C11

# MOpiS: A Multiple Opinion Summarizer

Fotis Kokkoras[1], Efstratia Lampridou[1], Konstantinos Ntonas[2],
and Ioannis Vlahavas[1]

[1] Department of Informatics, Aristotle University of Thessaloniki,
54 124 Thessaloniki, Greece
{kokkoras,elamprid,vlahavas}@csd.auth.gr
[2] University of Macedonia Library & Information Center,
Egnatias 156, 54 006 Thessaloniki, Greece
kntonas@gmail.com

**Abstract.** Product reviews written by on-line shoppers is a valuable source of information for potential new customers who desire to make an informed purchase decision. Manually processing quite a few dozens, or even hundreds, of reviews for a single product is tedious and time consuming. Although there exist mature and generic text summarization techniques, they are focused primarily on article type content and do not perform well on short and usually repetitive snippets of text found at on-line shops. In this paper, we propose MOpiS, a multiple opinion summarization algorithm that generates improved summaries of product reviews by taking into consideration metadata information that usually accompanies the on-line review text. We demonstrate the effectiveness of our approach with experimental results.

**Keywords:** text summarization, opinion mining, product reviews.

## 1 Introduction

The Web has changed the way people express their opinion. They can now easily discuss and express their views about everything, generating in this way huge amounts of on-line data. One particular on-line activity that generates such data is on-line shopping. Modern successful on-line shops and product comparison sites allow consumers to express their opinion on products and services they purchased. Although such information can be useful to other potential customers, reading and mentally processing quite a few dozens or hundreds of reviews for a single product are tedious and time consuming.

Mature text summarization systems can provide a shortened version of a (rather long) text, which contains the most important points of the original text [1]. These summaries are produced based on attributes (or features) that are usually derived empirically, by using statistical and/or computational linguistics methods. The values of these attributes are derived from the original text and the summaries typically have 10%-30% of the size of the original text [2].

Although there are text summarizers available that perform well on article type content, the discreteness of the on-line reviews suggests that alternative techniques

J. Darzentas et al. (Eds.): SETN 2008, LNAI 5138, pp. 110–122, 2008.

are required. On-line reviews are usually short and express only the subjective opinion of each reviewer. The power of these reviews lies behind their large number. As more and more reviews for a specific product or service are becoming available, possible real issues or weaknesses of it are revealed as they are reported by more users. The same holds for the strong features of it.

The problem with all these written opinions is that it takes much time for someone to consult them. Sometimes it is even impossible to read them all due to their large number. Mining and summarizing customer reviews is the recent trend in the, so called, research field of *opinion mining*. Unlike traditional summarization, opinion (or review) summarization mines the features of the product on which the customers have expressed their opinions and tries to identify whether the opinions are positive or negative [3, 4, 5]. It does not summarize the reviews by selecting a subset or rewriting some of the original sentences.

Popular on-line shops such as newegg.com or product comparison portals such as pricegrabber.com, contain already categorized reviews (with pros and cons) that contain additional metadata such as the familiarity of the user with the domain of the product, the duration of ownership at the time of the review, the usefulness of the review to other users, etc. Such augmented reviews can help us decide what is better to include in a summary. For example, they might provide hints for the reliability of the reviewer.

In this paper, we identify cases of such valuable metadata and propose MOpiS, a novel summarization approach for multiple, metadata augmented, product reviews. We work with the reviews at the sentence level. We first create a dictionary of the domain and then score the available sentences using a simple statistical method. We then utilise the available metadata of each review to increase or decrease this score in a weighted fashion. At the end we provide a redundancy elimination step to improve the quality of the summary produced.

We also present experimental results, which provide strong evidence for the validity of our claims. The summarization algorithm we propose outperforms two commercial, general purpose summarizers and a naive version of our approach that all ignore such metadata.

The rest of the paper is organized as follows: Section 2 presents related word, while Section 3 identifies potential metadata that can serve our approach and describes the way we collect all these data (review text and metadata). Our summarization algorithm is described in detail in Section 4, while Section 5 includes our experimental results and discussion about them. Finally, Section 6 concludes the paper and gives insight for future work.

## 2 Related Work

Most of the related research work in review summarization focuses on the problem of identifying important product features and classifying a review as positive or negative for the product or service under consideration.

Hu and Liou in [4] mine the features of the product on which the customers have expressed their opinions and decide whether the opinions are positive or negative. They do not summarize the reviews by selecting a subset neither rewrite some of the original sentences from the reviews to capture the main points, as in the classic text summarization.

Morinaga et al. [5, 6] collect statements regarding target products using a general search engine and then extract opinions from them, using syntactic and linguistic rules derived by human experts from text test samples. They then mine these opinions to find statistically meaningful information.

In [7], Dave et al. use information retrieval techniques for feature extraction and scoring. They use their system to identify and classify (as positive or negative) review sentences in web content.

Nguyen et al. in [8] classify the sentences of restaurant reviews into negative and positive and then categorize each sentence into predefined types, such as food and service. From each type, both a negative and a positive review are selected for the summary.

OPINE is an unsupervised information extraction system presented in [9], which extracts fine-grained features and opinions from on-line reviews. It uses a relaxation-labelling technique to determine the semantic orientation of potential opinion words, in the context of the extracted product features and specific review sentences.

None of the approaches mentioned above take advantage of the additional available metadata of each review to improve the efficiency of the task, whether they perform summarization or opinion categorization. To the best of our knowledge, our approach is novel and outlined as follows:

- We weigh the importance of the available metadata by using a multicriteria approach and use web content extraction and a simple statistical approach to build (once) a dictionary of the domain.
- We rank the sentences of multiple reviews on the basis of the frequency of their words and the dictionary, and then adjust their importance to some degree (weighted adjustment) by considering the available metadata.
- We select the sentences for the final summary eliminating redundancy at the same time.

# 3 Metadata Identification and Extraction

## 3.1 Metadata Identification

We have examined the review facilities provided by many popular on-line shops, for additional information that accompanies the review text and that can potentially contribute to a summarization task. The features we located are presented in Table 1.

Since the features in the list we built do not all exist in every e-shop, we focused on providing a usage methodology that can be followed even in the absence of any of these metadata. It is obvious though that, in such a case, some performance degradation is expected. Our main concern was to keep it graceful.

Note also that our approach is based on reviews that are already categorised by the reviewers, by providing separate positive and negative comments. As a result we discriminate between positive (*pros*) and negative (*cons*) reviews. If the reviews are not categorised, then an opinion categorization step is required. This is future work for our case. Finally, we consider parameters with calculated values (*Respectability* in

Table 1). Such metadata, if required, can be calculated by using web content extraction techniques.

**Table 1.** Useful and common metadata, accompanying product reviews in e-shops

| Field | Possible Values |
|---|---|
| Tech Level (of the reviewer) | *average*, *somewhat high*, *high* |
| Ownership Duration (of the product under review) | *a few days*, *about week, a few weeks, a few months, a year, more that a year* |
| Usefulness (of the review) | *"**n** out of **m** people found this review helpful"*<br>number of people (n) who vote this review useful out of the total number of people (m) who voted either for or against the review |
| Respectability (of the reviewer) | this is a calculated metadata: percentage value, equal to the average usefulness of all the reviews this user has made |

### 3.2 Metadata Extraction

Unfortunately, the data required for the summarization task usually resides in proprietary databases and is considered inaccessible for automated processing. The reviews are only available in HTML pages generated automatically from page templates and database content. The only way to gather arbitrary such semi-structured data is to use web content extraction techniques.

For the web data extraction task we developed ΔEiXTo [11], a general purpose, web content extraction tool which consists of two separate components:

- GUI ΔEiXTo a graphical application that is used to visually build, test, fine-tune, execute and maintain extraction rules (wrappers), and
- ΔEiXTo executor, an open source Perl application that screen scrapes desired web content based on extraction rules created with GUI ΔEiXTo.

Data extracted with ΔEiXTo can be saved in various formats, suitable for further processing, including XML and RSS. Additionally, both components can be easily scheduled to run automatically and extract desired content. Some kind of cooperative extraction (between two or more wrappers) is also possible with ΔEiXTo. The detailed presentation of ΔEiXTo is beyond the scope of this paper and will be done in the near future.

## 4 The MOpiS Summarization Algorithm

In this section we present MOpiS, the proposed **M**ultiple **Opi**nion **S**ummarization algorithm. MOpiS works at the sentence level. The available positive and negative comments from the reviews of a product are aggregated to form the positive and the negative sum, respectively. Then, each sum is partitioned into individual sentences from which we remove the stop words, the punctuation, the numbers and the symbols. As a result, a *Pros* and a *Cons* sentences set is produced.

Besides the review text, our approach uses an automatically generated dictionary, containing certain keywords related to the domain of the product in question. The dictionaries (one for each product category) are produced once by a Perl script that processes a large amount of reviews on products of the domain in question. The exact dictionary generation procedure is the following:

- Extract review data for 50 products using ΔEiXTo (we collected a few thousands reviews for each domain).
- For each domain, create a single text file containing the pros and cons part of the review data.
- Remove the stop words (articles, prepositions, pronouns), as well as 500 quite common English words.
- For each word calculate the frequency of occurrence and keep the 150 most frequent words.

Finally, we identify which of the metadata of Table 1 are present in our reviews and use ΔEiXTo to extract them. The extraction takes place in the same task that collects the review text (ΔEiXTo is capable of extracting many fields at the same time).

Thus, for each product *p* for which we want to summarize the reviews, our algorithm takes as input a set with the review data of *p* (either *Pros* or *Cons*), the dictionary *D* of the domain and $k \leq 4$ sets of metadata. The summarization algorithm is described next.

## 4.1 The Scoring Procedure

### 4.1.1 Text Contribution

The main concept of the scoring procedure is that each sentence should be given a score depending on the importance of the words that it contains, but also on the additional metadata of the review that it belongs to. For each sentence *i*, we calculate an initial score $R_i$ based on the text and then adjust this score according to the metadata presented. This is expressed with equation (1) in which $w_j$ is a factor which defines the importance we give to this metadata category.

$$S_i = R_i + R_i \cdot \sum_{j=1}^{k} w_j \qquad (1)$$

For each sentence, the $R_i$ parameter in equation (1) is calculated on the basis of the importance of the words the sentence contains. Each word $v_l$ of the sentence contributes to the score its frequency of occurrence $f_{v_l}$, unless this word belongs to the dictionary *D*, in which case its contribution is doubled. This is depicted in equation (2).

By doubling the contribution of dictionary words to the initial score of a sentence, we increase the probability to have this sentence in the final summary, as the more dictionary words it contains the more important it is considered.

$$R_i = \sum_{v_l \notin D} f_{v_l} + 2 \cdot \sum_{v_l \in D} f_{v_l} \qquad (2)$$

### 4.2 Metadata Contribution

Let us now define the way the metadata of each review contribute to the total score $S_i$ (equation 1) of each sentence. Since the various metadata fields are of different nature, those considered more important should contribute more to $S_i$, that is, their $w$ value should be greater. For the task of assigning proper values to the factors $w_j$, we used the Analytic Hierarchy Process (AHP [10]), which provides a methodology to estimate consistent weight values for criteria, according to the subjective importance we give to these criteria. This importance values are selected from a predefined (by AHP) scale between 1 and 9.

Particularly and according to [10], we considered:

- the ownership duration to be "very little more important" than the technology level of the user (importance 2 in AHP),
- the usefulness of the review to be "a little more important" than the ownership duration (importance 3 in AHP), "more important" than the technology level of the reviewer (importance 4 in AHP) and "very little more important" than the respectability of the reviewer (importance 2 in AHP),
- the respectability of the reviewer to be "very little more important" than the duration of ownership (importance 2 in AHP) and "a little more important" than the technology level of the reviewer (importance 3 in AHP).

With these considerations we were able to define the pairwise comparison matrix required by the AHP for the calculation of initial weight values $w'_j$. This is a 4x4 matrix if all four possible metadata categories of **Table 1** are used, 3x3 if one is omitted, etc. We also calculated the consistency criterion, as described in AHP. This metric provides evidence that we made no contradicting assumptions on the importance we assigned to the metadata categories.

In each case, the calculated $w'_j$ provides good initial values for the $w_j$ of equation (1). To provide further flexibility based on the values of the metadata category under consideration, we allow the replacement of $w'_j$ with a function of $g(d, w'_j)$, where $d$ is some function of the metadata value in hand.

For example, say that $w_1$ corresponds to tech level and we wish to give more credit to reviews from users of high tech level (a rational decision). We can define function $g(high, w'_1)$ as in equation 3.

$$w_1 = \begin{cases} w_1^{'} & TechLevel = high \\ 0 & otherwise \end{cases} \tag{3}$$

### 4.3 Redundancy Elimination

When the sentence scoring is over, *MOpiS* enters into its final step which is the elimination of redundant sentences. This step tries to prevent the inclusion of many sentences that have the same meaning with sentences that are already into the final summary.

First, the sentence $S_i$ with the highest rank is chosen. However, if the sentence is quite long (we used a threshold of 30 words) it is rejected and the next sentence is chosen. This rejection arises due to our observation that very long sentences were somehow artificially lengthy, because the reviewer did not obeyed common syntactic rules.

When the sentence with the highest score $S_u$ is selected, it is removed from the ranked list and is added to the final summary. At the same time the score $S_i$ of all of the rest sentences ($i \neq u$) in the ranked list is readjusted according to equation (4):

$$S_i' = S_i - \sum_{\forall v_l \notin D} f_{v_l} - 2 \cdot \sum_{\forall v_l \in D} f_{v_l} \qquad i \neq u \tag{4}$$

In equation (4), $v_l$ is a word of sentence $S_u$ which is already selected for the summary, and $f_{v_l}$ is its initially calculated frequency of occurrence. The rest of the symbols are as defined in equation (2).

Actually, the score of each of the rest sentences is decreased for every word that has been given bonus before, but now already appears in the summarization text. Thus, the recurrence of concepts in the summarization text is reduced.

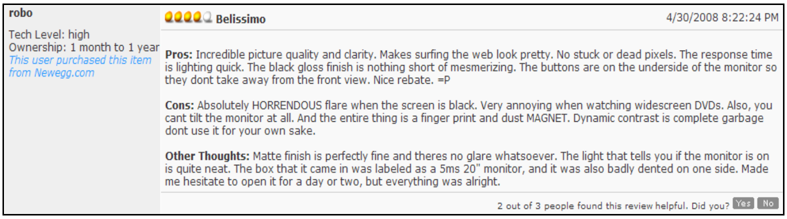
This selection-readjustment procedure is repeated for the next sentence in the top of the ranked list until the desired number of sentences is added into the summary.

The whole task described in Section 4 is performed once for the *pros* summary and a second time for the *cons* summary. The only difference in these two "runs" is the initial set of sentences.

# 5 Experimental Results

## 5.1 Case Study A

We extracted 1587 review records for 9 different products belonging to 3 different product categories (3 randomly selected products from each category) from newegg.com, one of the most successful on-line stores, where each review is organized in the way presented in Fig. 1.



**robo**

Tech Level: high
Ownership: 1 month to 1 year
*This user purchased this item from Newegg.com*

**Belissimo** 4/30/2008 8:22:24 PM

**Pros:** Incredible picture quality and clarity. Makes surfing the web look pretty. No stuck or dead pixels. The response time is lighting quick. The black gloss finish is nothing short of mesmerizing. The buttons are on the underside of the monitor so they dont take away from the front view. Nice rebate. =P

**Cons:** Absolutely HORRENDOUS flare when the screen is black. Very annoying when watching widescreen DVDs. Also, you cant tilt the monitor at all. And the entire thing is a finger print and dust MAGNET. Dynamic contrast is complete garbage dont use it for your own sake.

**Other Thoughts:** Matte finish is perfectly fine and theres no glare whatsoever. The light that tells you if the monitor is on is quite neat. The box that it came in was labeled as a 5ms 20" monitor, and it was also badly dented on one side. Made me hesitate to open it for a day or two, but everything was alright.

2 out of 3 people found this review helpful. Did you?

**Fig. 1.** A typical review record at newegg.com

We used a single extraction rule capable of performing a sequence of page fetches (by following "Next Page" links) and capturing all reviews and data fields under interest. A total of 160 web pages were processed. The amount of the extracted data is summarized in Table 2.

**Table 2.** The dataset used

| **Domain:** | Monitors | | | Printers | | | CPU Coolers | | |
|---|---|---|---|---|---|---|---|---|---|
| **Models:** | A | B | C | A | B | C | A | B | C |
| **#Reviews:** | 218 | 130 | 358 | 124 | 86 | 86 | 293 | 126 | 166 |

In particular, each review contains positive comments (*pros*), negative comments (*cons*), how familiar is the user with the related technology (*tech level*), the duration of ownership of the product (*ownership*) and the *usefulness* of the review to other users.

Using AHP and the importance values we discussed in Section 4.2, we calculated the following initial values: $w'_1$=0.14, $w'_2$=0.24 and $w'_3$=0.62 ($w'_1$ for *Tech Level*, $w'_2$ for *Ownership Duration* and $w'_3$ the *Usefulness* of the review).

We further adjusted $w_j$ using equation (3) for $w_1$, equation (5) for $w_2$ and equation (6) for $w_3$.

$$w_2 = \begin{cases} w_2' & Ownership = more\,than\,a\,year \\ 0 & otherwise \end{cases} \tag{5}$$

$$w_3 = g(\delta_v, w_3') = \left( \frac{1}{1+e^{-0.2\bullet\delta_v}} - \frac{w_3'}{1.24} \right) \cdot 1.24 \tag{6}$$

In regard to the factor $w_3$ of the usefulness of each review, a sigmoid function was used (equation (6)) to adjust $w'_3$ according to the difference $\delta_v$ between the positive and negative votes of a review. This favors reviews that were found useful by most users and penalizes reviews that were not considered useful by the majority of users.

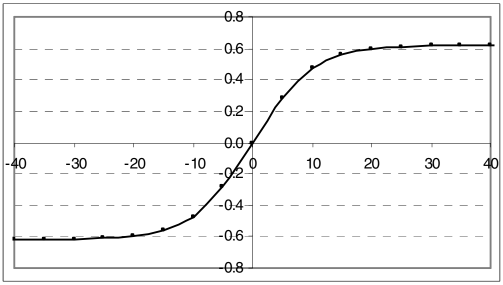


**Fig. 2.** The sigmoid function $g(\delta_v, w'_3)$ that modulates the factor $w_3$ according to $\delta_v$

The rest parameters of equation (6) were decided on the need to vary $w_3$ between $w'_3$ and -$w'_3$ (the value calculated with AHP) and move the plateau of $g$ away from values of $|\delta_v|<20$, because we observed that the majority of $\delta_v$ values lies in this range. Fig. 2 displays the way $w_3$ depends on $\delta_v$, through $g(\delta_v, w'_3)$.

As mentioned in Section 4.2, we used the AHP because it provides a methodology to check the consistency of the subjective importance values we assigned to each of the criteria. We applied this methodology to the importance values we assigned and found them to be consistent.

Besides *MOpiS*, we also used two well known, commercial summarizers, the *Copernic* [12] and the *TextAnalyst* [13]. Both are general purpose summarizers. This means that they work better with lengthy, article style texts. Reviews on the other hand are usually not so lengthy, they are many and some of them have almost the same meaning. Additionally we calculated how the *MOpiS* algorithm performed when we ignore the second addendum of equation (1), that is, ignore the metadata contribution – we call this version *naive MOpiS*.

*Copernic* produces document summaries by detecting the concepts of the text and then extracting sentences that reflect these concepts. It mostly uses statistical methods to identify the concepts. Additional important words cannot be inserted by the user, as the concepts extracted are considered to be the keywords required.

*TextAnalyst* can analyze unstructured text and create a semantic network from it. The semantic network is utilized to score the individual sentences. The system collects those sentences that have a semantic weight greater than a certain adjustable threshold value. It is possible to define an external dictionary of concepts but early tests with the dictionaries we had created led to reduced performance. Therefore no dictionary was set for *TextAnalyst*.

The results of our experiments are summarized in Table 3. We adjusted all systems so as to create a summary of 10 sentences for *pros* and 10 sentences for *cons*. The numbers in parenthesis are the performance of *naive MOpiS*. It is obvious that inclusion of metadata information in the way we suggested, improves the summary (to a degree of about 16% in our experiments), confirming our initial hypothesis. Precision and recall measures are average values that were calculated on the basis of three human-generated summaries. These individuals were provided only with the text of the reviews (without the additional metadata) and the variation in their judgment was les than 3.1%.

It is also obvious that the other two summarizers, although quite sophisticated without any doubt, do not perform very well with this kind of data (many short reviews with overlapped information).

Regarding *TextAnalyst*, the blank cells at recall and precision in Table 3 are due to our inability to adjust the system so as to produce summary of the desired length. In those cases, the summary contained either too many or too few sentences. Consequently, it was not comparable with the summary of *MOpiS* and *Copernic*.

Further investigation of the resulted summaries revealed some interesting facts. The most recent reviews for monitor B were from customers that owned the product more than a year. All of them complained about severe malfunctions after one year of possession (this was also the warranty period). Moreover, it was said that when warranty was over, service was no longer provided by the company. Although such facts were not reported by the majority of the reviewers, these two aspects were depicted in our summary, as they came from reviews with long duration of ownership that were subsidized by our algorithm. They were not mentioned though by neither *Copernic* and *TextAnalyst* nor the *naive MOpiS*.

The contribution of the usefulness of a review is also distinct. By increasing the score of a sentence belonging to a useful review and decreasing it in the opposite case

**Table 3.** Experimental Results for newegg.com

| | | | **MOpiS (naive MOpiS)** | | **Copernic** | | **TextAnalyst** | |
|---|---|---|---|---|---|---|---|---|
| | | | **Recall** | **Precision** | **Rec** | **Prec** | **Rec** | **Prec** |
| **Monitors** | **A** | **Pros** | 90.9 (90.9) | 70 (70) | 60 | 60 | 45.3 | 30 |
| | | **Cons** | 75 (62.5) | 70 (50) | 25 | 30 | 62.5 | 60 |
| | **B** | **Pros** | 100 (77.8) | 90 (80) | 100 | 60 | 66.7 | 70 |
| | | **Cons** | 88.8 (66.7) | 70 (60) | 75 | 60 | 33.3 | 70 |
| | **C** | **Pros** | 100 (100) | 90 (80) | 72.7 | 60 | - | - |
| | | **Cons** | 88.9 (66.7) | 80 (60) | 60 | 40 | - | - |
| **Printers** | **A** | **Pros** | 85.7 (85.7) | 70 (70) | 62.5 | 40 | - | - |
| | | **Cons** | 87.5 (62.5) | 60 (40) | 50 | 40 | 50 | 40 |
| | **B** | **Pros** | 100 (100) | 60 (40) | 83.3 | 60 | 66.7 | 40 |
| | | **Cons** | 87.5 (37.5) | 70 (40) | 75 | 60 | 50 | 70 |
| | **C** | **Pros** | 87.5 (75) | 80 (70) | 87.5 | 60 | 62.5 | 50 |
| | | **Cons** | 100 (100) | 70 (70) | 71.4 | 70 | 50 | 60 |
| **CPU Coolers** | **A** | **Pros** | 100 (100) | 70 (60) | 66.7 | 70 | - | - |
| | | **Cons** | 100 (80) | 80 (70) | 60 | 60 | 60 | 62.5 |
| | **B** | **Pros** | 83.3 (83.3) | 100 (100) | 66.7 | 80 | 50 | 90 |
| | | **Cons** | 100 (75) | 70 (60) | 60 | 60 | 25 | 10 |
| | **C** | **Pros** | 75 (75) | 70 (50) | 75 | 70 | - | - |
| | | **Cons** | 100 (80) | 50 (60) | 100 | 70 | 80 | 40 |
| **Average:** | | | **91.7 (78.8)** | **73.3 (62.8)** | **69.5** | **58.3** | **54** | **53.3** |

(equation (6)), significant sentences were kept in the summary while those with no importance were excluded. Because of that, the occurrence of false information in the summary due to malicious reviews is highly unlikely, as those reviews get negative votes of usefulness by the other users. For instance, the following review from monitor A gathered 21 negative votes and 0 positive for being useful:

*Monitor had a sticker on it "Certified for Windows Premium", but when I tried to install the software it said "This software does not work with Vista". I phoned <company> - they refused to send me replacement software that will work with Vista!*

This sentence was selected by *TextAnalyst* as, despite its meaning, it contains important words. *MOpiS* decreased its score by setting $w_3$=-0.60 in equation (1). Similarly, *naive MOpiS* selected a sentence from an abusive review that was voted down by the users. None such sentence was selected by *MOpiS*, resulting in summary of better quality.

On the contrary, reviews that received many positive votes are considered more useful and likely to hold important information, so their sentences are given precedence. This is also a way of not depending exclusively on statistical methods, because important statements may not have a high word frequency.

For example, in printer A, there were reviews complaining about the printer being reset in Japanese. Human summarization can easily identify this as a negative aspect in spite of the low frequency (it was not mentioned by many reviews). *MOpiS'* summary reported it though, because of the high usefulness of the reviews. None of the other systems tracked it down.

Moreover, the redundancy elimination aspect of *MOpiS* performed well. Repetition on concepts was minimal or absent since it does not select the highly rated sentences but readjusts the score of all the rest sentences according to the one that was selected for the summary. In *TextAnalyst* however, the repetition of the concepts presented was evident. Actually, in one case, its summary included two identical sentences, coming from a review that was submitted twice! Redundancy elimination also helped *naive MOpiS* to outperform the two commercial summarizers.

It was also observed that, the special nature of the review data affects the performance of plain text summarizers like *Copernic*. Although its function is based on statistical methods, its results were affected to a great degree by the structure of the text. When the same data (aggregated reviews) had different order, different summary was generated.

Finally, we used *MOpiS* in another summarization task worth mentioned. In one case, we were asked to verify if there were problems reported regarding the operation of a RAID controller in a certain computer motherboard under a certain operating system. We summarized 142 negative comments (cons) and this "rumor" was reflected in the summary. Neither *Copernic* nor *TextAnalyst* verified it though.

## 5.1 Case Study B

We conducted another experiment, this time on a different web site, the pricegrabber.com. This site does not contain the amount of reviews of newegg.com, since it directs the buyer to retailer e-shops for the final transaction – it seems the buyers prefer to review the product at the retailer's site. We extracted data for two printers A and B (27 and 33 records respectively).

Reviews in pricegrabber.com are quite short and contain *strengths* and *weaknesses* instead of *pros* and *cons*. They do not include the *tech level* of the reviewer but provide access to other reviews of the same person. As a result, we decided to create a calculated *Respectability* value by averaging the usefulness values of his reviews. The initial $w_j$ factors were calculated as: $w'_1$= 0.16 (for ownership), $w'_2$= 0.3 (for respectability) and $w'_3$= 0.54 (for usefulness). We further adjusted those values like we did in case study A. We used equation (5) for *ownership*, a sigmoid function like equation (6) for *usefulness* and $w_3=0.006{*}\mathit{respectability}\text{-}w'_3$ for *respectability*.

Due to the small length of the reviews, some sights of saturation were observed. *Copernic* and *TextAnalyst* combine sentences to create new. As a result, they packed

**Table 4.** Experimental results for pricegrabber.com

| | | | **MOpiS (naive MOpiS)** | | **Copernic** | | **TextAnalyst** | |
|---|---|---|---|---|---|---|---|---|
| | | | **Recall** | **Precision** | **Rec** | **Prec** | **Rec** | **Prec** |
| **Printers** | **A** | **Pros** | 100 (100) | 90 (60) | 100 | 80 | 83.3 | 90 |
| | | **Cons** | 100 (100) | 80 (70) | 100 | 60 | - | - |
| | **B** | **Pros** | 100 (100) | 90 (60) | 100 | 70 | - | - |
| | | **Cons** | 83.3 (83.3) | 70 (70) | 100 | 70 | 100 | 80 |
| **Average:** | | | **95.8 (95.8)** | **82.5 (65)** | **100** | **58.3** | **91.65** | **85** |

many small reviews into long sentences, including in this way almost all the initial reviews. They couldn't prevent though the repetition of the same fact many times in their reviews. Repetition was minimal in *MOpiS* which also adapted well to the different kind of metadata of pricegrabber.com.

## 6 Conclusions

In this paper, we proposed a novel, multi review/opinion summarization algorithm that is based not only on the text of the review but on additional review metadata. We detected four such frequently found metadata and based on AHP, we consistently defined how important we consider them in a useful review, in relation to each other. We used these importance values to define weights that control the way these metadata contribute to our review scoring procedure.

The additive nature of our algorithm allows it to adapt to review sites with any subset of the set of metadata we detected. Moreover, we allow custom modulation of the initial calculated weight to give bonus or penalize certain values for the metadata field of the review. Finally, the redundancy elimination step reduces concept repetition in the final summary.

Our experimental results demonstrated the usefulness of this metadata inclusion by means of improved precision and recall metrics. This is clearly demonstrated by the improved performance of *MOpiS* compared to *naive MOpiS*.

Consequently, our next step is to remove the requirement for categorized reviews (pros/strengths and cons/weaknesses) since there exist many sites which do not discriminate between pros and cons, but rather have a single, mixed review.

A limited version of *MOpiS* (3 product categories from newegg.com), is available online at http://deixto.csd.auth.gr/newegg/newegg.html, for real-time demonstration.

## References

1. Mani, I.: Automatic Summarization. John Benjamins Publishing Company, Amsterdam (2001)
2. Mani, I., Maybury, M.T.: Advances in Automatic Text Summarization. MIT Press, Cambridge (1999)
3. Liu, B.: Web Data Mining. Springer, Heidelberg (2007)
4. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, SIGKDD 2004, pp. 168–177 (2004)
5. Hu, M., Liu, B.: Mining Opinion Features in Customer Reviews. In: Proceedings of the 19th National Conference on Artificial Intelligence (AAAI 2004), San Jose, USA (2004)
6. Morinaga, S., Yamanishi, K., Tateishi, K., Fukushima, T.: Mining Product Reputations on the Web. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discover and Data Mining, KDD 2002, pp. 341–349 (2002)
7. Dave, K., Lawrence, S., Pennock, D.N.: Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews. In: Proceedings of the 12th International World Wide Web Conference, WWW 2003, pp. 451–460 (2003)

8. Nguyen, P., Mahajan, M., Zweig, G.: Summarization of Multiple User Reviews in the Restaurant Domain. Technical Report, Microsoft Research, MSR-TR-2007-126 (2007)
9. Popescu, A.M., Etzioni, O.: Extracting product features and opinions from reviews. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, Vancouver, Canada, pp. 339–346 (2005)
10. Saaty, T.L.: Decision Making for Leaders: The Analytic Hierarchy Process for Decisions in a Complex World. RWS Publications, Pittsburgh (1999)
11. ΔEiXTo web data extraction tool, `http://deixto.csd.auth.gr`
12. Copernic Summarizer, `http://www.copernic.com`
13. TextAnalyst, `http://www.megaputer.com/textanalyst.php`