Summarization of Multiple, Metadata Rich, Product Reviews

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

Abstract. 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 even hundreds of reviews for a single product is tedious and time consuming.

In this paper, we propose *ReSum* a novel summarization approach for multiple, metadata augmented, product reviews. We argue that the contribution of additional information (metadata) such as the user's expertise, the usefulness of the review to other users, etc., is significant and can result in improved summaries. The summarization algorithm we propose outperforms two commercial, general purpose summarizers that ignore such metadata.

1 INTRODUCTION

Product reviews written by on-line shoppers is a valuable source of information for potential new customers of these products, who desire to make an informed purchase decision. Popular, high profile on-line shops such as newegg.com or product comparison portals such as pricegrabber.com contain categorized reviews (pros and cons) that are attributed with additional metadata such as the level of familiarity of the user with the domain of the product, the time of ownership and the usefulness of the review to other users.

Text summarization systems generate a summary of the original text that allows the user to obtain the main pieces of information available in that text, but with a much shorter reading time [1]. The 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 the text summarization research field is quite old and there exist commercially available text summarizers, the discreteness of the on-line reviews suggests that alternative techniques are required. These techniques lay in the field of opinion mining and besides being challenging due to the inherent difficulties in natural language processing, they are also very useful in practice to potential new customers, directed advertisement, etc. [3].

On-line reviews are usually short and convey only the subjective opinion of each reviewer. The power of these reviews lay 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 elevated as they are evidenced by more users. The same holds for the strong features of it.

The problem with all these written opinions is that it takes time for someone to consult them. Sometimes it is even impossible to read them all due to their large number. Thus, summarization techniques are required, specialized for that kind of metadata rich, textual reviews.

[1] Department of Informatics, Aristotle University of Thessaloniki, Greece, email: kokkoras@csd.auth.gr

In this paper we propose *ReSum*, a multi review summarization algorithm for on-line, metadata rich, product reviews. We also present preliminary, experimental results, which provide strong evidence for the validity of our claims.

The rest of the paper is organized as follows: Section 2 presents related word, while Section 3 describes our approach to the non-trivial task of collecting all these on-line reviews. Our summarization algorithm is described 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 whether the opinions are positive or negative. They do not summarize the reviews by selecting a subset or rewrite some of the original sentences to capture the main points, as in the classic text summarization.

Morinaga et al. in [5], 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 [6], Dave et al. use information retrieval techniques to identify product attributes (feature extraction) and train a classifier using a corpus of self-tagged reviews available from web sites. They then refine their classifier using the same corpus, before applying it to sentences mined from broad web searches. Their aim is to determine whether reviews are positive or negative.

Nguyen et al. in [7], 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 [8], which extracts fine-grained features and associated opinions from on-line reviews. It uses a relaxation-labeling technique to determine the semantic orientation of potential opinion words, in the context of the extracted product features and specific review sentences.

Our work mainly differentiates in that, it is focused on exploiting additional available metadata regarding each review, rather than classifying a review as being positive or negative. We use web content extraction and a simple statistical approach to build a dictionary of the domain, rank sentences of multiple reviews on the basis of this dictionary and then adjust their importance by considering features such as the familiarity of the user with the domain,

product possession duration and the usefulness of each review to other users. This latter task greatly improves the resulted summary.

3 DATA 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 such unstructured or semi-structured data is to use web content extraction techniques.

3.1 Data Source

In our experiments, we used data from a well known online shop, the newegg.com, where customer reviews are available in dedicated web pages and each review is organized in the way presented in Figure **1**.

Figure **1**. A typical review record in newegg.com

In particular, each review contains positive comments (*Pros*), negative comments (*Cons*), how familiar is the user with the related technology (*Tech Level*), the time of ownership of the product (*Ownership*) and the usefulness of the review to other users. The possible values for these fields are given in Table **1**.

Table **1**. Possible content of reviews in newegg.com

Field	Possible Values
Pros, Cons	free text
Tech Level	any of: *average*, *somewhat high*, *high*
Time of Ownership	any of: *1 day to 1 week*, *1 week to 1 month*, *1 month to 1 year*, *more than a year*
Usefulness	*"**n** out of **m** people found this review helpful"* 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

3.2 ΔEiXTo: a Web Data Extraction Tool

For the content extraction task we developed ΔEiXTo [9], a general purpose, web content extraction tool which consists of two separate applications:

- GUI ΔEiXTo (Figure **2**) a graphical application that is used to visually build, test, fine-tune and maintain extraction rules, 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, including XML and RSS. Additionally, both modules can be easily scheduled to run periodically and extract specified content. The detailed presentation of ΔEiXTo is beyond the scope of this paper and will be done in the near future.

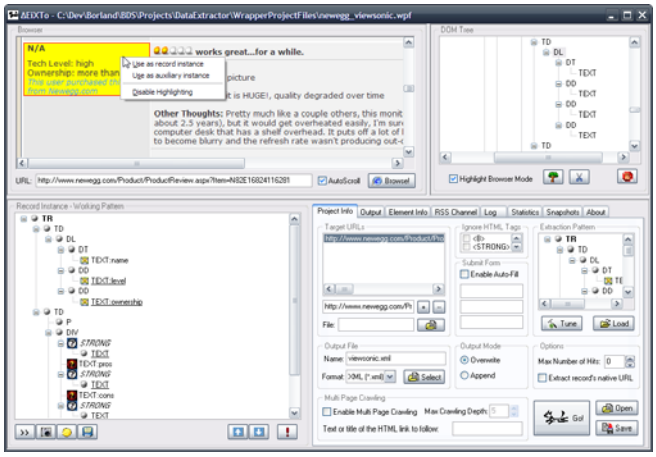

Figure **2**. ΔEiXTo (the GUI version)

3.3 Dataset and Dictionary Preparation

We extracted 1587 review records for 9 different products belonging to 3 different product categories (3 randomly selected products from each category). 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 field of 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

Apart from the review data, our approach uses an automatically generated dictionary, containing certain keywords related to the domain of the product. These three 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 the same extraction rule (we collected about 4000 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 more common English words.
- Calculate the frequency of occurrence of each word and keep the 150 most frequent words.

Thus, for each product for which we want to summarize the reviews, our algorithm takes as input a set of review data for the product and the dictionary *D* of the domain of the product. The summarization algorithm is described next.

4 THE SUMMARIZATION ALGORITHM

ReSum, our product review summarization algorithm, processes the positive (*Pros*) and the negative (*Cons*) reviews separately and creates a "pros" and a "cons" summary, respectively. First, it splits each review into sentences and removes from each the stop words, the punctuation, the numbers and the symbols. Then, the frequency of occurrence f_{v_j} of each word v_j in the corpus is calculated and each sentence s_i is scored with the procedure described next.

4.1 Contribution of the Review Text

The main concept of the scoring procedure is that each sentence s_i should be given a score S_i depending on the importance of the words that it contains, but also on the additional attributes of the review that it belongs to. For each sentence s_i, *ReSum* calculates an initial score R_i based on the words it contains and then adjusts this score according to the *Tech Level*, the *Time of Ownership* and the *Usefulness* of the review this sentence belongs to. This is mathematically expressed with equation (1) in which w_k is a factor which defines the importance we give to the additional information available for each review (w_1 for *Tech Level*, w_2 for *Time of Ownership* and w_3 the *Usefulness* of the review).

$$S_i = R_i + R_i \cdot \sum_{k=1}^{3} w_k \tag{1}$$

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

$$R_i = \sum_{v_j \notin D} f_{v_j} + 2 \cdot \sum_{v_j \in D} f_{v_j} \tag{2}$$

By doubling the contribution of dictionary words to the score of a sentence, we increase the probability to have this sentence in the final summary.

4.2 Contribution of the Review's Metadata

Overall, the scoring equation (1) is of multi-criteria nature. The important aspect of it is the assignment of proper values to the factors w_k. For this task, we used the Analytic Hierarchy Process (AHP) [10], which provides a methodology to calculate consistent weight values for criteria, according to the subjective importance we assign to these criteria. The importance value is selected from *the fundamental scale for pairwise comparison of the criteria* [10], which ranges between 1 and 9. Particularly, we set:

- ownership duration is *"very little more important"* than the technology level of the user (importance 2 in the fundamental scale of the AHP),
- the usefulness of the review is *"a little more important"* than the time of ownership (importance 3 in the AHP)
- the usefulness of the review is *"more important"* than the technology level of the user (importance 4 in the AHP).

With the above settings, we were able to define the pairwise comparison matrix required by the AHP for the calculation of weight values for our three criteria: *Tech Level*, *Time of Ownership* and *Usefulness* of the review.

The values calculated were w'_1=0.14, w'_2=0.24 and w'_3=0.62. Based on these values, we defined w_k of equation (1) as follows:

$$w_1 = \begin{cases} w'_1 & TechLevel = high \\ 0 & otherwise \end{cases} \tag{3}$$

$$w_2 = \begin{cases} w'_2 & Ownership = more\,than\,a\,year \\ 0 & otherwise \end{cases} \tag{4}$$

In regard to the factor w_3 of the usefulness of each review, a sigmoid function was used (equation (5)) to adjust w'_3 according to the algebraic difference δ_v between the positive and negative votes (raw numbers) of the review. This step is required to favor reviews that were found useful by more users and penalize reviews that were considered not useful by the majority of users.

$$w_3 = \Phi(\delta_v) = \left(\frac{1}{1 + e^{-0.2 \bullet \delta_v}} - 0.5 \right) \cdot 1.24 \tag{5}$$

The rest parameters of equation (5) 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 Φ away from values of δ_v<20, because we observed that there lies the majority of δ_v values. Figure **3** displays the way w_3 is depended on δ_v, through $\Phi(\delta_v)$.

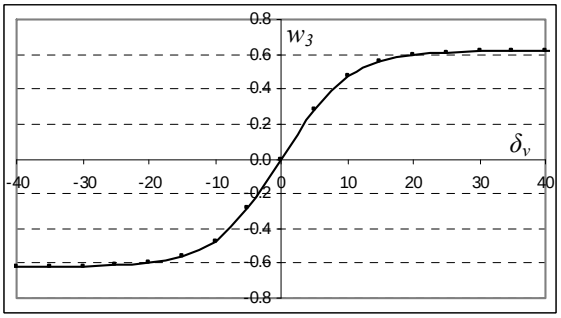

Figure **3**. The sigmoid function $\Phi(\delta_v)$ that modulates the factor w_3.

It should be noted here that we used the AHP because it provides a methodology to check the consistency of the subjective importance values we assigned to each of the three criteria. We applied this methodology to the importance values we set and found them to be consistent.

4.3 Redundancy Elimination

When the sentence scoring is over, *ReSum* enters into its final step which is the elimination of redundant sentences. This step tries to prevent the inclusion of many sentences that convey the same meaning with sentences that are already in to 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. Due to the additive nature of equation (1), very long sentences tend to get very high score and this could be a feature for exploitation.

When the sentence with the highest score is selected, it is removed from the ranked list and is added to the final summary. At the same time the score of all the rest sentences in the ranked list is readjusted according to equation (6):

$$R_i^{'} = R_i - \sum_{v_j \notin D} f_{v_j} - 2 \cdot \sum_{v_j \in D} f_{v_j} \quad (6)$$

where R'_i is the adjusted score, R_i is the initial score, v_j are the words of sentence s_i which is already selected for the summary and f_{v_j} is their 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, yet now already appears in the summarization text. In this way, the recurrence of concepts in the summarization text is reduced.

This procedure is repeated for the next sentence in the top of the ranked list until the desirable number of sentences is incorporated in the summary.

5 EXPERIMENTAL RESULTS

Apart from *ReSum*, we also used two well known, commercial summarizes, the *Copernic* [11] and the *TextAnalyst* [12]. 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, there are many of them and they usually overlap in the conveyed message.

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 upon it. The semantic network is utilized to score the individual sentences and the system collect those sentences that have a semantic weight higher that a certain adjustable threshold value. It is possible to define an external dictionary of concepts but early tests with the dictionaries we created led to reduced performance. Therefore no dictionary was set for *TextAnalyst*.

The results of our experiments are summarized in Table **3**. 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%.

We adjusted all systems so as to create a summary of 10 sentences for *pros* and 10 sentences for *cons*. The number in parenthesis in the *ReSum* columns was calculated by ignoring the contribution of the additional metadata (that is, ignoring the second addendum of equation (1) – we call this version *naive ReSum*). It is obvious that inclusion of this information in the way we suggested, improves the summary (to a degree of about 16% in our experiments), testifying our initial hypothesis.

It is also obvious that the other two summarizers, although quite sophisticated with no doubt, are not performed very well in this kind of data (many sort reviews with overlapped information).

Table **3**. Experimental Results

			ReSum		**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**

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, so as to not being comparable with the summary of *ReSum* and *Copernic*.

Further investigation of the resulted summaries unveiled some interesting facts. The most recent reviews for monitor B were from costumers 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 is 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 imprinted in our summary, as they came from reviews with long duration of ownership that subsidized by our algorithm. They were not mentioned though by neither *Copernic* and *TextAnalyst* nor the *naive ReSum*.

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 (equation (5)), significant sentences were kept in the summary while those with no importance were excluded. Owning to that, the appearance in the summary of untrue information coming from malign reviews is highly unlikely, as they 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. *ReSum* decreased its score by setting w_3= -0.60 in equation (1). Similarly, *naive ReSum* selected a

sentence from an abusive review that was voted down by the users. None such sentence was selected by *ReSum*, resulting in summary of better quality.

On the contrary, reviews that received many positive votes are considered more useful and are candidates that possibly hold important information, so their sentences are given precedence. This is also a way of not depending exclusively on statistical methods, as substantial 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). *ReSum*'s summary reported it though, because of the high usefulness of the reviews. None of the other systems tracked it down.

The redundancy elimination aspect of *ReSum* performed well. Repetition on concepts and/or evidence was minimal or absent since it does not select the highly rated sentences but readjusts the score of all the sentences according to the one that selected for the summary. In *TextAnalyst* however, it was evident the repetition of the concepts presented. Actually, in one case, it included in its summary two identical sentences, coming from a review that was submitted twice!

It was also observed that, the special nature of the review data affects the performance of plain text summarizers like *Copernic*. Although its operation is based on statistical methods, its results were affected on 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 *ReSum* in two more summarization tasks worth mentioned. In the first 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. The sentence that was selected by the summarizer was:

> *"*<OS> *refused to recognize the RAID-1 array I created in the BIOS using either of the controllers on the board during the install".*

In the second case, a *ReSum*'s summary for a printer was not only comparable to a human-made summary, but also in agreement to an expert's review, a heavy user of the printer. The summary was 77% accurate according to the human summarization and 75% according to the expert's review.

6 CONCLUSIONS AND FUTURE WORK

The availability of on-line reviews for goods and services is a valuable source of information that can help potential new customers in making an informed purchase decision.

In this paper, we proposed a novel, multi review, summarization algorithm that is based not only on the text of the review but on additional available metadata as well. These metadata include the reviewer's expertise to the domain, the time of ownership of the product or service under review, and most importantly, the usefulness of his/her review to other people that read it. Our experimental results demonstrate the usefulness of this metadata inclusion by means of improved precision and recall metrics.

Although our work was based on data from a certain on-line shop, the availability of similar metadata is becoming more common because it is an important factor for the success of online shops. We verified the applicability of our approach to other sources or reviews, such as the product comparison shop pricegrabber.com and we work towards a generalized version of our methodology that adapts to the availability or not of the various metadata. This is possible thanks to the modularity of equations (1) and (6) which encode the scoring algorithm that determines what to include in the calculated summary.

REFERENCES

[1] I. Mani, *Automatic Summarization*. John Benjamins Publishing Company, (2001).

[2] I. Mani, M.T. Maybury, *Advances in Automatic Text Summarization*, MIT Press, (1999).

[3] B. Liu, *Web Data Mining*, Springer, (2007).

[4] M. Hu and B. Liu, "Mining and summarizing customer reviews", In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, SIGKDD '04, 168-177 (2004).

[5] S. Morinaga, K. Yamanishi, K. Tateishi and T. Fukushima, "Mining Product Reputations on the Web", In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discover and Data Mining*, KDD '02, ACM Press, 341-349, (2002).

[6] K. Dave, S. Lawrence, and D.N. Pennock, "Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews", In *Proceedings of the 12th International World Wide Web Conference*, WWW 2003, ACM Press, 451-460, (2003).

[7] P. Nguyen, M. Mahajan and G. Zweig, "Summarization of Multiple User Reviews in the Restaurant Domain", Technical Report, Microsoft Research, MSR-TR-2007-126, September, (2007).

[8] A.M. Popescu and O. Etzioni, "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, Association for Computational Linguistics, 339-346, (2005).

[9] ΔEiXTo web data extraction tool: http://deixto.csd.auth.gr

[10] T.L. Saaty, *Decision Making for Leaders: The Analytic Hierarchy Process for Decisions in a Complex World*, Pittsburgh, Pennsylvania, RWS Publications, (1999).

[11] Copernic Summarizer: http://www.copernic.com

[12] TextAnalyst: http://www.megaputer.com/textanalyst.php