Mining Reviews for Product Comparison and Recommendation

Jianshu Sun, Chong Long, Xiaoyan Zhu, and Minlie Huang

***Abstract*—Recently, as the amount of customer reviews grows rapidly on product service websites, it costs customers much time to select and compare their favorite products. Researchers have been aware of this problem and many studies are investigated to mine the opinions from the online reviews. Unfortunately, few previous works give comparisons or recommendations among the products. In this paper, we propose an automated system to address this problem. We first build a product feature sentiment database from the reviews. Then we perform the comparison among various products from both subjective and objective perspectives on the feature level. Finally, product recommendations can be suggested according to the previous comparisons and an evolution tree constructed from the reviews. Experiment results demonstrate the effectiveness of the proposed approach in mining the digital camera reviews. And now a demo system is put in to practical use.**

***Index Terms*—Review mining, comparison, recommendation, evolution tree.**

I. Introduction

DUE to the emergence and development of Web2.0, more and more online review websites, such as Amazon [15] and Epinions [16], emphasize participation of the users. They encourage people to express their opinions on the products that they have purchased. These reviews are useful for both customers and manufacturers. However, it costs people a lot of time to find or collect useful information they want from so many reviews. Moreover, the judgment might be biased if only few reviews are analyzed. Instead of giving the users abundant but tedious reviews, it is better to summarize the reviews first, then perform comparisons among various products, and recommend good products according to the customer's demands.

Many researchers have proposed various approaches to mine product reviews. Hu *et al.* [1] and Liu *et al.* [2] developed a feature-based summarization approach on a large number of reviews of a product. In their work, they firstly tried to mine product features, and then identified opinion sentences with a positive or negative sentiment, which were summarized finally. In [6], the author proposed a novel relaxation-labeling technique to determine the semantic orientation of potential opinion words in the context of the extracted product features and specific review sentences. However, most of the foregoing work focuses on determining the sentiment polarity of a sentence or a review. Some researchers have noticed this limitation and try to evaluate the product by giving a sentiment score. In Scaffidi *et al.*'s work [8], they identified the product features and scores each product on each feature.

In this paper, we propose a system[1] to compare various products , perform recommendations to the customers and visualize the results. People can compare the products on feature level to help them make informed decision. Moreover, the user can clearly tell the strengths and weakness of each product via comparison, as Fig. 1 shows. To recommend products, we build a visualized evolution tree to help customers find candidate products, such as the one with better performance but lower price in the same generation, or one of best-selling products in the next generation.

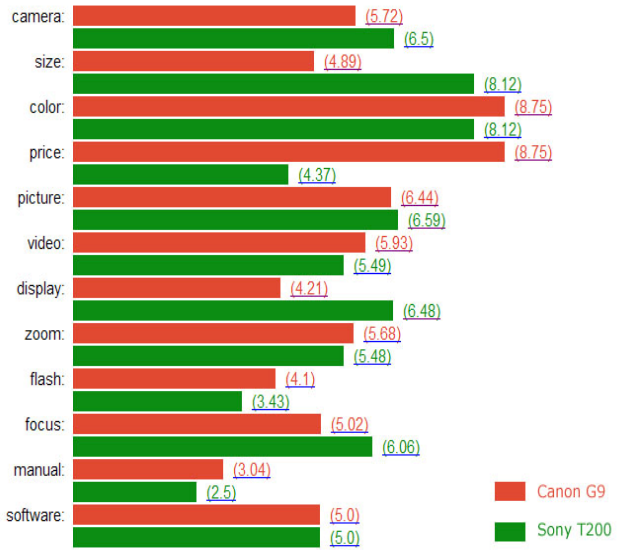

Fig. 1. Comparison visualization.

In summary, this paper has the following contributions:

- The proposed system can not only perform comparisons by mining reviews from the subjective perspective, but also incorporate product technical details to improve the comparison results from the objective perspective, which brings customers complete information.

Manuscript received November 5, 2008. Manuscript accepted for publication February 19, 2009.

Authors are with State Key Laboratory of Intelligent Technology and Systems Tsinghua National Laboratory for Information Science and Technology Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China. (e-mail: bigtree2005@gmail.com, longc05@mails.tsinghua.edu.cn, {zxy-dcs, aihuang}@tsinghua.edu.cn).

[1] Please visit our online system at "http://60.195.250.72/procar/"

- In our system, a new recommendation technique base on opinion comparison is proposed to suggest people some products with better performance. Moreover, we take the generation of product into consideration to ensure that the recommended products always have better physical performance.
- To the best of our knowledge, our system is the first one to construct the evolution tree of products. The evolution tree visualizes evolutionary process of products, which can indirectly recommend people potential favorable products.

The reminder of this paper is organized as follows: Section 2 describes some related work. The system architecture is discussed in Section 3. In Section 4, the proposed procedure is presented and the method is discussed. Experimental results are provided to confirm the effectiveness of the proposed approach in Section 5. Finally, the conclusion and future work are presented in Section 6.

II. Related Work

A. Comparative Opinion Mining

Comparison is one of the most convincing ways of evaluation. For example, "*The display of Sony T200 is good*" provides different information against "*The display of Sony T200 is better than Canon G9*". Clearly, the latter provides more useful message about the camera Sony T200. Moreover, in many cases, customers want to compare products in a fine granularity, such as display of a digital camera or the battery life of a mobile phone. Before purchasing a product, a customer may compare various features in details among his/her candidates to make decisions. In this sense, product comparisons are essential in E-commerce.

Researchers have paid their attention to this aspect via various approaches. Liu *et al.* compares one product with another one by identifying comparative sentences [3] and mining relations between two entities with respect to some common features [4]. His methods can achieve a relatively high precision. However, for the comparative sentences are rare in product reviews, it is hard to perform comparisons among any products on any features. Liu *et al*'s another work is implementing a prototype system called "Opinion Observer" [1] which focuses on analyzing and comparing opinions on the web. The system visualizes the comparison results so that the user is able to clearly see the strengths and weakness of each product in terms of various product features. However, the strength is simply generated by counting the number of positive opinions and negative opinions on one feature. In fact, the sentiment strength of each opinion is also very important when customers express their experience of a product. For example, the sentence "The display of Sony T200 is very excellent" obviously contributes more positive strength on the "display" feature than the ordinary statement "The display of Sony T200 is good". Pang *et al* [7] has focused on identifying opinion strength by classifying author's reviews into multi-point scale (e.g., one to five "stars"). While he cannot tell detailed scores on each feature, his work is just focusing on the document level.

Our work has gone further: we not only consider the strength of each customer's opinion, but also give a whole evaluation of each feature for a product, including incorporating product technical details. Comparison results based on each feature's evaluation have achieved a high precision.

B. Product Recommendation

Nowadays, recommendation is very common in electronic commerce's websites such as Amazon [15], Cnet [17]. When viewing a product's detailed description, customers are presented a product list similar as "What do customers ultimately buy after viewing this item?" or "Similar products". This recommendation technique mainly base on customer's visit records and previous classified categories. However, recommendation has much more requirements beyond that, including presenting products with better user experience and with suitable physical details.

In Scaffidi *et al*'s work [8], they implemented a prototype system called Red Opal to score each product on each feature for the users to locate products rapidly based on features. But simply ranking products according to user specific desired feature cannot satisfy the customer's demand, such as "Please recommend some digital cameras whose screen, size and picture quality are better than those of Sony T200", and failed to consider the product generation. There are also some researchers who perform product recommendation by modeling user preferences to implement personalized recommendation. Zhang *et al.* [9] have proposed a content-based personalized recommendation system which can learn user specific profiles from user feedback so that it can deliver information tailored to each individual user's interest. Differing from these personalized recommendation systems, our system focuses on statistical user opinions and recommends customers products with better subjective user experiences.

III. System Architecture

A. Definitions

Feature: A feature is an attribute/component of the product that has been commented on in reviews.

Opinion: The opinion of a feature in reviews is phrase (consecutive words) that expresses an opinion on the feature.

Feature-opinion pair: When a feature and its opinion occur in one sentence, we called them a feature-opinion pair.
For example, "*photos*" as a feature and "*very good*" as its opinion constitute a feature-opinion pair which expresses a positive opinion on the photo.
The photos come out very good.

Sentiment value (strength): A sentiment value is a scaled score from 0 to 1, evaluating the positivity of a sentiment. While 1 represents the most positive sentiment, 0 represents the most negative sentiment. Neural sentiment is scored 0.5.

TABLE I
FOUR CATEGORIES BY INTENSE FACTORS

category	adverbs	intense factor	adjective n-grams (sentiment value)
intense words	*very, too, -est*	*1.2*	*very excellent (0.75 = 0.625 * 1.2)*
ordinary words	*relatively*	*1*	*relatively excellent (0.625)*
weak words	*just,*	*0.8*	*just excellent (0.5 = 0.625 * 0.8)*
negative words.	*not, seldomly*	$\frac{1-sentiment}{sentiment}$ (*)	*not excellent (0.375 = 0.625 * (1-0.625)/0.625)*

Comments for Table I: The sentiment value of original "*excellent*" is 0.625. (*) The sentiment value of negative adjective n-grams equals the result subtracting the "*sentiment*" of original adjective from 1. The intense factor in the table is filled for consistency.

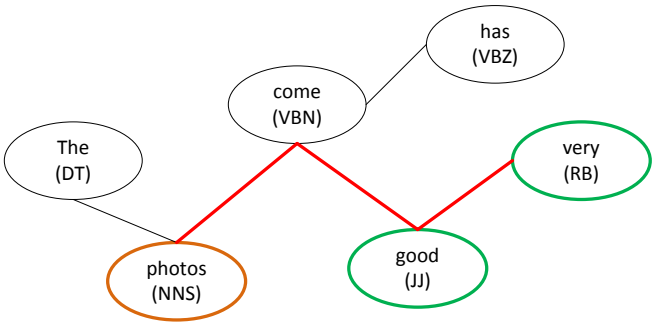

Fig. 3. Dependency grammar graph. This shows the dependency grammar graph generated by Stanford Parser [13]. The broad line indicates the dependency path from feature ("*photos*") to opinion ("*very good*").

2) Evaluate product feature sentiment

Since we want to compare products on a specific feature, product feature sentiment evaluation is essential in our work.

After finishing feature-opinion pairs mining procedure, the key problem now is how to assess the opinion on product features, which contains two sub-problems: how to evaluate one feature-opinion pair's sentiment and how to summarize all sentiment values on one product feature. Fortunately for the first problem, *SentiWordnet* [10] provides a list of words, in which each one has a positivity-score and a negativity-score with a scope of [0, 1]. We expand the word list a newly n-grams list, as we called "*Expanded SentiWordnet*", each word of which has a scaled sentiment value. The expanding rules mainly deal with "adjective n-grams" by multiplying the sentiment value of the adjective word by an *intense factor* α when there is an adverb before the adjective word. We previously classified the adverbs in *SentiWordnet* into four categories: intense words, ordinary words, weak words and negative words. Examples of four kinds of adverbs are showed as Table I.

We develop a *weighted voting method* to deal with the second problem. The method combines the opinion n-grams' frequency and its own sentiment value. Formally, for a product *i*, the score $feature\_score$ on a feature *j* can be calculated by Equation 1.

$$feature\_score(i, j) = \sum_{opinion\_weight_k >= 2} opinion\_weight_k \times opinion\_score_k . \quad (1)$$

In Equation 1, the opinion n-grams frequency $opinion\_weight$ is calculated by all the feature-opinion pairs related the feature. An $opinion\_score$ is looked up from the "*Expanded SentiWordnet*". In order to remove the noise, we only consider the opinion n-grams that occur more than once.

For each product, we can extract all the feature-opinion pairs from all the reviews of the product, and integrate all the sentiments of pairs into the feature level. After the features of all the products are evaluated in the same scale, we can store the results into databases.

B. Product Comparison and Recommendation

Differing from a 5-star schema in Amazon [15], our system compares products on the feature level. For example, from statistical results on Amazon, *Canon G9* and *Sony T200* have a 4.3-rating and 3.74-rating respectively, which means *Canon G9* has better overall user experience than *Sony T200*. But in our system, we want to tell that on "*display*" feature, *Sony T200* is better than Canon G9 (See the Fig. 4 for visualized results).

The system performs product comparisons based on previous *product feature sentiment database*. Formally, for any two products A and B, considering the feature j, whether product A is better than product B depends upon the value of Equation 2.

$$feature\_score(A, j) > feature\_score(B, j) ? \quad (2)$$

Most of time, two products have clearly different sentiment scores on identical feature. However, what if two products have the same or similar sentiment scores on the same feature? Here is an example: the "*picture*", "*zoom*" feature of "*Sony T200*", "*Canon G9*", "*Canon SD750*" display in the left chart of Fig. 4. The three products have achieved similar scores. An explanation of this phenomenon is that the three products have so high *picture* pixels (at least 7.1MP) that over satisfy ordinary people's demands, leading people unable to distinguish them. In addition, there are some circumstances when people may have biased opinions or have no comments on one feature. All of these demonstrate that subjective views have their own limitations.

contribution between review time distribution and core features.

After the scaled feature vector is sent to a cluster procedure, all the products are divided into several clusters. The number of clusters is predefined as N. In our experiments, we find N=3 leads to the best visualization of the evolution tree. By comparing the average time of product reviews in each cluster, we label each cluster as "*generation 0*", "*generation 1*", "*generation 2*", etc.

2) Building parent-child relations

Parent-child relations are constructed mainly according to product brands and *product models*, which means product's full name is essential in this step. This is reasonable, because judging whether "*Canon G9*" is the next generation of "*Canon G7*" only depends on the product "full name" information [15].

From top to bottom, the products in the current generation try to find parents in the hyper-generation. If being found, the product and the parent will be connected by a parent-child relation. If not, the product will recursively try to find parents in the next hyper-generation.

To ensure each product can find a parent, we construct the tree root as "*generation type*" by the product type, such as "*digital camera*", "*mobile phone*", followed by constructing node "*generation -brand*" by all the product brands, such "*Canon*", "*Sony*" (Please refer to Fig. 7 in Section 5.C for a visualized demonstration). Even if a product cannot find a real product parent, it can still find his brand node in the "*Generation brand*".

3) Merging the same parents

It is noticed that a child may be connected with more than one parent in the previous steps, so we need to merge all the parents that have the same child, which means the node of evolution tree may contains more than one product. This step is necessary for maintaining the tree structure. For example in Fig. 7, "*Sony W50*" is both the child of "*Sony W1*" and "*Sony W5*" in the hyper generation. So we merge the two parent nodes into one node as "*Sony W Sony W5*".

Note that steps 2 and 3 can be computed together. We present them separately for clarity. Please refer to Fig. 7 in Section 5.C for visualized evolution tree of a digital camera.

V. Experimental Results

To evaluate our proposed method, we perform extensive experiments on a corpus consisting of 23,585 product reviews from Amazon [15]. There are total 209 types of digital cameras, each of which has more than 50 reviews.

A. Product Comparison

1) Mining feature opinion pairs

To test the performance of product comparison module, 9 pairs of products with 18 product features are randomly selected from the 209 products.

Three people from different backgrounds are invited to annotate the comparison results. Each comparison on each feature is labeled by three annotators independently, with one of the following three labels according to the comprehension of feature sentence and related reviews:

- Label 'T': the left product is better than the right product on the feature.
- Label 'F': the left product is worse than the right product on the feature.
- Label 'E': the two products have the same or similar sentiment score on the feature.

Three people's annotations are combined into the final annotation as follows: If more than one annotator has the same label, the label is the final annotation. If three annotators have different labels respectively, the final annotation is 'E'.

From Fig. 5, we can see that our system have achieved encouraging performance on the product pairs. Our final average accuracy is 0.759, which means that the system can correctly performs comparisons on 14 out of 18 features in average. And when focusing on each pairs, we can find out limited differences. This is reasonable because some pairs have similar performance but some ones differ apparently, reflected by not only their ratings in Amazon, but also *product technical details*. For example, in the second pair, "*Canon S400*" has a 3.56-rating, while "*Canon SD750*" has a high 4.70-rating. In the last pair, "*Sony T200*" and "*Canon G9*", famous for their pocket size and advanced performance respectively, have tremendous differences in *product technical details*. The third pair of "*Panasonic FZ50*" and "*Canon SD750*" has the same ratings in Amazon and similar physical performance, which leads a relative low accuracy of the comparison results.

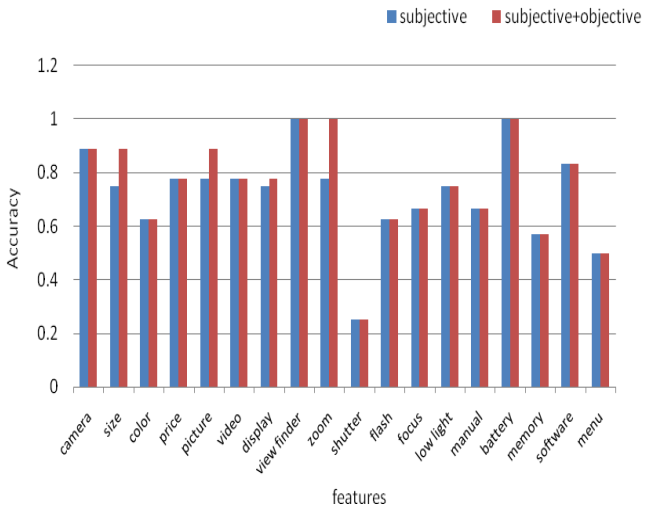

Fig. 6. Accuracy performance of comparison results over 18 product features.

From another perspective, Fig. 6 demonstrates the results over 18 product features. As you can see, the system achieves different accuracy performance over different features. On "*view finder*", "*zoom*" and "*battery*", the system even gives no errors. This is mainly because customers care about these features most when they choose a product. On the other side, common customers pay less attention to "*shutter*" or "*menu*", which leads to a bad performance on these features. An

No.2007CB311003, and Microsoft joint project "Opinion Summarization toward Opinion Search". The work was also supported by a grant from the International Development Research Center, Canada.

REFERENCES

[1] Bing Liu, Minqing Hu and Junsheng Cheng, “Opinion Observer: Analyzing and comparing opinions on the web,” in *Proceedings of WWW 2005*, pp. 342-351, 2005.

[2] Minqing Hu and Bing Liu, “Mining and summarizing customer reviews,” in *Proceedings of ACM-KDD 2004*, pp. 168-177, 2004.

[3] Nitin Jindal and Bing Liu, “Identifying comparative sentences in text documents,” in *Proc. of SIGIR-06*, pp. 244-251, 2006.

[4] Nitin Jindal and Bing Liu, “Mining comparative sentences and relations,” in *Proc. of AAAI'06*, pp. 244-251, 2006.

[5] Li Zhuang, Feng Jing, and Xiaoyan Zhu, “Movie review mining and summarization,” in *Proc. of CIKM*, pp. 43–50, 2006.

[6] Ana-Maria Popescu and Oren Etzioni, “Extracting product features and opinions from review,” in *Proc. of EMNLP-05*, pp. 339-346, 2005.

[7] Bo Pang and Lillian Lee, “Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales,” in *Proc. of ACL 2005*, pp. 115–124, 2005.

[8] Christopher Scaffidi, Kevin Bierhoff, Eric Chang, Mikhael Felker, Herman Ng and Chun Jin, “Red Opal: Product-Feature Scoring from Reviews,” in *Proc. of ACM EC*, pp. 182-191, 2007.

[9] Yi Zhang and Jonathan Koren, “Efficient Bayesian hierarchical user modeling for recommendation systems,” in *Proc. of SIGIR-07*, pp. 47-54, 2007.

[10] Andrea Esuli and Fabrizio Sebastiani, “Sentiwordnet: A publicly available lexical resource for opinion mining,” in *Proceedings of LREC 2006*, pp. 417-422, 2006.

[11] Yiming Yang and Jan O. Pedersen, “A comparative study on feature selection in text categorization,” in *Proc. of International Conference on Machine Learning (ICML)*, pp. 412-420, 1997.

[12] Marie-Catherine de Marneffe, Bill MacCartney and Christopher D. Manning, “Generating typed dependency parses from phrase structure parses,” in *Proc. of LREC 2006*, 2006.

[13] *Stanford Parser*, http://www-nlp.stanford.edu/software/lex-parser.shtml

[14] *SentParBreaker*, http://text0.mib.man.ac.uk:8080/scottpiao/sent_detector

[15] *Amazon*, http://www.amazon.com

[16] *Epinions*, http://www.epinions.com

[17] *Cnet*, http://www.cnet.com