler system can also be built for an online merchant site that has reviews. Figure 5 gives the system architecture. Below, we describe each component:

1. Review sources: These are Web pages containing reviews of the products that the user is interested in. The entry page URLs of these sources are provided by the analyst/user.
2. Review extraction: It extracts all reviews from the given URLs and put them in the database (see Section 3.5).
3. Database: It stores both raw reviews and processed reviews. Currently, we use the database system, mySql [25].
   a. Raw reviews: these are the original reviews extracted from the user-supplied sources on the Web.
   b. Processed reviews: These are reviews that have been processed by the automatic techniques and/or interactively tagged (corrected) by the analyst(s) (see below).
4. Automatic review processing: This component automatic performs review processing to produce the results as described in Section 3.3 and [17].
5. Analyst: This is the company analyst who takes the automatically processed reviews and corrects any errors interactively using the user interface (Figures 4).
6. UI (user interface): It enables analysts and users to interact with the system. Some of the interfaces are shown in Figures 2 and 4.

Clearly, this general architecture can be simplified or customized for other usage situations.

5. EXPERIMENT RESULTS

We now evaluate the proposed automatic technique to see how effective it is in identifying product features from Pros and Cons in reviews of format (2). We also assess the time saved by semi-automatic tagging over manual tagging.

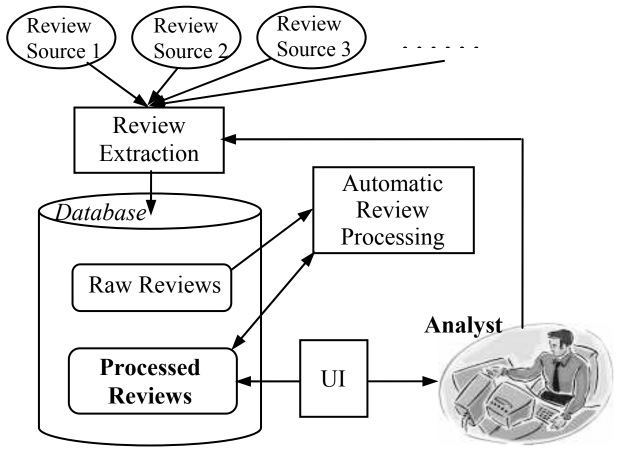


Figure 5: System architecture.

Training and test review data: We manually tagged a large collection of reviews of 15 electronic products. 10 of them are used as the training data to mine patterns. These patterns are then used to extract product features from test reviews of the rest 5 products (Pros and Cons are considered separately). All the reviews are extracted from epinions.com.

Evaluation measures: We use recall (r) and precision (p) as evaluation measures:

$$r = \frac{\sum_{i=1}^{n} EC_i}{\sum_{i=1}^{n} C_i}, \quad \text{and} \quad p = \frac{\sum_{i=1}^{n} EC_i}{\sum_{i=1}^{n} E_i},$$

where n is the total number of reviews of a particular product, EC_i is the number of extracted features from review i that are correct, C_i is the number of actual features in review i, E_i is the number of extracted features from review i. This evaluation is based on the result of every review as it is crucial to extract features correctly from every review. It is not suitable to simply compare the set of extracted features (no duplicates) from all reviews of a product with the set of manually identified features as it does not measure how effective the extraction is for individual reviews.

Note that we generate language patterns and product features separately for Pros and Cons as this gives better results. Table 1 shows the experimental results for Pros. Column 1 lists each data set (product). Columns 2 to 7 give each stage of product feature extraction. Columns 2 and 3 are the recall and precision results of using only automatic generated language patterns. Columns 4 and 5 show the recall and precision results after the frequent-noun strategy is applied to refine the features extracted by using only patterns. Columns 6 and 7 give the recall and precision results after the frequent-term strategy is applied to refine the features extracted by using only patterns. Comparing the two strategies, we observe that the frequent-term strategy gives better results than the frequent-noun strategy. The reason for this is that some features are not expressed as nouns and POS tagger also makes mistakes. From columns 6 and 7, we can see that the frequent-term strategy improves the results of patterns only significantly.

Table 2 shows the same set of results for Cons. Again, the same observations can be made. Tables 1 and 2 demonstrate the proposed techniques are very effective (with high accuracy).

Table 3 gives the results of two baselines methods. Columns 2-3 and 6-7 show the results of using nouns and noun phrases as features based on POS tagging for Pros and Cons respectively. Using nouns and noun phrases is reasonable because intuitively product features are nouns. The results indicate that many features appear explicitly as nouns or noun phrases. However, there are still some adjectives and verbs appear as implicit features, which cannot be found. We also observe that POS tagging makes many mistakes due to the brief segments (incomplete sentences) in Pros and Cons. Columns 4-5 and 8-9 show the recall and precision of the FBS system in [17]. The low recall and precision values indicate that the techniques there are not suitable for Pros and Cons, which are mostly short phrases or incomplete sentences. Clearly, from Tables 1, 2, and 3, we can see that the recall and precision of the proposed technique are much higher than those of the two existing methods.

From the tables, we also observe that the results for Pros are better than those for Cons. After reading through the reviews and the generated patterns for Pros and Cons carefully, we found that people tend to use similar words like 'excellent', 'great', 'good' in Pros for various product features. In contrast, the words that people use to complain differ a lot in Cons. Consequently, there are some patterns contain specific words for Pros, e.g., excellent <NN> [feature], great <NN> [feature], but for Cons, there is no such pattern but only those patterns consisting of POS tags, e.g., <JJ> <NN> [feature]. Thus, this results in significantly fewer generated patterns for Cons than for Pros (22 vs. 117). Because we use nouns or noun phrases if a segment does not match any pattern, the small number of patterns for Cons result in a large number of segments using nouns or noun phrases as product features. As we discussed before, there are still features that are adjectives and verbs, which are missed. Cons needs further investigation in order to achieve better results.

Semi-automatic tagging: If the analyst wishes to correct errors made by the automatic techniques. He/she can read the reviews and use the user interface in Figure 4 to perform the task. Since most results produced by our automatic techniques are correct, the process is much more efficient than manual tagging. We experimented in two settings using the same interface:

(1) Manual tagging (i.e., without using the results of automatic techniques): The analyst reads, manually extracts each feature (via cut-and-paste and/or search through the drop-down list) and decides the opinion orientation.

(2) Semi-automatic tagging (using the results from the automatic techniques). The analyst only corrects errors.

Table 1: Recall and precision results for Pros

Pros	Patterns only		Frequent-noun strategy		**Frequent-term strategy**	
	Recall	Prec.	Recall	Prec.	**Recall**	**Prec.**
data1	0.878	0.880	0.849	0.861	**0.922**	**0.876**
data2	0.787	0.804	0.798	0.821	**0.894**	**0.902**
data3	0.782	0.806	0.758	0.782	**0.825**	**0.825**
data4	0.943	0.926	0.939	0.926	**0.942**	**0.922**
data5	0.899	0.893	0.878	0.881	**0.930**	**0.923**
Avg.	0.857	0.862	0.844	0.854	**0.902**	**0.889**

Table 2: Recall and precision results for Cons

Cons	Patterns only		Frequent-noun strategy		**Frequent-term strategy**	
	Recall	Prec	Recall	Prec	**Recall**	**Prec**
data1	0.900	0.856	0.867	0.848	**0.850**	**0.798**
data2	0.795	0.794	0.808	0.804	**0.860**	**0.833**
data3	0.677	0.699	0.834	0.801	**0.846**	**0.769**
data4	0.632	0.623	0.654	0.623	**0.681**	**0.657**
data5	0.772	0.772	0.839	0.867	**0.881**	**0.897**
Avg.	0.755	0.748	0.801	0.788	**0.824**	**0.791**

Table 3: Recall and precision results of nouns and FBS

	Pros				**Cons**			
	Noun/noun phrases		FBS		Noun/noun phrases		FBS	
	Recall	Prec	Recall	Prec	Recall	Prec	Recall	Prec
data1	0.543	0.524	0.400	0.476	0.681	0.409	0.419	0.424
data2	0.747	0.642	0.494	0.567	0.536	0.249	0.485	0.508
data3	0.551	0.521	0.431	0.508	0.642	0.327	0.486	0.494
data4	0.728	0.682	0.411	0.441	0.758	0.354	0.496	0.506
data5	0.664	0.631	0.480	0.560	0.859	0.487	0.469	0.474
Avg.	0.647	0.600	0.443	0.510	0.70	0.365	0.471	0.481

Our experiment results with two human taggers show that the amount of time saved by the second method is around 45% (including time used for reading the reviews). Without our visual interface, the manual method will be much more time consuming.

Another saving in time and effort is from automatic extraction of reviews from Web pages. Manual cut-and-paste will be extremely time consuming, and cannot scale to a large number of reviews.

Finally, regarding synonym grouping, our method achieves 52% recall and 100% precision on these data as the method is very conservative. The main problem with our simple method is that it does not handle context-dependent synonyms. This is a hard topic in NLP and has not been the focus of this work. We will study this more in the future. We do not list the result for each dataset as there are only a few synonyms in each dataset.

6. CONCLUSIONS

Consumer opinions used to be very difficult to find before the Web was available. Companies often conduct surveys or engage external consultants to find such opinions about their products and those of their competitors. Now much of the information is publicly available on the Web. In this paper, we focused on one type of opinion sources, customer reviews of products. We proposed a novel visual analysis system to compare consumer opinions of multiple products. To support visual analysis, we designed a supervised pattern discovery method to automatically identify product features from Pros and Cons in reviews of format (2). A friendly interface is also provided to enable the analyst to interactively correct errors of the automatic system, if needed, which is much more efficient than manual tagging. Experiment results show that the system is highly effective. In our future work, we will improve the automatic techniques, study the strength of opinions, and investigate how to extract useful information from other types of opinion sources.

7. ACKNOWLEDGMENTS

We thank Tom Tirpak, Weimin Xiao, Richard Anderson and Ralph Pini from Motorola for many useful discussions. The name of our system (Opinion Observer) was also suggested by Tom Tirpak. This work was partially supported by NSF (IIS-0307239).

8. REFERENCES

[1]. Agrawal, R., Rajagopalan, S., Srikant, R., Xu, Y. 2003. Mining newsgroups using networks arising from social behavior. *WWW'03*.

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

[3]. Aoki, P. and Woodruff, A. 2004. "User Interfaces" and the Social Negotiation of Availability**.** *ACM SIGCHI Workshop on Forecasting Presence and Availability*.

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

[5]. Bunescu, R., Mooney, R. 2004. Collective Information Extraction with Relational Markov Networks. *ACL'2004*.

[6]. Chi, E., Hong, L., Heiser, J., Card, S. 2004. eBooks with Indexes that Reorganize Conceptually. *Proc. of CHI2004 Conference Companion*.

[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]. Etzioni, O., Cafarella, M., Downey, D., Kok, S. Popescu, A., Shaked, T., Soderland, S., Weld, S. Web-Scale Information Extraction in KnowItAll (Preliminary Results). *WWW'2004*.

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

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

[13]. Freire, J., Kumar, B. and Lieuwen, D. 2001. WebViews: Accessing personalized Web content and services. *WWW'10*.

[14]. Freitag, D., McCallum, A. 2000. Information extraction with HMM structures learned by stochastic optimization. *AAAI'00*.

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

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

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

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

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

[20]. Lafferty, J., McCallum, A., Pereira, F. 2001. Conditional random fields: probabilistic models for segmenting and labeling or sequence data. *ICML'01*.

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

[22]. Liu, B., Grossman, R., and Zhai, Y. "Mining data records from Web pages." *KDD'03*, 2003.

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

[24]. Muslea, I., Minton, S. and Knoblock, C. "A hierarchical approach to wrapper induction." *Agents'99*, 1999.

[25]. MySql, http://www.mysql.com/

[26]. Nasukawa, T. and Yi, J. 2003. Sentiment analysis: Capturing favorability using natural language processing. *Proceedings of the 2nd Intl. Conf. on Knowledge Capture (K-CA'2003)*.

[27]. Nigam, K., and Hurst, M. Towards a Robust Metric of Opinion. *AAAI Spring Symposium on Exploring Attitude and Affect in Text*. 2004.

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

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

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

[31]. Rosario, B., and Hearst, M., Classifying Semantic Relations in Bioscience Text. *ACL'2004*.

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

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

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

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

[36]. Yu, H., and Hatzivassiloglou, V. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. *EMNLP'03*.

[37]. Zhai, Y., and Liu, B. Web data extraction based on partial tree alignment. *WWW'05*, 2005.