CUCIS Technical Report
Mining Online Customer Reviews for Ranking Products

Kunpeng Zhang, Ramanathan Narayanan, Alok Choudhary
Department of Electrical Engineering and Computer Science,
Northwestern University,
Evanston, IL, 60208, USA
{kzh980, ran310, choudhar}@eecs.northwestern.edu

Abstract

The rapid increase in internet usage over the last few years has led to an extraordinary increase in electronic commerce. E-commerce web site like Amazon.com has made shopping online convenient, reliable and fast. While virtually any product can be purchased online today, it has become increasingly difficult for consumers to make their purchasing decisions based only on pictures and short description of a product. Since many online merchant sites allow customers to add reviews of the products they have bought, these reviews have become a diverse, reliable resource to aid consumers. The number of consumer reviews available has increased to an extent where it is no longer possible for a user to peruse them all manually. For example, some digital cameras sold on Amazon.com have several hundreds of reviews containing thousands of sentences. In this paper, we propose a novel text mining technique which uses customer reviews to rank products. We identify subjective and comparative sentences in reviews, and use these to build a weighted, directed product graph. This graph is then mined to find the top-ranked products. Experiments on real-world datasets show that our ranking algorithm produces results which correspond well with a manual ranking performed by subject experts.

Keywords Sentiment orientation, sentence classification, product ranking, graph, customer reviews.

1 Introduction

The rapid proliferation of internet connectivity has led to increasingly large volumes of electronic commerce. A study by Forrester Research[29] predicted that e-commerce and retail sales in the US during 2008 were expected to reach $204 Billion, an increase of 17% over the previous year. As more consumers are turning towards online shopping over brick and mortar stores, a number of websites offering such services have prospered. Amazon.com, Zappos.com, ebay.com, newegg.com are a few examples of e-commerce retailers which offer consumers a vast variety of products. These platforms aim to provide the consumers a comprehensive shopping experience by allowing them to choose products based on parameters like price, manufacturer, product features etc. Since it is difficult for consumers to make their purchasing decisions based only on an image and (often biased) product description provided by the manufacturer, these e-commerce platforms allow users to add their own reviews. Consumer reviews of a product are considered more honest, unbiased and comprehensive than a description provided by the seller. They also relate to customer's use of a product thereby linking different product features to its overall performance. A study by comScore and the Kelsey group [30] showed that online customer reviews have significant impact on prospective buyers. As more customers provide reviews of the products they purchase, it becomes almost impossible for a single user to read them all and comprehend them to make informed decisions. For example, there are several popular digital cameras at Amazon.com with several hundreds of reviews, often with very differing opinions. While most websites offer the customer the opportunity to specify their overall opinion using a quantitative measure, it leads to obfuscation of the multiple views expressed in a review. More importantly, reviews often contain information comparing competing products which cannot be reflected using a number measure. Also, different users have varying levels of quantitative measures(ex. easy grades vs. tough graders) thereby making the use of such numerical ratings even more difficult.

The widespread availability of customer reviews has led a number of scholars doing valuable and interesting research related to mining and summarizing customer reviews[1, 2, 3, 4, 5, 6]. There has also been considerable work on sentiment analysis of sentences in reviews, as well as the sentiment orientation of the review as a whole[25]. In this work, we aim to perform a ranking of

products based on customer reviews they have received. The eventual goal of this work is to create a tool that will allow users to select products based on reviews they have received. To the best of our knowledge, there has been no comprehensive study using thousands of customer reviews to rank products. It is our hypothesis that this ranking will aid consumers in making better choices. In a typical review, we identify two kinds of sentences that are useful in ranking products

1. Subjective sentences: Sentences containing positive/negative opinions regarding a product. *Examples*:This camera has great picture quality and conveniently priced (positive subjective sentence), The picture quality of this camera is really bad. (negative subjective sentence).

2. Comparative sentences: Reviewers often compare products in terms of the features, price, reliability etc. Comparisons of this kind are crucial in determining the relative worth of products. *Examples*: This camera has superior shutter speed when compared to the Nikon P40, This is the worst camera I have seen so far.

After developing techniques to identify such sentences, we build a product graph that captures the sentiments expressed by users in reviews. The advantage of using a directed graph whose edges indicate the preference of a user of pone product over another is that we can use several graph mining algorithms to generate a product ranking.

Particularly, the main contributions in this work are

- It is known that certain kinds of sentences reflect the sentiment/intent of the customer and are therefore more useful while building a ranking system. We use natural language processing methods and a dynamic programming technique to identify comparative and subjective sentences within reviews.

- Using the sentence classification technique, we build a weighted, directed graph which reflects the inherent quality of products as well as the comparative relationships between them.

- We then mine the product graph using our ranking algorithm. By considering both edge weights and node weights, we are able to develop a ranking system that incorporates the comments made regarding a product as well as comparisons between products. We show that our technique yields a ranking consistent with that performed by subject experts manually.

In addition, we also study the relationship between the rank generated by our algorithm and parameters like product cost, sales rank and average customer rating.

The remainder of this paper is organized as follows. Section 2 contains a summary of research related to our study. We explain our proposed technique in Section 3. Section 4 contains the details of our datasets and experimental evaluation followed by the conclusions in Section 5.

2 Related Work

Our work is partly based on and closely related to opinion mining and sentence sentiment classification. In [7], the authors study the problem of generating a "rated aspect summary" of short comments, which is a decomposed view of the overall ratings for the major aspects described in the comment. As a result, the user can gain different perspectives towards the target entity. They propose a topic modeling method, called Structured PLSA[8], modeling the dependency structure of phrases in short comments to extract major aspects. In addition, they use two unsupervised approaches (local prediction and global prediction) to predict ratings for each aspect from the overall ratings. Our work differs from theirs in the following aspects: (1) We rate products in terms of overall customer evaluations; not from the perspective of certain aspects. (2) We do not extract and classify any aspects or features but deal with sentences which express a customer's opinions and views. (3) A different algorithm and framework is employed to rank products instead of using machine learning methods. In [4, 1, 2, 9, 10], Liu et al. carry out detailed studies on solving problems related to extraction, summarization and classification of opinionated sentences. They focus on analyzing the syntactic organization of a sentence and use opinion words collected from WordNet[11] to classify the sentence into positive or negative category. Our research builds on this effort by performing a more detailed classification for sentences. Each sentence will be put into one of five groups (positive comparative, negative comparative, positive subjective, negative subjective and others). Another related work is [12], where the authors use mine sentences to get syntax and semantic structure patterns.

While some researchers focus their studies on the impact of online product reviews on sales, an important question remains unanswered, that is, can online product reviews reveal the true quality of the product? To test the validity of this hypothesis, the authors in [13] use data from Amazon to test the underlying distribution of online reviews and try to answer this question.

In summary, most of the current related work focuses on problems in opinion mining, product aspect

rating, review summarization etc. To the best of our knowledge, there has been no focused study regarding ranking products based on customer reviews.

3 Methods

Figure 1 shows the overall architecture of our product ranking process. The first step is data collection and preprocessing which includes crawling, downloading reviews, extracting relevant product information, splitting each review into sentences, and tagging parts of speech[18] for each sentence. This step will generate formatted data which is used as input for the sentence classification step. We aim to identify comparative and subjective sentences in customer reviews. For the comparative sentences, we need to know the pairs of products being compared. We use a dynamic programming based technique to identify these pairs. We also need to identify the sentiment orientation(positive/negative) of a sentence. The classified sentences, along with their sentiment orientation are stored in a sentence statistics database. This dataset is used to build a directed product graph. The edge weights and node weights of this product graph are determined by information gleaned from customer reviews. We then describe a ranking algorithm, which uses the product graph to generate a ranked list of products.

3.1 Data Collection The number of customer reviews available online is growing tremendously as online shopping becomes more popular. It is impossible to collect these online customer reviews manually. As review blogs and social networking sites emerge, it is also becoming more difficult to define what a customer review is. In general, we have to use web crawling techniques to extract reviews. In this paper, we make use of APIs provided by online merchants to get product information and customer reviews.

3.2 Sentence Splitter and Part-Of-Speech Tagging A customer review typically comprises of several sentences. It is not uncommon to see that customer express multiple positive and negative opinions of a product within a single review. For example, a customer reviewing a digital camera may use a couple of sentences to praise the quality of picture but use other sentences to belittle the weight and color of the camera. It is very hard to determine the sentiment orientation of such a review as a whole. To simplify the problem, we split reviews into sentences in which case, where it is easier to assign positive or negative sentiments. In this study, we do not consider sentences which express both positive and negative sentiments. We use MXPOST[14] to split reviews into sentences.

It is known that most sentiment bearing words are adjectives, and this information is useful to determine if a sentence is subjective, and whether it expresses positive/negative sentiment. In order to help us identify the sentiment orientation of a sentence, or to identify if a sentence describe a comparison between two products, we need to know the part-of-speech information. CRFTagger[15], a java-based conditional random field part-of-speech (POS) tagger for English is employed here to label each word. After finishing these operations, each sentence is saved along with the POS tag information. An example of a review sentence from digital camera domain after part-of-speech tagging can be seen below.

It/PRP 's/VBZ very/RB easy/JJ to/TO learn/VB and/CC very/RB light/JJ weight/NN too/RB

3.3 Sentence Labeling Customers express their opinions about products in multiple ways. We identify two distinct categories of sentences, which capture a vast majority of customer opinions: (1) Direct praise/deprecation(Subjective sentences) and (2) Indirectly expressing an opinion by performing a comparison (Comparative sentences). In this section, we describe the methods used for identifying these different types of sentences. Section 3.3.1 explains how we use certain keywords, sentence semantics, and structural patterns to recognize comparative sentences. Since not every comparative sentence describes a relationship between two distinct products, we need to perform some refining steps to make each comparative sentence meet our requirements. In Section 3.3.3, we describe our strategy for determining sentence orientations(positive or negative) for subjective and comparative sentences.

3.3.1 Identifying Comparative Sentences Earlier work by researchers [16, 17] has shown that a small set of keywords can help identify almost all comparative sentences. This set consists of 126 words most of which are verbs. Some of these words explicitly show comparisons ("outperform, exceed, compare, superior, etc."), whereas some of them are implicit("prefer, choose, like, etc."). Using only this set of words to identify comparative sentences leads to a high recall but a relatively low precision. To improve the precision, the authors analyze the semantics of a sentence and its structural patterns. They have devised some rules which convey comparative implications as well. If an adjective or an adverb occurs in a comparative form, it delivers us comparative meanings regarding two entities. If an adjective or an adverb comes in a superlative form, it shows a comparative relationship between one entity and all other entities

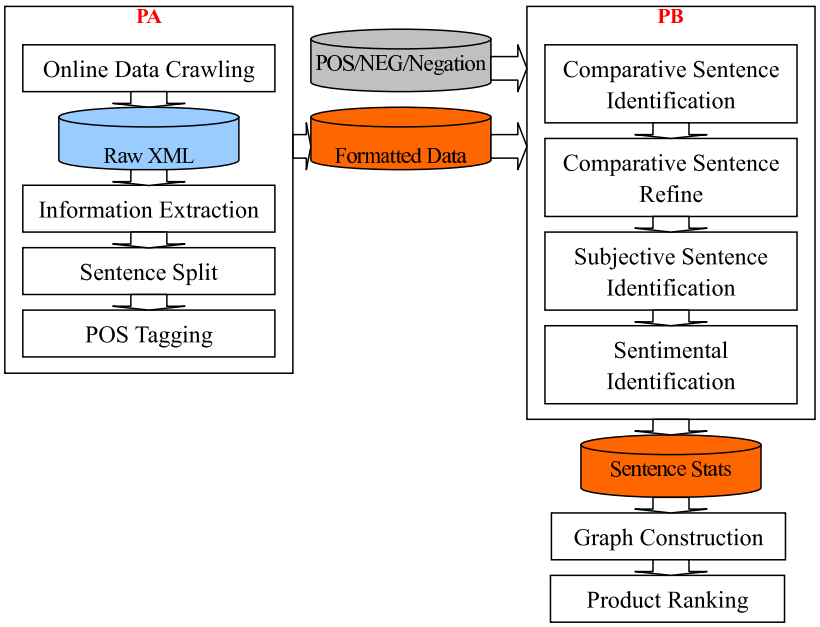

Figure 1: Online Customer Review-Based Product Ranking for Sellers and Buyers. *PA*: data collection and preprocessing; *PB*: sentence labeling. POS is the positive word set, NEG is the negative word set, and Negation is the negation set. Formatted data consists of product information and customer review sentences with POS tags. Sentence stats stores positive subjective, negative subjective, positive comparative, and negative comparative sentences.

under consideration. Therefore, words with POS tags of JJR(comparative adjective), RBR(comparative adverb), JJS(superlative adjective), and RBS(superlative adverb) are good indicators of comparative sentences. In addition, some special structural patterns also convey comparative meanings, for example, "as <word> as, the same as, similar to, etc". Altogether, the rules we use for identifying comparative sentences, $Rule_{CS}$, are as follows:

- Check if the sentence contains any word within the set of 126 comparative keywords
- Recognize any words with POS tags in JJR, RBR, JJS, RBS
- Scan if any predefined structural patterns are present in the sentence (as <word> as, the same as, similar to, etc.)

It is important to be aware of the fact that not all sentences satisfying these rules are comparative sentences in terms of product comparison. For example, the sentence "*I bought this camera for my son because he got a higher grade in his second statistical exam.*" does not show any comparative meanings or implications over other camera products. Therefore, we propose a more refined technique to find comparative sentences specifically related to product comparisons. A refined technique to identify comparative sentence with product comparison pairs is described in detail below.

3.3.2 Enhancing Comparative Sentences In order to perform product ranking, it is not sufficient to obtain raw comparative sentences by applying the rules given above. For our product ranking application, we need to know the products being compared by the customer. These sentences consist of some product comparison pairs explicitly or implicitly. A simple strategy to address this issue is to choose as candidates those sentences with at least one distinct product that is different from the product being reviewed. For example, the sentence "*This camera has superior shutter speed when compared to the Nikon P40.*" is a typical product-product comparative sentence. Based on this strategy, we propose a dynamic programming technique to identify such sentences. We also recognize a special case where comparative sentences without distinct product names also fall into this category. The reason is that they include superlative words. For example, the sentence "*This is the worst camera I have seen so far.*" indicates that this camera is worse than all other camera products. This means that it has a comparative relationship with all other camera products. Here we

use a dynamic programming technique (longest common subsequence) to capture all product-product comparison pairs. Before we dive into details of the algorithm, it is worthwhile for us to spend some time talking about the structure of product names. Many product names(supplied by manufacturers) are so long that customers will never use the name in its entirety. Rather customers prefer to write their reviews by using abbreviations or key words of product names, as long as they can be distinguished from other similar products. It is seen that the first three words of most products are sufficient to identify them uniquely. Also customers rarely use more than three words while describing a product. So we make a simplifying assumption that product names contain at most three words. An example of a product name which can be identified using only a subset of words is "*Canon PowerShot SD870IS 8MP Digital Camera with 3.8x Wide Angle Optical Image Stabilized Zoom (Silver)*". Our aim is to find all occurrences of a product name in a sentence, assuming that we have a known universe of product names. Unfortunately, customers do not follow any particular rule while mentioning products. Therefore, a simple string matching algorithm will not suffice. Our matching algorithm has three basic assumptions (ignoring the edge cases like when the product name is only a single word). Given a product name, and a candidate match: (1) If the candidate only matches the first word of a product name, we ignore this candidate match because the first word of a product name generally does not provide enough information to recognize a product (for example, words like "Lenovo, Apple, Mac are too generic)" (2) If the candidate matches the second word of a product name and the second word is included in predefined generic word set (words like "Powershot, ThinkPad etc."), we also ignore this match because it is again too generic to narrow down the products. A set of such generic keywords may be obtained by simple frequency thresholding of the product name universe. If the matched word does not belong to this set, we consider it a successful matching and use the first two words of that product name. (3) If it matches the third word (which is usually very specific to a particular product), we assume it matches this product. The detailed algorithm(**Get_Comparsion_Pairs**) is described below.

Algorithm 1 Get_Comparison_Pairs(CSENT, PNAME, PDS).
Input: Set of all comparative sentences: SENT, the universe of product names in that category: PNAME, predefined set of generic names: PDS;
Output: Each sentence along with a set of product comparison pairs.

```
1: for each sentence csent ∈ CSENT do
2:    print sent;
3:    oname = the name of the product being reviewed;
4:    for each product name pname ∈ PNAME do
5:       (firstw, secondw, thirdw) = split(pname);
6:       lcs = DO MATCHING Using Dynamic Programming;
7:       if (lcs == firstw) or (lcs == secondw and secondw ∈ PDS) then
8:          do nothing;
9:       else if lcs == firstw + secondw and secondw not ∈ PDS then
10:         relation pair: [oname, (firstw + secondw)];
11:      else if (lcs == thirdw) then
12:         relation pair: [oname, pname];
13:      end if
14:   end for
15: end for
```

3.3.3 Sentence Sentiment Orientation In this section, we describe how we assign sentiment orientations for a sentence. We only consider positive and negative sentiments in this work. Unfortunately, dictionaries and other sources like WordNet[11] do not include sentiment orientation information for each word. Some researchers[19] have used supervised learning algorithm to infer the sentiment orientation of adjectives from constraints on conjunctions. In [25], the author summarizes all kinds of techniques related to sentiment analysis. In this paper, we are using a simple yet powerful method by utilizing the adjective/adverb synonym set and antonym set in WordNet to form a positive word set(POS) and a negative word set(NEG) which are added to the sets from[28]. We use a set of adjectives/adverbs which are known to indicate sentiment, and then grow this set by searching in WordNet. To have a reasonably broad range of adjectives/adverbs, we use a manually derived set of very common adjectives/adverbs as the seed list. Then synonym and antonym searching functions provided by WordNet to expand this list. We are assuming that synonyms and antonyms of our seed list should also imply a corresponding sentiment. At the end of this process, we get a list of 1974 words for the positive set and 4605 words for the negative set, both of which are almost the same as[28]. We use a simple technique to identify the orientation of a sentence using these words. If the sentence contains a word which is also in the positive set, we label this sentence with a positive tag. Negative sentiment words are handled similarly. However, many customers do not like to express their opinions by using assertive sentences but using some negations

Algorithm 2 Sent_Labeling(SENT, POS, NEG, Negation).
Input: Set of review sentences of all products: SENT, the positive set: POS, the negative set: NEG, and the negation set: Negation;
Output: Sentence stats.

1: Classify SENT into comparative sentence set(*Comp*) and Non-comparative sentence set(*Non-Comp*): SENT⇒(*Comp, Non-Comp*);
2: Split comparative set into the set containing refined comparative sentences with comparison pairs(*RComp*) and general comparative sentence set(*GComp*) by using Dynamic Programming: *Comp*⇒(*RComp, GComp*);
3: **for** each *sent* ∈ *Non-Comp* **do**
4: **if** any word in the *sent* belongs to *POS, NEG* **then**
5: *sent* → the subjective sentence set(*Sub*);
6: **end if**
7: **end for**
8: Merge subjective sentences: *Sub*←*GComp* + *Sub*;
9: Identify sentiments for subjective sentences to get positive subjective set(*PS*) and negative subjective set(*NS*): *Sub*⇒*PS, NS*;
10: Identify sentiments for comparative sentences to get positive comparative set(*PC*) and negative comparative set(*NC*): *Comp*⇒*PC, NC*;
11: write *PS, NS, PC, NC*→sentence stats;

in their reviews. In this case, the orientation should be switched. We constructed a set of 28 negation words manually. It should be mentioned that we determine the sentence orientation for comparative sentences as well using the same list of sentiment words. Algorithm 2(**Sent_Labeling**) is listed below to summarize the flow of sentence labeling.

3.4 Constructing the Product Graph We use the subjective and comparative sentences found to construct a directed and weighted graph that can be mined to reveal the relative quality of products. The graph is defined as follows: $G = V, E$ where

- V is the set of nodes, $V = \{p_i \mid$ each product represents a node, $0 < i < n\}$,
- E a set of node pairs, called arcs or directed edges. An arc $e = (p_i, p_j)$ is considered to be directed from p_i to p_j. $E = \{e_k = (p_i, p_j), \mid W_{e_i}$ is the weight of the edge $e_i, 0 < i, j < n, 0 < k < m\}$,

where n is the number of products, m is the number of edges.

Consider a comparative sentence in the reviews for a product P_i. If this sentence compares product P_i with product P_j, we add an directed edge from P_j to P_i. The second step is to assign a weight to this edge. A comparative sentence occurring in the reviews for product P_i and comparing it with product P_j is considered a positive comparative($PC(P_i, P_j)$) if it implies that P_i is better than P_j. If it implies that P_i is worse that P_j, it is considered a negative comparative($NC(P_i, P_j)$). For each edge(P_j, P_i), we count the number of positive (PC) and negative (NC) comparative sentences associated with the pair (P_i, P_j) respectively. We assign the ratio PC/NC as the weight of the edge linking P_j to P_i. The last step is to assign weights for nodes. For a node P_i, we use the ratio of the number of positive(PS) and negative(NS) subjective sentences(PS/NS) as its weight.

3.5 Ranking Algorithm We borrow the concept of the PageRank algorithm[20] to evaluate the relative importance of each product. In the PageRank algoirthm, a node has a higher importance if it is pointed to from relatively important nodes. But our rank is a bit different in that we not only consider the relative importance among products, but also take the importance of the product itself into account. This means that the node weight is also crucial to the ranking, in addition to the edge weights. Therefore, we use a modified version of the PageRank algorithm, where node weights are non-zero. The termination of the PageRank algorithm is dependent on the *damping factor*. In PageRank, various studies have shown that a damping factor of 0.85 yields the best results, and we use this value. Let us illustrate the ranking process using a simple example. We have four products(A, B, C, D). The numbers of positive/negative, subjective/comparative sentences related to products are listed below.

$PS(A) = 1, PS(B) = 2, PS(C) = 3, PS(D) = 4$
$NS(A) = 3$
$PC(B, A) = 3, PC(B, C) = 7$

Table 1: The ranking output for the graph. Vertex ID 1, 2, 3, 4 represent node A, B, C, and D respectively.

Rank Score	Vertex ID
Rank 1: 0.820731	Vertex Id: 2
Rank 2: 0.072917	Vertex Id: 4
Rank 3: 0.053571	Vertex Id: 3
Rank 4: 0.052781	Vertex Id: 1

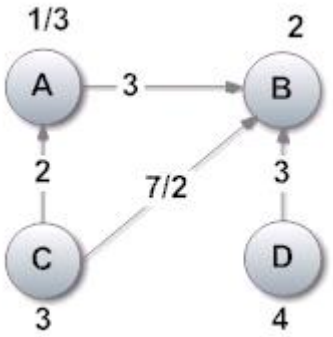

Figure 2: A simple ranking example with graph G having 4 products.

$PC(B, D) = 3, P(A, C) = 2$
$NC(B, C) = 2.$

Based on these sentence statistics, we could build a product graph G(see Fig. 2). Edge weights are determined by comparative sentences, and node weights are determined by subjective sentences. If the reviews of product P_1 have 1 positive comparative sentence mentioning product P_2, and 2 negative comparative sentences regarding mentioning P_2, there is an edge from P_2 to P_1 with weight 0.5. It must be mentioned that to prevent edges with infinite length (when the number of negative comparative sentences is 0), we set the minimum value of the denominator to 1 while computing edge weights.

By using our algorithm, we could get the ranking score for each node shown in the Table 1).

The ranking order(the smaller, the product better) for this graph is $B \rightarrow D \rightarrow C \rightarrow A$. From the graph, we clearly see that A, C, D are worse than B because all of them have edges pointing to B. D has more positive subjective sentences than A, C and their comparative weights with B are approximately equal. C has a better ranking than A because two sentences say A is better than C and 1/3 sentences say A is good comparing to 3 sentences saying C is good.

4 Experiment Results

4.1 Experiment Data We use customer review data from Amazon(www.amazon.com) to demonstrate the effectiveness of our ranking system. Amazon is one of the largest online retailers and has extensive customer review data. We use APIs provided by Amazon to download customer reviews and product information which include product name, product ID, product price, product sales rank, and customer rating score. All information is embedded into raw XML files which are downloaded by using the following algorithm. Though we experimented with data from various domains, only data from two domains(digital camera and television) are presented in the results section. Table 1 provides some metadata about the review datasets we are using. Further details about the datasets and the APIs used to generate this data can be found at [21]. The Table 2 shows the sentence statistics of each category. We could clearly see that 11% sentences are subjective sentences and 2% sentences are comparative sentences for Digital Camera data comparing to 10% and 4% for TV data.

Algorithm 3 Downloading(index,bn).
Input: The search index *index* and the browse node *bn* for each category;
Output: A database "RX"(Raw XML) storing product information and customer reviews.

1: Send HTTP Request along with *index* and *bn*;
2: ASINs = Find ASIN;
3: **for** each ASIN *asin* ∈ ASINs **do**
4: *review* = Find reviews;
5: *product_info* = Find product information;
6: Push (*review, product_info*) → "RX";
7: SASINs = Find ASINs for similar products;
8: **end for**
9: **for** each ASIN *sasin* ∈ SASINs **do**
10: Repeat from step (4) to step (7)
11: **end for**

4.2 Comparison with Expert Ranking In order to test the performance of our ranking algorithm, we compare the results with an expert ranking performed by subject experts. However, in order to ensure a fair comparison, we need to weed out certain products that do not belong to the category. For example, several products in the 'digital camera' category of Amazon are actually digital camera accessories. Since an expert evaluation would not contain such products, we need to manually remove them before performing the comparison. We use the expert recommendations present in SmartRatings.com[22] as a gold standard while evaluating our results. SmartRatings expert community is

Table 2: Experiment data summary of two types of data.

Category	Search Index	Browse Node	#Products	#Reviews	#Customers
Digital Cameras	Photo	281052	3990	83005	78026
TV	Electronics	172659	1765	24495	22611

Table 3: Sentences statistics for each category.

Category	# of Sentences	# of Subjective Sentences		# of Comparative Sentences	
		Positive	Negative	Positive	Negative
Digital Cameras	1469940	71565	97349	16246	10890
TV	460610	17843	28510	10224	9162

Table 4: Digital camera ranking comparison with expert ranking. We calculated the overlap between our top 10% products in the ranking list and products in expert ranking list. For a particular price range, if a product within top 10% of our ranking list also comes up in the expert ranking list, we increase the overlap count by 1. The first half of the third column in this table is the overlap number. It should be noted that experts review only a subset of products whereas there are a large number of products bought by customers which are not reviewed by the experts. For example, in the price range of $100-$200, 171 products were found to have reviews, all of which were used by our algorithm, whereas the experts only reviewed 17 of those products. Thus the overlap measure serves as a metric of confidence, but our ranking incorporates order of magnitude more products for ranking.

Price Range	# of Products	10%	# of Top Rated Products(Expert Ranking)
<100	159	-	0
100∼200	171	9/17	17
200∼300	98	7/10	10
300∼400	51	2/4	4
400∼500	25	2/3	6
500∼700	28	2/3	2
700∼1000	24	-	0
>1000	29	2/3	2
Average Probability of Overlap		**62.2%**	

Table 5: TV ranking comparison with expert ranking. We calculated the # of overlap between our top 10%, 20% products in the ranking list and products in expert ranking list. Here we have more expert ranking data than Cameras. The first half numbers of the third column and the fourth column in this table are the overlap numbers.

Price Range	# of Products	10%	20%	# of Top Rated Products(Expert Ranking)
<300	72	-	-	1
300∼400	46	-	-	1
400∼500	38	-	-	2
500∼600	27	2/3	3/4	4
600∼700	25	1/3	4/5	5
700∼800	21	1/2	3/4	6
800∼1000	47	3/5	5/9	9
1000∼1500	62	2/6	8/12	18
1500∼1800	18	1/2	2/4	15
1800∼3000	24	1/2	1/4	6
>3000	8	-	-	5
Average Probability of Overlap		**50%**	**62.3%**	

made up of devoted industry veterans, avid product enthusiasts and ardent consumer advocates. They are very knowledgeable and have a great deal of hands on experience with the products and services they cover. However, it must be noted that the number of products ranked by these experts is very less compared to the total number of products being sold on a large online retail website like Amazon. This is an indication of how difficult and time-consuming it is for human beings to identify and quantify product quality. Another observation is that products with different price ranges should not be compared together. Since a large price difference will have significant impact on product quality, and therefore, customer reviews. So it is reasonable to deal with different features and different types of products separately. In this paper, we put products into buckets with a price range of $100 for lower price ranges. For products with a higher price range, the number of products is limited; therefore we create buckets based on common sense. Table 2 and Table 3) show the overlap probabilities with expert rankings for digital camera and TV respectively. For digital cameras, the number of products ranked by experts is approximately 10% of the number of products in our list, while this figure is slightly higher in the TV domain. The table cells with a '-' sign indicate that, in the price range, either no products were ranked by the experts ranking list or the total number of products is close to zero. The results indicate that there is satisfactory agreement with expert ranking. It must be emphasized that our ranking algorithm uses only unstructured customer reviews, and yet achieves significant agreement with evaluations done by subject experts with several years of experience and insight in their respective fields.

4.3 Sales Rank as an indicator of Product Quality The Amazon sales rank is a number that says how many other products sold more than a particular product. The smaller the Amazon Sales Rank number, the better the sales of a product. The Amazon sales rank is normally re-computed daily. One might ask whether buyers' preferences correlate with sales rank. In [23, 27] and RankTracer[26], the authors conclude that sales rank is not a good indicator of the quality of products. A lower sales rank does not necessarily mean that a product has better quality or customer satisfaction. The results of our ranking algorithm also corroborate this point. Figure 3- Figure 8 show that there is no apparent positive(or negative) correlation between sales rank and ranking order. Similarly, average customer rating score does not indicate the overall quality of a product as well because different customers prefer different products based on their own interests. Therefore we have not used these metrics while computing the product ranking.

5 Discussion and Conclusion

Customers who are shopping online are highly influenced by customer reviews of products. However, as the number of customer reviews increase, it becomes impractical for a single user to read them all. In this paper, we proposed a novel technique for ranking products based on online customer reviews. We use natural language processing techniques to identify sentences in reviews that provide subjective and comparative information regarding products. The goal of this work is to help customers make better decisions without having to read a large repository of customer reviews. Our experimental results indicate that our product ranking is consistent with rankings done by subject experts.

In our future work, we plan to further improve and refine our methods. Firstly, current orientation identification of a sentence focuses on identifying either positive or negative sentiments. Since certain sentiment related words/phrases are highly opinionated as compared to other mildly opinionated words/phrases, it might make sense to have varying levels of sentiment. Secondly, we assign a rank to products based on its overall quality as perceived by customers. In many cases, customers are more interested in certain aspects/features of a product(in the case of digital cameras, these features might be picture quality, shutter speed, lens quality etc.). Finally, developing more efficient and precise ranking algorithms is continuing endeavor.

References

[1] M. Hu and B. Liu, *Mining and Summarizing Customer Reviews*, Proceedings of the 10th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD-2004), 8 (2004), pp. 168–174.

[2] M. Hu and B. Liu, *Mining Opinion Features in Customer Reviews*, Proceedings of the 19th National Conference on Artificial Intelligence., 7 (2004), pp. 755-760.

[3] A. Popescu and O. Etzioni, *Extracting product features and opinions from reviews*, Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing., (2005), pp. 339-346.

[4] B. Liu, M. Hu, and J. Cheng, *Opinion Observer: Analyzing and Comparing Opinions*, WWW., 5 (2005), pp. 342-351.

[5] S. Kim, P. Pantel, T. Chklovski, and M. Pennacchiotti, *Automatically Assessing Review Helpfulneww*, EMNLP., 7 (2006), pp. 423-430.

[6] B. He, C. Macdonald, J. He, and I. Ounis, *An Effective Statistical Approach to Blog Post Opinion Retrieval*, CIKM., 10 (2008), pp. 1063-1069.

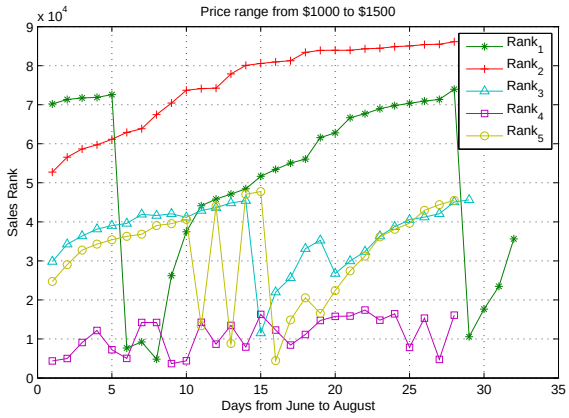

Figure 3: Price range from $1000 to $1500 for TV. Variety of sales rank over time for a few top ranked products. Horizontal axis represents days (from June to August) we collected data, the vertical axis is sales rank value. Each line in the graph represents the sales rank trend of a product with a ranking order calculated by our ranking system. The smaller the rank order, the better the product is. The sales ranking means how well the product is selling. From these graphs, it is clear that sales rank has no correlation with the ranking of a product.

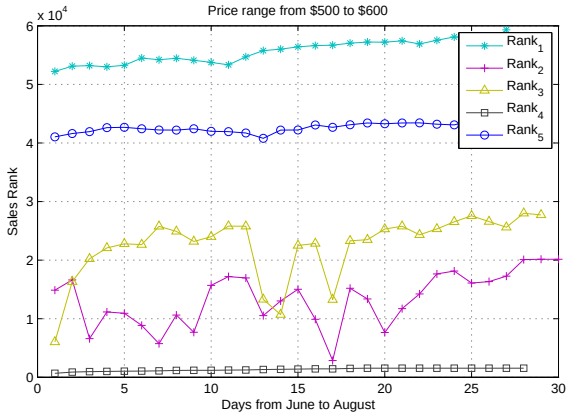

Figure 4: Price range from $500 to $600 for TV.

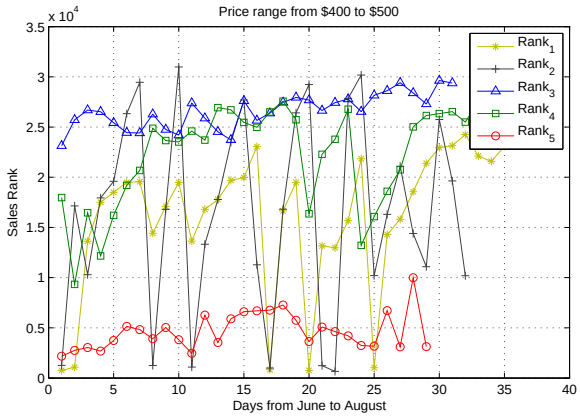

Figure 5: Price range from $400 to $500 for Digital Camera. The sales rank trends of digital cameras with different price ranges. Horizontal axis represents days we collect data, the vertical axis is sales rank. Each line in the graph represents the sales rank trend of a product with a ranking order calculated by our ranking system. The smaller the rank order, the better the product is. The sales rank indicates how well the product is selling. It has no correlation with rank order from these graphs.

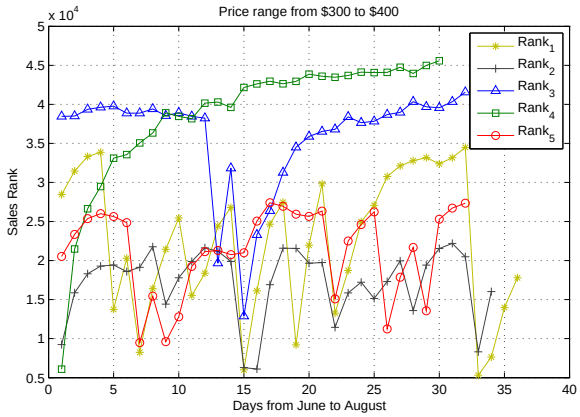

Figure 6: Price range from $300 to $400 for Digital Camera.

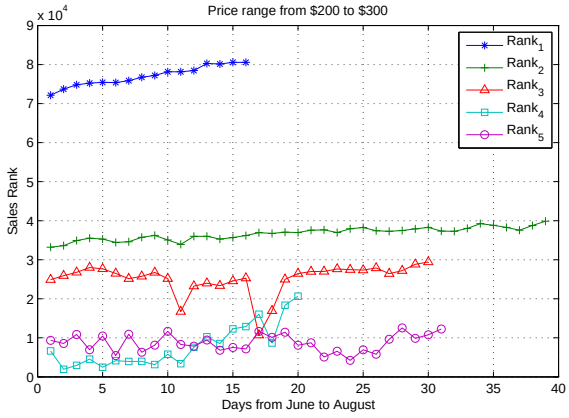

Figure 7: Price range from $200 to $300 for Digital Camera.

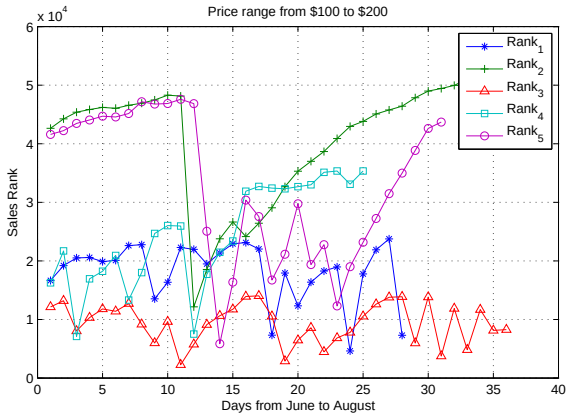

Figure 8: Price range from $100 to $200 for Digital Camera.

[7] Y. Lu, C. Zhai, and N. Sundaresan, *Rated Aspect Summarization of Short Comments*, WWW., 4 (2009), pp. 131-140.

[8] T. Hofmann, *Probabilistic Latent Semantic Analysis*, UAI., Stockholm, 1999.

[9] M. Hu, and B. Liu, *Opinion extraction and summarization on the web*, AAAI., 7 (2006), pp. 1621-1624.

[10] B. Liu, X. Li, W. S. Lee, and P. S.Yu, *Text Classification by Labeling Words*, AAAI., 2004, pp. 425-430.

[11] G. A. Mille, C. Fellbaum, etc., *WordNet: An Electronic Lexical Database*, 5 (1998),MIT Press.

[12] S. Arora, M. Joshi, and C. P. Rose, *Identifying Types of Claims in Online Customer Reviews*, NAACL HLT., 6 (2009), pp. 37-40.

[13] N. Hu, P. Pavlou, and J. Zhang, *Can Online Reviews Reveal a Product's True Quality? Empirical Findings and Analytical Modeling of Online Word-ofMouth Communication*, EC., 6 (2006), pp. 324-330.

[14] A. Ratnaparkhi, *A Maximum Entropy Part-Of-Speech Tagger*, In Proceedings of the Empirical Methods in Natural Language Processing Conference, 5 (1996), University of Pennsylvania.

[15] X. Phan, *CRFTagger: CRF English POS Tagger*, http://crftagger.sourceforge.net/, 2006.

[16] N. Jindal, and B. Liu, *Identifying Comparative Sentences in Text Documents*, SIGIR., 8 (2006), pp. 244-251.

[17] N. Jindal, and B. Liu, *Mining Comparative Sentences and Relations*, AAAI., (2006), pp. 1331-1336.

[18] C. Manning, and H. Schutze, *Foundations of Statistical Natural Language Processing*, 5 (1999), MIT Press.

[19] V. Hatzivassiloglou, and K. R. McKeown, *Predicting the semantic orientation of adjectives*, Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics, 1997, pp. 174-181.

[20] L. Page, S. Brin, R. Motwani, and T. Winograd, *The PageRank citation ranking: Bringing order to the Web*, Stanford University, 1999.

[21] `http://www.browsenodes.com/`.

[22] `http://www.smartratings.com/`.

[23] P. Chen, S. Wu, and J. Yoon, *The Impact of Online Recommendations and Consumer Feedback on Sales*, ICIS., 12 (2004), pp. 711-723.

[24] `http://en.wikipedia.org/wiki/Analysis_of_variance`.

[25] B. Liu, Sentiment Analysis and Subjectivity, to appear in Handbook of Natural Language Processing, Second Edition, 2010.

[26] `http://www.ranktracer.com/ripwidget.php`.

[27] F. Reichheld, *The One Number You Need to Grow*, Harvard Business Review, pp. 1-9.

[28] MPQA corpus `http://www.cs.pitt.edu/mpqa`, 2002.

[29] Forrester research, `http://www.comscore.com/Press_Events/Press_Releases/2007/11/Online_Consumer_Reviews_Impact_Offline_Purchasing_Behavior`.

[30] comscore and Kelsey, `http://www.shop.org/c/journal_articles/view_article_content?groupId=1&articleId=702&version=1.0`.