Mining Product Reputations on the Web

Satoshi Morinaga, Kenji Yamanishi
NEC Corporation
4-1-1, Miyazaki, Miyamae, Kawasaki, Kanagawa
216-8555, JAPAN. TEL: 81-44-856-2143
morinaga@cw.jp.nec.com, k-yamanishi@cw.jp.nec.com

Kenji Tateishi, Toshikazu Fukushima
NEC Corporation
8916-47, Takayama-cho, Ikoma, Nara
630-0101, JAPAN. TEL: 81-743-72-3341
k-tateishi@bq.jp.nec.com, t-fukushima@cj.jp.nec.com

ABSTRACT

Knowing the reputations of your own and/or competitors' products is important for marketing and customer relationship management. It is, however, very costly to collect and analyze survey data manually. This paper presents a new framework for mining product reputations on the Internet. It automatically collects people's opinions about target products from Web pages, and it uses text mining techniques to obtain the reputations of those products.

On the basis of human-test samples, we generate in advance syntactic and linguistic rules to determine whether any given statement is an opinion or not, as well as whether such any opinion is positive or negative in nature. We first collect statements regarding target products using a general search engine, and then, using the rules, extract opinions from among them and attach three labels to each opinion, labels indicating the positive/negative determination, the product name itself, and an numerical value expressing the degree of system confidence that the statement is, in fact, an opinion. The labeled opinions are then input into an opinion database.

The mining of reputations, i.e., the finding of statistically meaningful information included in the database, is then conducted. We specify target categories using label values (such as positive opinions of product A) and perform four types of text mining: extraction of 1) characteristic words, 2) co-occurrence words, 3) typical sentences, for individual target categories, and 4) correspondence analysis among multiple target categories.

Actual marketing data is used to demonstrate the validity and effectiveness of the framework, which offers a drastic reduction in the overall cost of reputation analysis over that of conventional survey approaches and supports the discovery of knowledge from the pool of opinions on the web.

1. INTRODUCTION

1.1 Motivation

SIGKDD 02 Edmonton, Alberta, Canada

Knowing the reputations of your own and/or competitors' products is important for marketing and customer relationship management. Questionnaire surveys are conducted for this purpose, and open questions are generally used in the hope of gaining valuable information about reputations.

One problem in dealing with survey data is that the manual handling of it is both cumbersome and very costly, especially when it exists in large volume, and computerized mining of open answers (i.e., the answers to open questions) is crucial. For this purpose, we have previously proposed a text-classification-type survey analysis tool[15] that is particularly well-suited to brand image analysis. Throughout this paper, we refer to this tool as *SA*[1].

Another problem is that it is very costly to gather the large volume of high quality survey data, which is necessary for meaningful analysis of reputations. One approach which promises to reduce costs in this regard is the recently proposed *opinion extraction* [24], which is able to automatically extract opinions about specific products as expressed on the web. This can be thought of as a kind of information extraction (see, e.g., [1], [4], [10], [8], [9], [7] or as a kind of question-answering (see e.g., [13],[6],[17],[22],[23],[12],[18],[19],[25]), both of which have extensively been explored in the field of web text retrieval.

The purpose of this paper is to provide a general new framework for automatically collecting and analyzing opinions on the Internet. With it, it is possible to drastically reduce total costs for marketing research and analysis and to support the knowledge discovery on the Internet. This framework has been created by combining the opinion extraction technique developed in [24] with text mining methodologies, two of which were previously employed in *SA* [15]. The key to the combination of opinion extraction with text mining is *opinion labeling*. In the opinion extraction process, labels are attached to each of the opinions, and then, in the text mining process, supervised learning from labeled opinions is conducted to extract statistically meaningful knowledge, which we refer to here as *reputation*.

Let us roughly illustrate how the proposed framework works. A user inputs, for example, three PDA(personal digital assistants) product names (products A, B, and C). The system then collects people's opinions about them from the Internet and attaches three labels to each: 1) the name of the product referred to, 2) the positive/negative nature of the opinion, and 3) opinion-likeliness, i.e., a numerical value

[1]This tool is available by the name of SurveyAnalyzer in Japan. SurveyAnalyzer^{TM} is a trademark of NEC Corporation in Japan.

the degree of system confidence that the extracted statement is, in fact, an opinion. These labeled opinions are then put into an opinion database.

The next step is reputation analysis. Each possible combination of labels represents a category of the opinion data. A user specifies a target category for the analysis using the value of labels (e.g., [product A, positive]), and the system conducts text mining to extract reputations corresponding to the specified target. Text mining consists of the following four tasks:

a) *Rule analysis (extracting characteristic words):* For a specified target category, we extract individual words and combinations of words that are characteristic of it. For example, for the category of [product A, positive], words such as "monochrome", "inexpensive", as well as such pairings as "lightweight" and "convenient," would be extracted if they appear significantly more frequently there than in other categories. The information criterion used for this purpose is *stochastic complexity* [21], defined as the minimum code-length required to encode a sequence into a binary sequence under the prefix coding condition. By extracting characteristic words, a user is able to gain an overall sense of the features of the target category.

b) *Co-occurrence analysis:* For each characteristic word extracted in Rule analysis a), we further extract a list of words which significantly co-occur with it. For example, for the category of [product A, positive], for the word "monochrome," we might obtain a list of significantly co-occurring words that includes: "display," "text," "email," etc. This extraction is also conducted on the basis of stochastic complexity. Extracting co-occurring words helps users better understand contexts in which characteristic words appear.

c) *Typical sentence analysis:* For a specific category, we assign to each opinion in it a score that indicates how typical its vocabulary is with respect to the other opinions in that category. Opinions are then output in descending order of these scores. Individual scores are calculated on the basis of the Baysian theory and the naive Bayes approach to posterior probability computation.

d) *Correspondence analysis*[5]: We, next generate a two-dimensional positioning map that visually displays the correspondence relationships among target categories and the characteristic words extracted for them, with distance on the map being a representation of correspondence (closeness). This is done by applying the statistical method of *principal component analysis* to frequency data for individual extracted words. Analyzing correspondence relationships helps the user to understand what categories are close one another and what keywords are shared in common by different categories.

We consider these four tasks to be the most fundamental for useful reputation analysis. They are all based on probabilistic modeling of mechanisms for text generation and are conducted on the basis of information-theoretical/statistical approaches. Analyses a) and d) were previously introduced in [15] as *SA* functions, b) and c) are first being introduced here.

1.2 Related Work

Although our proposed framework for reputation mining–combination of two components: opinion extraction and text mining using *SA*, is new in the area of marketing, each of the components itself also has novelty and effectiveness for marketing.

Opinion extraction, one of the key technologies used in our framework, was developed by Tateishi, Ishiguro, and Fukushima [24]. It is closely related to *information extraction* techniques (see e.g., [1], [4], [10], [8], [9], [7]), in which a wrapper or a specific extraction procedure is built automatically or manually and used to extract specific pieces of information requested by a user. For example, Shopbot [9], [7] uses HTML tags to automatically extract product prices or specifications. Our proposed framework, by way of contrast, attaches labels to extracted information, which makes it possible to apply a supervised learning approach to text mining, and this distinguishes it from conventional information extraction techniques.

Opinion extraction can also be thought of as a kind of *QA (question-answering)* (see e.g., [13],[6],[17],[22],[23],[12]), which has extensively been explored in the field of text retrieval. In fact, it can be conducted by asking an ordinary QA system what opinions exist for target products. The QA system will output opinions, giving each a likelihood of its being included in an answer to that question. Among QA systems, Tateishi et.al.'s system [24] is particularly well suited to opinion extraction since it prepares specialized dictionaries for major product fields and unique syntactic rules for calculating opinion-likeliness. These improve opinion-search results significantly, and it reportedly offers a particularly high rate of accuracy with respect to searching out opinions about a target product: A precision rate of 86.6% for the top 17.1% of total search results, while the portion of total search results actually containing opinions, obtained by a general-purpose search engine (Google) was 15.9% in total[24].

Various text mining techniques have been applied to analyzing open answers in questionnaire data, including those which employ text-clustering techniques (e.g., [14]). The idea here is to view each answer as a vector of words, and to cluster vectors on the basis of similarity measures. Such methods are effective for summarizing (grouping) answers, but they are not effective for extracting analysis target characteristics. Methods for analyzing open answers on the basis of associations between words have also been proposed (e.g., [11]). More specifically, associations between word pairs are calculated on the basis of their co-occurrences in open answers, and they are visually presented on a two-dimensional positioning map. In most of the previous work on positioning maps, redundant words tend to appear too frequently, making the maps hard to understand. This suggests the necessity of preprocessing step in which characteristic words are first extracted.

A text classification approach to survey analysis has been proposed by Li and Yamanishi [15]. Texts consist of open answers contained in questionnaire results, and categories are specified on the basis of closed answers (i.e., answers to closed questions, for which possible responses have been limited, as in check lists). In this approach, a classification rule that assigns a text into one of some number of categories is learned from training examples, and the keywords that appear in the rule can be thought of as those that are characteristic of the category. The key to this approach is supervised learning from labeled examples, which is well-suited to the analysis of the labeled opinions that the opinion extraction system produces.

In our framework we combine an information extraction

technique (here, opinion extraction[24]) with a text classification technique (here, that of Li and Yamanishi[15]). To the best of our knowledge, such a combination has never been reported before.

Section 2 below gives a brief sketch of our reputation analysis framework. Section 3 describes the opinion extraction technique proposed in [24]. Sections 4–7 describe how we analyze extracted opinion data with, respectively, rule analysis, co-occurrence analysis, typical sentence analysis, and correspondence analysis. In Section 8 we evaluate the validity and effectiveness of our framework. Section 9 gives concluding remarks.

2. REPUTATION MINING SYSTEM

Figure 1 gives a flow overview for our reputation mining

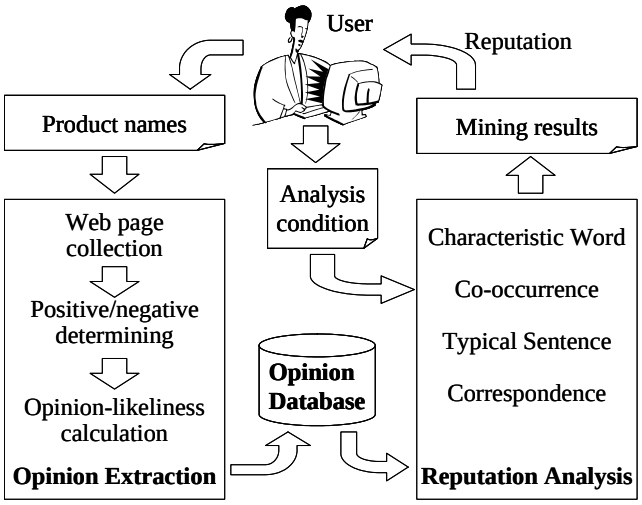

Figure 1: Reputation mining

framework. The system performs two functions: opinion extraction and reputation analysis. A user can first input product names (e.g., Products A, B, and C, all of which are cellular phones) into the system, and the opinion extraction function will use a search engine to collect web pages that include those names. It then extracts sentences that express *opinions* regarding these products and inputs them into an opinion database. The text mining function, which is the major subject of this paper, takes as input an analysis condition specifying the target category, and it outputs its mining results.

3. OPINION EXTRACTION

The opinion extraction function consists of the following modules [24]: Web page collection module, positive/negative determining module, and opinion-likeliness calculation module

1. *Web page collection module:*
This module uses a crawler to collect web pages relevant to input product names.

2. *Positive/negative determining module:*
For each of the collected pages, this module first extracts sets of sentences that include *evaluation-expressions* about the products. These are checked with a previously prepared *evaluation-expression dictionary*. It then selects from among them sentences in which evaluation expressions are located within a certain distance of the relevant product name, and it designates these as opinions. Here each evaluation-expression is registered in the dictionary as being either positive or negative. For example, in the field of computer equipment, "fast," "good," "light," "satisfied," and "recommendable" are positive evaluation expressions, while "heavy," "easily broken," "noisy," and "unstable" are negative. On the basis of evaluation expression dictionary entries, the "positive" or "negative" nature of each opinion is determined, taking into account, naturally, linguistic negation. If, for example, an inherently "positive" expression like "low-cost" appears within a certain distance of a negating expression, such as "insufficient," the opinion as a whole will be deemed "negative."

3. *Opinion-likeliness calculation module:*
For each opinion obtained by the previous module, this module calculates its opinion-likeliness score, a real value ranging from 1 to 5, indicating the relative likelihood that the statement represents an opinion: the higher the score, the higher the likelihood. This score is calculated using syntactic property rules, which can either be learned manually from training examples (see [24] for details) or by a standard machine learning technique, such as decision-tree-induction.

The labeled opinions are input into the opinion database. Table 1 shows an example of 6 such opinions. (In this paper, all opinions have been translated from Japanese to English.)

4. REPUTATION ANALYSIS

4.1 Rule Analysis (Characteristic-Word Extraction)

The first step in mining opinions here is to extract keywords that are indicative of a specified category. In order to do this, we learn text classification rules and association rules from examples (see [15],[16]). The learned rules are basically lists of words that must be present for a new text to be classified into a specific category. These are the *characteristic words* of the category. Extracting characteristic words for each category helps us to discover differences in opinions between the target category and other categories.

The task of rule-based text classification can be described as follows: We have a number of categories, each already containing a number of texts as training examples; we are to automatically acquire a set of rules from them and then classify new texts on the basis of those acquired rules. Here we employ text classification based on a *stochastic decision list* [26],[15],[16] consisting of an ordered sequence of IF-THEN-ELSE rules for assignment of new opinions to a given category. The condition part (IF part) may require the simultaneous presence of several words or simply the presence of a single word, and the consequence part (THEN part) specifies a category. Each rule also attaches a probability (relative frequency) value to its assignment.

The words in the condition part of the obtained classification rules for a specific category represent the characteristics of opinions there. We can extract characteristic words for a specific category in the opinion database by learning text classification rules, using the opinion database itself for training examples.

For example, Figure 2 shows a classification rule for the category specified by [product name = cellular phone A]

Table 1: Data Records in the Opinion Databese

Product name	Nature	Opinion-likeliness	Opinion
cellular phone A	Positive	4.05	cellular phone A is my favorite.
cellular phone A	Negative	2.74	I am a cellular phone A user, even though it is said to be inconvenient in some ways.
cellular phone C	Negative	3.37	I feel a little unsatisfied with cellular phone C because it has fewer functions than other models.
cellular phone E	Positive	2.91	I'm satisfied with my present phone –cellular phone E–.
cellular phone B	Positive	4.12	You can only download five melodies to cellular phone C, so I recommend cellular phone B.

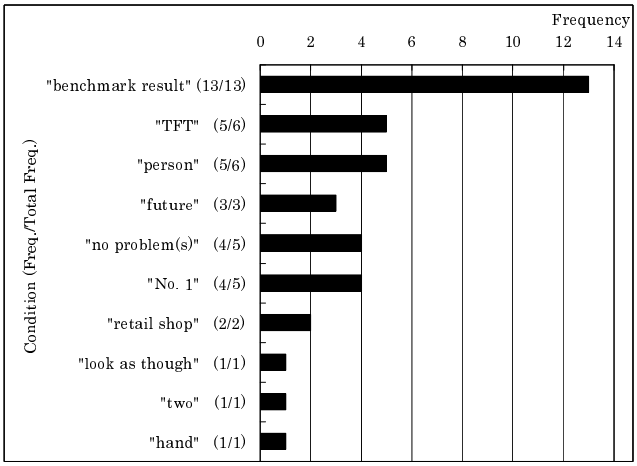

Figure 2: Classification Rules for Cellular Phone A

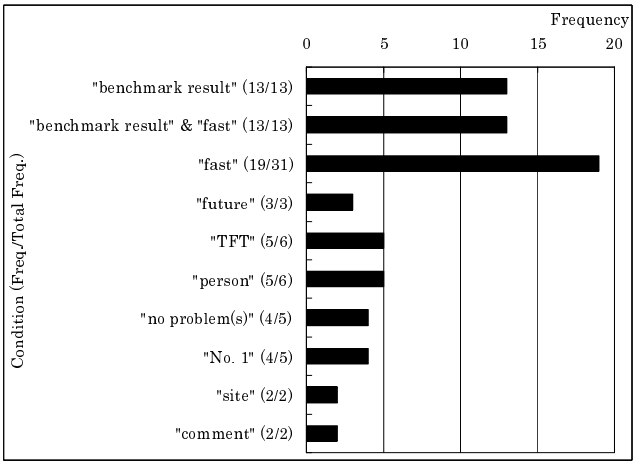

Figure 3: Association Rules for Cellular Phone A

(each English expression has been translated from a single Japanese vocabulary item). The first rule indicates that if an opinion contains the expression "benchmark result" then it should be classified as an opinion about cellular phone A with a probability of 13/13. If it does not but does contain the word "TFT," then it should also be classified as an opinion about cellular phone A, though with a probability of 5/6. The open answer is examined in this way with respect to the rules, from beginning to end. The extracted words such as "benchmark result," "TFT," etc. can be thought of characteristics of cellular phone A, which we recognize as a reputation of this product.

In Figure 2, frequency bars denote the number of opinions in the category that contain each individual word, respectively. The two numbers in parentheses next to each expression indicate the number of opinions containing each expression in the category and the database as a whole, respectively.

We also employ association rules consisting of an ordered sequence of IF-THEN-OR rules. These represent the strength of associations between opinions and a target category. Each rule has a condition for a given association, one that the presence of a single word or the simultaneous presence of some number of words. It also attaches a probability (relative frequency) value to each association.

As an example, Figure 3 shows association rules for the category [cellular phone A]. The first rule indicates that there are 13 opinions containing the expression "benchmark result" in the opinion database, and that all of them are in the category of [product name = cellular phone A]. The third rule indicates that in the opinions at large there are 31 opinions containing the word "fast" and that 19 of them are opinions regarding cellular phone A. These extracted words or combinations of words can be thought of as characteristic of the category.

For purposes of comparison, let us look at a word histogram, which consist of words listed in descending order with respect to the number of opinions containing them in the category. Figure 4 shows a word histogram for the category [cellular phone A]. Note that such generic words as "do" or "become" would obviously have no relevance to our purposes.

In constructing classification and association rules, we employ stochastic complexity [21] as a criterion for selecting words. Below we describe the mathematics of such selection.

We denote a text with label A as "1" and that with any other label as "0." Thereby we denote a set of texts D as a binary sequence. We denote a subsequence of D which consists of texts including a word or a phrase w as $E(w)$ and the remaining sequence as $D - E(w)$.

Let $I(E(w))$ and $I(D-E(w))$ be the information-theoretic complexity of $E(w)$ and $D - E(w)$, respectively. In general, for a binary sequence x, its information-theoretic complexity, denoted as $I(x)$, is calculated using stochastic complexity as follows:

$$I(x) = mH\left(\frac{m_1}{m}\right) + \frac{1}{2}\log\frac{m\pi}{2}. \qquad (1)$$

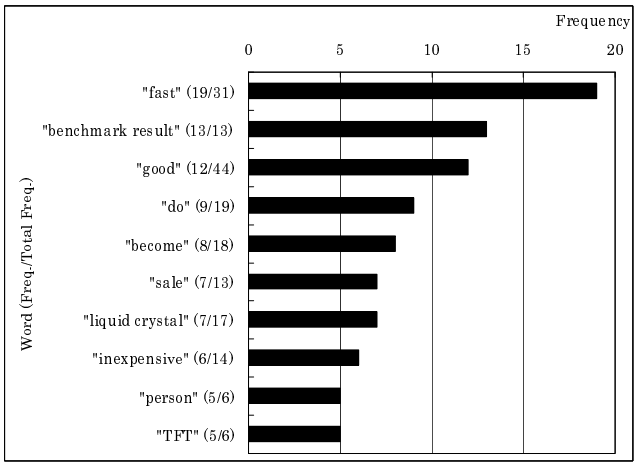

Figure 4: Word Histogram for Cellular Phone A

Alternatively, we may calculate $I(x)$ using *extended stochastic complexity* [27] as follows:

$$I(x) = \min\{m_1, m - m_1\} + C\sqrt{m \log m}. \quad (2)$$

Here, m denotes the length of x, m_1 the number of "1"s in x, $H(z) = -z\log_2 z - (1-z)\log_2(1-z)$, and C is a certain positive constant. Note that the stochastic complexity of x is interpreted as the shortest code length required to encode x using a given probabilistic model (in this case, a Bernoulli model) under the prefix coding condition [21]. Here, stochastic complexity is also considered from the viewpoint of a statistical decision theory a loss for predicting x in the case where the logarithmic loss function is used as a distortion measure. Extended stochastic complexity can be considered to be a general extension of stochastic complexity, in the sense that a general loss function is employed as a distortion measure (in (2), we use a "discrete loss").

We calculate the score of w as follows:

$$Score(w) = \frac{1}{m}\left(I(D) - (I(E(w)) + I(D - E(w)))\right). \quad (3)$$

$Score(w)$ represents information gain achieved by the selection of w, which intuitively shows how much the stochastic complexity of the original data sequence can be reduced by separating it into two parts: that which contains w and that which does not. A larger $Score(w)$ indicates that w is either characteristic of the set of all "1" texts or of the set of all "0" texts. If we were to eliminate the second term from stochastic complexity formula (1), leaving only the entropy (the first term), $Score(w)$ would then become equivalent to the mutual information often used in decision-tree splitting[20]. Notice here that stochastic-complexity-based $Score(w)$ is a more precise measure of information included in a data sequence of finite length than entropy, and the former will converge to the latter as the length of the sequence increases to infinity.

In the learning of classification rules, we calculate, on the basis of all the data, a $Score(w)$ value for each of the possible rules. Here, the condition of a rule may include not only the presence of one word, but also the simultaneous presence of several words. We select as a first rule that for which the $Score(w)$ value is the largest. We then remove from the data those that satisfy the condition of the first rule. For the remaining data, we again calculate the $Score(w)$ value for each of the remaining possible rules, and select as a second rule that for which the $Score(w)$ value is the largest. We repeat this process until we cannot find any rule which is significant in terms of $Score(w)$.

In the learning of association rules, we calculate, on the basis of all the data, a $Score(w)$ value for each of the possible rules, and sort the rules in descending order of their $Score(w)$ values. Note that this algorithm is different from conventional association rule mining algorithms (see e.g., [2]), which perform based on support ("Total Freq." in this paper) and confidence ("Freq./Total Freq." in this paper) only.

4.2 Co-occurrence Analysis

For each characteristic word or phrase extracted from open answers belonging to a specific category, we extract a list of words or phrases that co-occur with that word or phrase. Through this list we are better able to understand the contexts in which the characteristic keywords appear.

Table 2 shows a co-occurring word list for the "No. 1" and "no problem" shown in Figure 2. With no contextual information, characteristic words such as "No. 1" or "no problem" have little meaning for us, but co-occurrence analysis can, for example, help us here to see that cellular phone A is recognized as the No. 1 candidate for something, and that there is no problem in booting up cellular phone A. These contexts help form reputations for cellular phone A.

Table 2: Co-occurring Words

Characteristic Word	Co-occurring Word	Freq.	Score
No. 1	candidate	1	0.0024
	all	1	0.0024
no problem(s)	boot up	3	0.0070
	operation	2	0.0034

Below we describe how to calculate a co-occurrence score for any given pair of keywords. For a word or phrase w, let D_w denote the sequence of texts including w. For another word or phrase w', let $D_w(w')$ denote the subsequence of D each of which includes w'. Then, letting $I(D_w)$, $I(D_w(w'))$, $I(D_w - D_w(w'))$ be the stochastic complexities of D_w, $D_w(w')$, $D_w - D_w(w')$ (as calculated in (1)), respectively, we define the co-occurrence score of w' with respect to w as

$$Score(w' : w) = \frac{1}{m}\left(I(D_w) - (I(D_w(w')) + I(D_w - D_w(w')))\right), \quad (4)$$

where m is the number of texts included in D_w. The larger $Score(w' : w)$ is, the larger the degree of co-occurrence of w' relative to w is.

Maximizing $Score(w' : w)$ w.r.t. w' is equivalent to minimizing $I(D_w(w')) + I(D_w - D_w(w'))$ w.r.t. w' since $I(D_w)$ is independent of w'. Notice also that $Score(w' : w)$ is asymmetric with respect to w and w'; that is, $Score(w' : w) \neq Score(w : w')$ in general.

4.3 Typical Sentence Analysis

For a set of opinions belonging to a specific category, we give a score to each of them, with a high score indicating a high possibility of its being a typical opinion for the cate-

gory. This gives the user a simple overview of tendencies in original opinion data.

Table 3 shows a list of typical sentences in the category specified by [product name = cellular phone A]. The characteristic words extracted above (such as "benchmark result", "no problem(s)", etc.) appear with high frequency in typical high-scoring sentences.

Below we describe how to calculate a score for any given opinion sentence s. Let $\mathcal{W}$ be a set of all words appearing in opinion database and $\mathcal{C}$ be a set of all categories. Let N_c be the number of opinions belonging to the category $c \in \mathcal{C}$. Let $N = \sum_{c \in \mathcal{C}} N_c$. We calculate the occurrence probability of c using its MAP estimate:

$$p(c) = \frac{N_c + \beta}{N + |\mathcal{C}|\beta}, \tag{5}$$

where β is a positive constant usually set to 1/2.

For a category $c \in \mathcal{C}$, let $\mathcal{D}_c$ denote the set of opinion sentences belonging to c and let m_w be the number of occurrences of w in $\mathcal{D}_c$. We can then calculate the occurrence probability of w in c by using its MAP estimate:

$$p(w|c) = \frac{m_w + \beta}{\sum_{w \in \mathcal{D}_c} m_w + |\mathcal{W}|\beta}, \tag{6}$$

where β is a positive constant usually set to 1/2.

Suppose that an opinion s is represented as a sequence of words $w_1, \cdots, w_T$, where $w_i \in \mathcal{W}$ $(i = 1, \cdots, T)$. We calculate the score of s using the Bayes posterior probability of the category c for given s as follows:

$$Score(s) = \frac{p(c) \prod_{i=1}^{T} p(w_i|c)}{\sum_{c \in \mathcal{C}} p(c) \prod_{i=1}^{T} p(w_i|c)}, \tag{7}$$

where we used the naive Bayes assumption that each w_i is independent.

4.4 Correspondence Analysis

We conduct correspondence analysis in order to get what we call a two-dimensional positioning map over the set of analysis objects and keywords extracted from the opinion database. The map visually shows the relationships between the categories and characteristic words, with distance on the map being a representation of correspondence (closeness).

Before performing correspondence analysis, characteristic words are extracted for the categories designated by the user. Specifically, words considered indicative of individual target categories are extracted on the basis of rule analyses. These extracted keywords are in fact equivalent to those contained at the upper reaches of the association rules for the targets. We then construct a table that contains frequency data for extracted words for each of the target categories.

Correspondence analysis can be viewed as an extension of *principal component analysis (PCA)* (which is similar to Singular Value Decomposition). It is conventionally performed, as in [5], on the basis of the frequency data table. Examples of the result of correspondence analysis are shown in the next section.

Note that if correspondence analysis is conducted from original data without extracting characteristic words, an unreadable positioning map, in which many unnecessary words appear, may be generated as a result. That is why using rule analysis in preprocessing is crucial to effective correspondence analysis.

5. EXPERIMENTS

We conducted experiments in three different product fields: cellular phones, PDAs, and Internet service providers. We input the names of five cellular phones, four PDAs, and five Internet service providers into the system's opinion extraction section. The web page collection module ran on a SunEnterprise250 with a 400MHz UltraSPARK-II and a 512MB memory. It collected 1200 pages for each product name (total: 16800) within 3 hours. The positive/negative determining module and the opinion-likeliness calculation module ran on a DELL Precision420 with an 866MHz PentiumIII and a 768MB memory. They processed the pages within 1 hour. Labeled opinions with opinion likeliness not less than 2.5 were recorded in three separate databases in accord with their respective product fields (cellular phones, PDAs, and ISPs). The number of data records (extracted opinions) was, respectively, 519, 1195, and 605. We describe below the results of text mining for each of the fields.

5.1 Cellular Phones

We extracted characteristic words for each of the cellular phones on the basis of learned association rules and performed a correspondence analysis of them. The text mining was processed on an NEC MA86T with an 866MHz Pentium III and a 256MB memory. It took about 10 seconds to learn rules for a single product, as opposed to roughly 2 hours for manual rule analysis (conducted for time comparison). The correspondence analysis took about 5 seconds.

The rule analysis results gave a good indication of the reputation of each cellular phone. For example, we discovered that the red body of cellular phone C had captured the public's fancy, and that cellular phone D attracted people who wanted to replace older models. Figure 5 shows the obtained positioning map. The words plotted around each cellular phone are the top four characteristic words for each. We can see that cellular phone A has a good reputation for its basic performance ("benchmark result", "fast", "no problem(s)"), and cellular phone B has a bad reputation ("doesn't work well", "slow"). We can also see that for cellular phone D its "display" is more of an issue than its basic performance since the position of its product name is located closely to "display" but far from words representing performance.

5.2 PDAs

In reputation mining for PDAs (personal digital assistants), we used both of the product name label and positive/negative nature label. We extracted characteristic words for each of the PDAs on the basis of learned association rules for connecting positive opinions to product names. The target category was defined as [product name = X, nature = positive].

We discovered that PDA A has a good reputation with respect to the use of email when away from home/work. We also discovered that its monochrome screen version is popular. Since the monochromatic quality is generally regarded as negative, this reputation is very interesting. A search of the original opinion data for the word "monochrome" indicated that the long running time, the reasonable price, and the large keyboard of the monochrome version are reasons for its good reputation. Figure 6 shows the obtained positioning map for four PDAs. Although all of the products have good reputations, they show almost no sharing of

Table 3: Typical Sentences in the Category [cellular phone A]

No.	Score	Typical Sentence
1	0.82	cellular phone A turned out to be fast. -Benchmark result–
2	0.70	The fact is that cellular phone A is faster. -Benchmark result amendment–
3	0.69	Comment to Mr. ***: I was able to boot up cellular phone A with no problem.
4	0.54	Since the price of cellular phone A Type 1 is falling, I will replace my *** with it as soon as possible, and replace that with cellular phone A Type 2 when the price comes down in the future.
5	0.51	I tested all of them, and found that the worst one (where the expression for "worst" in Japanese contains a word equivalent to the English "No.1".) was cellular phone A.
6	0.48	cellular phone A might be a candidate for purchase

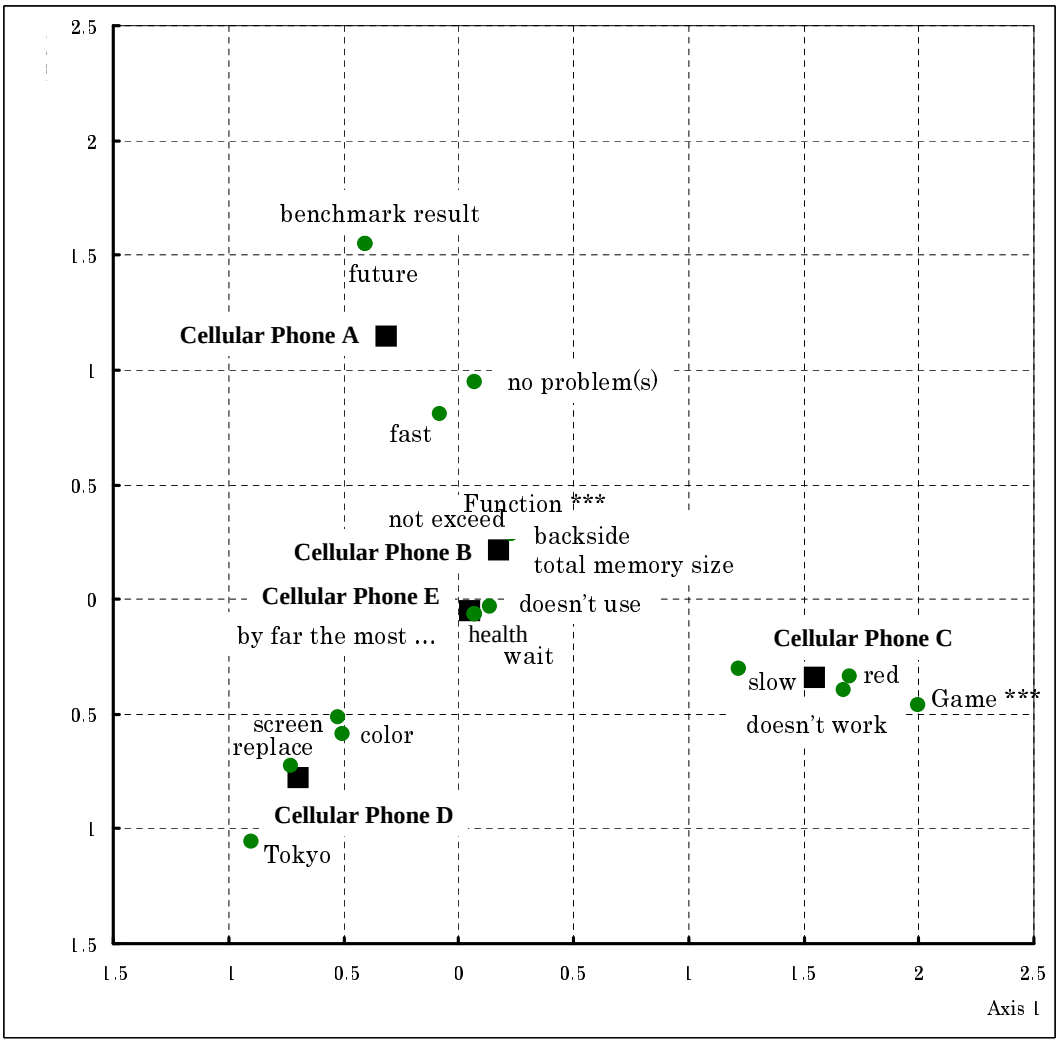

Figure 5: Positioning Map for Five Cellular phones and Their Extracted Characteristic Words

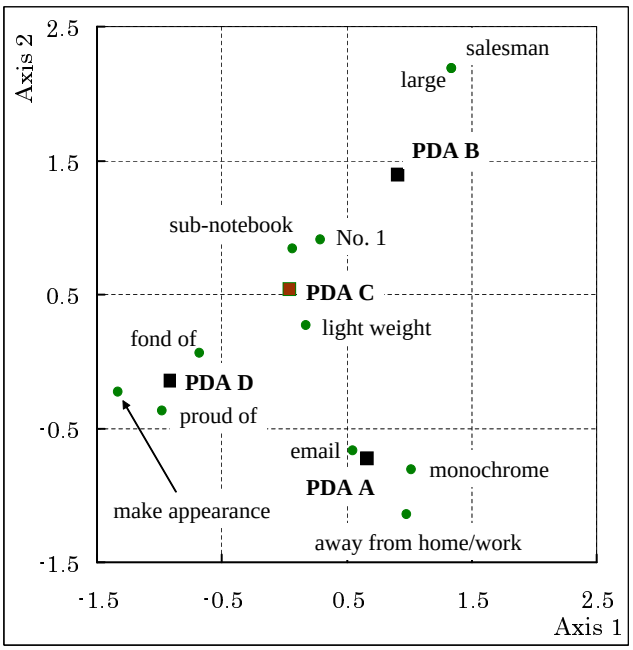

Figure 6: Positioning Map for PDAs and Characteristic Words

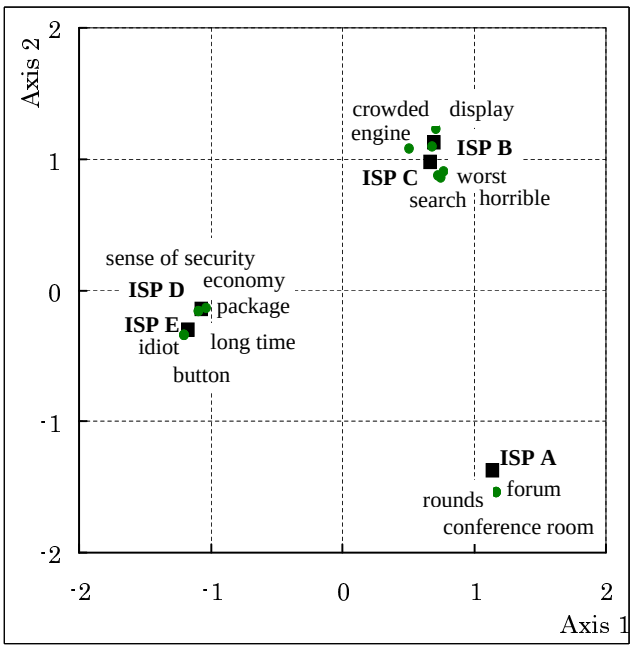

Figure 7: Positioning Map for ISPs and Characteristic Words

attribute-words, which implies that, in terms of extracted words, these products do not compete with each other.

5.3 Internet Service Providers

Fig 7 shows a positioning map five ISPs. As can be seen, they form three well-separated clusters. Providers within a single cluster have similar reputations. For example, two closely located ISPs, D and E, share reputations for both "economy" and "sense of security", and it is easy to imagine that they might compete each other.

6. CONCLUDING REMARKS

In this paper we have proposed a framework for mining product reputations on the web. It consists of an opinion extraction portion and a text mining portion; the former works as an application-specific question-answering system, and the latter conducts four fundamental tasks: characteristic word extraction, co-occurring word extraction, typical sentence extraction, and correspondence analysis. The key to combining these two parts is opinion labeling, which makes it possible to conduct supervised learning in the text mining portion. We used real data to empirically demonstrate that the proposed framework is able to help users discover significantly important knowledge regarding the reputations of products of interest, and to drastically reduce the cost of collecting and analyzing opinions. Our framework could, of course, also be well applied to mining reputations far beyond the area of industrial products, for example, individuals, events, services, companies, governments, etc.

7. REFERENCES

[1] B. Adelberg, Nodose - a tool for semi-automatically extracting structured and semistructured data from text documents, in *Proc. of the 1998 ACM SIGMOD International Conference on Management of Data*, pp:283-294, 1998.

[2] R. Agrawal and R. Srikant, Fast algorithms for mining association rules, in *Proc. 1994 Int'l. Conf. Very Large Data Bases (VLDB)*, pp:487–499, 1994.

[3] M.R. Anderberg, *Cluster Analysis for Applications*, Academic Press, 1973.

[4] N. Ashish and C. Knoblock, Wrapper generation for semi-structured internet sources, *SIGMOD Record*, 26(4), 1997.

[5] J.P. Benzecri, *Correspondence Analysis Handbook*, Mercel Dekker, 1992.

[6] V. Chaudhri and R. Fikes, Answering Systems, *the 1999 Fall Symposium. Technical Report, FS-98-04*, AAAI, November 1999.

[7] D. Clark, Shopbots Become Agents for Business Change, *Computer*, 33, pp:18–21, February 2000.

[8] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery, Learning to construct knowledge bases from World Wide Web, *Artificial Intelligence*, 118, pp:1–2, 2000.

[9] R. Doorenbos, O. Etzioni, and D. Weld, A scalable comparison-shopping agent for the World-Wide Web, in *Proc. of the First International Conference on Autonomous Agents Agents'97*, pp:39–48, 1997.

[10] D. Florescu, A. Levy, and A. Mendelzon, Database Techniques for the World-Wide Web: A Survey, *SIG-MOD Record*, 27(3), 1998.

[11] Fujitsu, Symfoware World http://www.fujitsu.co.jp/jp/soft/symfoware/index.html, 2001.

[12] S. Harabagiu, M. Pasca, and S. Maiorano,

Experiments with open-domain textual question answering, in *Proc. of COLING-2000*, pp:292-298, 2000.

[13] B. Katz, From sentence processing to information access on the World Wide Web. in *Natural Language Processing for the World Wide Web: the 1997 AAAI Spring Symposium*, pp:77–94, 1999.

[14] Komatsu Soft, Information Mining Tool VextSearch (in Japanese) http://www.komatsusoft.co.jp/develp/vxtsc/index.html, 2001.

[15] H. Li and K. Yamanishi, Mining from open answers in questionnaire data, in *Proc. of KDD 2001*, pp:443–449, 2001.

[16] H. Li and K. Yamanishi, Text classification using ESC-based stochastic decision lists, *Information Processing and Management*, 38, pp.343–361, 2002.

[17] K.C. Litkowski, Question-answering using semantic relation triples.in *Proc. of the 8th Text Retrieval Conference (TREC-8).*, pp:349–356, 1999.

[18] D. Moldovan and S. Harabagiu, The structure and performance of an open-domain question answering system, in *Proc. of the 38th Annual Meeting of the Association for Computational Linguistics*, pp:563–570, 2000.

[19] J. Prager, E. Brown, and A. Coden, Question-answering by predictive annotation, in *Proc. of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp:184–191, 2000.

[20] J.R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.

[21] J. Rissanen, Fisher information and stochastic complexity, *IEEE Transaction on Information Theory*, 42(1), pp:40–47, 1996.

[22] D. R. Radev, J. Prager, and V. Samn, The use of predictive annotation for question answering in *Proc. of the 8th Text Retrieval Conference (TREC-8)*, pp:399–411, 1999.

[23] R. Srihari and W. Li, Information extraction supported question answering, in *Proc. of the 8th Text Retrieval Conference (TREC-8)*, pp:185–196, 1999.

[24] K. Tateishi, Y. Ishiguro, and T. Fukushima, A reputation search engine that gathers people's opinions from the internet, (in Japanese) *Technical Report NL-144-11*, Information Processing Society of Japan, pp:75–82, 2001.

[25] E.M. Voorhees and D.M. Tice, Building a quesdtion answering test collection, in *Proc. of the 23rd Annual International ACM SIGIR Conference on Research and Development in Informtion Retrieval*, pp:200–207, 2000.

[26] K. Yamanishi, A learning criterion for stochastic rules, *Machine Learning*, 9, pp:165–203, 1992.

[27] K. Yamanishi, A decision-theoretic extension of stochastic complexity and its applications to learning, *IEEE Trans. on Infortmation Theory*, 44(4), pp:1424–1439, 1998.