What Reviews are Satisfactory: Novel Features for Automatic Helpfulness Voting

Yu Hong[1] Jun Lu[2] Jianmin Yao[3] Qiaoming Zhu[4] Guodong Zhou[5]
School of Computer Science and Technology, Soochow University
No.1 Shizi Street, Suzhou City
Jiangsu Province, China
+86-18604601106
{tianxianer[1], lujun59[2]}@gmail.com, {jyao[3], qmzhu[4], gdzhou[5]}@suda.edu.cn

ABSTRACT
This paper focuses on exploring the features of product reviews that satisfy users, by which to improve the automatic helpfulness voting for the reviews on commercial websites. Compared to the previous work, which single-mindedly adopts the textual features to assess the review helpfulness, we propose that user preferences are more explicit clues to infer the opinions of users on the review helpfulness. By using the user-preference based features, we firstly implement a binary helpfulness based review classification system to divide helpful reviews and useless, and on the basis, we secondly build a Ranking SVM based automatic helpfulness voting system (AHV) which rank reviews based on their helpfulness. Experiments used a large scale dataset containing over 34,266 reviews on 1289 products to test the systems, which achieves promising performances with accuracy of up to 0.72 and NDCG@10 of 0.25, and at least 9% accuracy improvement compared to the textual-feature based helpfulness assessment.

Categories and Subject Descriptors
H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing – *Linguistic processing*.

Keywords
Automatic helpfulness voting, user preference, helpfulness classification.

1. INTRODUCTION
Nowadays, product reviews have become new attention-getting parts of most commercial sites[1]. In contrast to gorgeous picture or brilliant tagline in advertisement, users prefer to accept the opinions of other users, especially the experienced ones. And the user-supplied review is exactly the best medium for this interaction. However, the free network environment permits anyone to submit any review, even meaningless reply (e.g. yup, haw-haw, etc.) or disparaging remarks about competitors. Rather than enhance user experience, these reviews actually make so many noises and misapprehensions. Thus, it is crucial to have a mechanism capable of assessing the quality of reviews and shielding users from useless reviews.

Some websites already offer the mechanism for users to evaluate reviews. For example, on Amazon.com, an interface allows customers to vote for the helpfulness of each review and then ranks the reviews based on the accumulated votes. From an Amazon annotation of helpfulness: "*67 of 68 people found the following review helpful*" (Each Amazon review has this type of helpfulness annotation at the beginning part.). It specifies that 68 users voted and 67 of them thought the review is helpful. Such a voting rate (e.g. 67/68) gives a quantitative evaluation of review helpfulness, which can directly help other users select and scan high-quality reviews.

Unfortunately, the helpfulness is normally estimated by having users manually assess it. Thus it is hard to properly annotate helpfulness of newly submitted reviews and reviews with few votes as lack of manual information for the assessment. For example, for all Book products on Amazon.com, 26% reviews receive three or fewer helpfulness votes. Worse still, most websites never provide interactive function to acquire user experiences of reviews. Therefore, it is necessary to have a mechanism of automatic helpfulness voting.

Although few, there are still some pilot studies on the automatic helpfulness voting (abbr. AHV) for commercial reviews. Most of these works regard the AHV as an issue of textual-feature based classification, and verify the contributions of variety of textual features on distinguishing the helpful reviews from the useless. For example, the length of review is believed to be a simple but effective feature because a long review may offer wealth of product information.

However, AHV should be a more complex problem, which not only considers textual features (e.g. text length, language fluency and clearness), but also user preferences. As we have observed, whether a review gives the product attributes that users prefer to know, whether it wins the trust of users, and whether it has the consistent sentiments with users, all are important factors that influence the helpfulness voting. Consider the following reviews on Sony camera (from Amazon.com):

(1) *I prefer to buy the old version or Nikon*. (*3/54*)
(2) *It costs more than Nikon*. (*89/129*)
(3) *Amazing. This version offers 20x optical zoom. I even can see the sweat pores of my friends in photos*. (*125/158*)

Obviously, the review (1) cannot offer any valuable product information. This causes a very low voting rate *3/54* (6%). On the contrary, the reviews (2) and (3) show the price comparison and

*Corresponding author: Guodong Zhou

SIGIR'12, August 12-16, 2012, Portland, Oregon, USA.

unique function, which are normally uppermost in most minds. This should be the reason why they achieve very high voting rates *89/129* (69%) and *125/158* (79%). Especially, the review (3) implies that the reviewer might have bought the product, which makes the words more persuasive and helpful. The examples show the textual features might be not always effective in assessing review helpfulness, e.g. the length of review (1) versus (2), but user preference learning seems to give a supplementary or even alternative method to solve the problem.

In this paper, we focus on learning three kinds of user preferences:

- Users prefer the reviews that meet their ***information needs***
- Users prefer the ***credible*** reviews
- Users prefer the reviews that have the ***mainstreaming opinion***

On the basis, we attempt to capture features that to some extent represent the preferences without any human intervention, and well use the features to improve the helpfulness-based review classification. Empirically, the textual features, e.g. the grammatical correctness, language fluency and clearness, etc., are more commonly used to detect whether the writings of reviews can make comfortable reading for users. Compared to this, our features are used to detect whether reviews can provide valuable, credible and authoritative product information for users.

The rest of the paper is organized as follows: Section 2 introduces the related work. In Section 3, we define the task of automatic helpfulness voting (AHV). In Section 4, we present the user preferences on product review and offer the quantitative evidences to support the preference-based AHV. In Section 5, we present the method of feature extraction and corresponding measurement. Section 6 gives the experimental setup. The main results are reported in Section 7. At last, we draw the conclusion in Section 8.

2. RELATED WORK

A large body of research shows the availability of product reviews in extracting product aspects[2][3] and features[4], mining user opinions[2][4] and summarizing corresponding sentiments[3]. However, there is so far a little research focusing on whether and how a product review should be determined to be helpful or useless (viz., reviews helpfulness assessment, abbr., RHA). This section specially introduces the related work on RHA.

Kim et al[1] exploited the multitude of user-rated reviews on Amazon.com, and trained an SVM regression system to learn a helpfulness function and then applied it to rank unlabeled reviews. Especially this work performed a detailed analysis of different features to study the importance of several feature classes in capturing helpfulness. Within the features, the length of reviews, unigrams, and product ratings were the most useful features, but structural features (other than length) and syntactic features had no significant impact.

Liu and Cao et al[5] studied the problem of detecting low-quality product reviews. This work firstly discovered three types of biases in the ground-truth used extensively in the previous work, and proposed a specification on the quality of product reviews. The three biases are imbalance vote bias, winner circle bias, and early bird bias. Secondly, rooting on the new ground-truth (conforming to the proposed specification), this work gave a classification-based approach to low-quality product review detection, which yields better performance of opinion summarization.

Jindal et al[6] regarded the issue as spam filtering, and gave three general types of spam reviews: 1) untruthful reviews, such as fake reviews 2) reviews on brands only which do not comment on the products but only the brands, the manufacturers or the sellers of the products and 3) non-reviews (e.g., questions, answers, and random texts). On the basis, this work detected reviews of type 2 and type 3 based on traditional classification learning using manually labeled spam and non-spam reviews, and detected type 1 by verifying whether reviews involve many opinions opposing to most other reviewers.

Cristian et al[7] confirmed that a review's perceived helpfulness depends not just on its content, but also the relation of its score to other scores. This dependence on the score contrasts with a number of theories from sociology and social psychology, but is consistent with a simple and natural model of individual bias in the presence of a mixture of opinion distributions.

Tsur et al[8] present a RevRank algorithm to rank book reviews according to review helpfulness. The RevRank is more like an adaptive Rocchio[9] algorithm when setting and modifying the "virtual core" of a set of reviews on a product. On the basis, reviews are ranked according to their distance from the "core". Although RevRank is proved to outperform a baseline imitating the Amazon user vote review ranking system, few evidences have be showed to illustrate the inevitability of the uselessness of those reviews which deviate from the "core".

Liu and Huang et al[10] et al show that the helpfulness of a review depends on three important factors: the reviewer's expertise, the writing style of the review, and the timeliness of the review. Although not clearly mentioned by Liu and Huang et al, the factors indeed reflect that the exhilarating reviews normally cater to users' tastes in the review quality (e.g. a new and well-written review from an expert is welcomed). This motivates us to explore the user-preference based review helpfulness prediction, and further search the features effectively describing user preferences.

Jo et al[11] propose an aspect and sentiment unification model to automatically discover what aspects are evaluated in reviews and how sentiments for different aspects are expressed. This work motivates us to consider whether sentiment similarity between users and reviewers affects the user preferences on reviews, and further is an effective feature to infer the determination of users for review helpfulness.

3. TASK DEFINITION

In this Section, we firstly define the task of automatic helpfulness voting (AHV), and secondly we introduce the data format of Amazon.com reviews. The data have been successfully used in previous AHV tests, so we in later Sections conduct all our analysis, illustrations and experiments on this type of data.

3.1 Task of AHV

The task of AHV firstly aims to automatically assess the review helpfulness, and secondly rank all reviews of a specific product based on their helpfulness scores. It should be performed only depending on the existing Web information without any manual intervention. For example, AHV should determine an Amazon review as helpful and highly rank it if it has the manual voting rate[1] of "*197 of 199*", and AHV should perform this under the condition of being blind to the rate. On the contrary, an Amazon review which has only "*16 of 199*" voting rate should be determined as useless and ranked lower in the ranking list of reviews.

[1] The voting rate cannot be used in automatic helpfulness voting (AHV) because it involves the manual intervention.

Therefore, given an input of a random sequence of reviews, the AHV system should output a helpfulness-based ranking list of reviews. Within this, the most key issue is to automatically assess the review helpfulness, so in this paper we firstly simplify the task of AHV, and regard it as a helpfulness-based binary classification of reviews, by which to explore the valuable features for the helpfulness assessment. On the basis, by using a Ranking SVM model[12], we finally implement a simple ranking system for product reviews.

To evaluate the system, we need two basic evaluation metrics: accuracy and NDCG@n. The accuracy is used to evaluate the performance of the helpfulness-based review classification and inspect whether every review (including both helpful and useless ones) obtain the correct decision[13]. To do this, we succeed the optimal helpfulness boundary proposed by Kim et al[1] who report the voting rate 0.6 can be used to approximately divide the helpful reviews from the useless. So we regard the boundary as the criterion to label the training and test samples for the binary classifier. The metric NDCG@n is used to evaluate the ranking system, and inspect whether its output meets the manually labeled helpfulness ranks.

Table 1: Format and availability of Amazon reviews

Review <serial number>	
<Product> *Nikon COOLPIX P300* </Product>	---Available
<Title> *Low Resolution* </Title>	---Available
<Star> *one-star* </Star>	---NA
<Time> *2011-3-4* </Time>	---Available
<Reviewer> *Unicorn John* </Reviewer>	---Available
<Content> *The photos are not clear.* </Content>	---Available
<Vote> *56/78* </Vote>	---NA
<Buyer?> Yes </Buyer?>	---NA
<Er> *The COOLPIX P300 is…* </Er>	---NA
<Pt> *9×7×6 inches …* </Pt>	---NA
<Ps> *From the manufacturer …* </Ps>	---NA
<Url> *http://www.amazon ...* </Url>	---Available

3.2 Amazon.com Data Format

In this paper, we adopt reviews on Amazon.com as our corpus to evaluate the AHV system. Amazon.com [2] provides 1,692,256 reviews on 13 types (91 subtypes) of products, such as that on Books, Kindle, Electronics, etc. An Amazon.com review normally includes following components: title, star, release time, reviewer, main content and vote. But for most reviews, especially the most recent ones, the item of <vote> is empty for lack of voters, which is the problem that should be solved by the AHV system. We standardized the format of Amazon reviews and restricted the availability of the components as the example in Table 1.

Within the review description, <Er>, <Pt> and <Ps> respectively denotes the editorial review, product detail and product description. The item <Er> is written by domain experts. The item <Pt> normally gives the specification, bestsellers rank, etc. And the item <Ps> often introduces the product functions, business analysis, and manufacturer. But not all Amazon merchandisers provide the contents of the three items, and particularly most commercial websites never involve that in reviews. Therefore, to meet the needs of the general AHV system, we restricted the use of <Er>, <Pt> and <Ps> (labeled by "NA" in Table 1). Besides, the <Star> denotes the vote on product value (from 1 to 5 stars) and the <Buyer?> records whether the reviewer is a buyer of the product. Because the two items need human intervention, we also restrict the use of them in AHV.

Actually the "Available" items in Table 1 may be all effective features to assess the helpfulness, but some of them cannot support our exploration on user-preference based AHV. For instance, the <Reviewer> provides a link to the historical reviews of a reviewer, and the corresponding voting rates can provide important reference for assessing current reviews (motivated by the hypothesis that an experienced reviewer may always give helpful reviews). But this doesn't belong to the application of user preferences in AHV, so we ignored the items in our system.

In this paper, we only regard the items <Product> and <Content> as the available local resources for AHV. The <Product> gives the product name, and the <Content> records the main contents of reviews. Other items, as discussed in the next Section, are only used to generate the quantitative evidences for our user-preference based AHV.

4. USER PREFERENCE LEARNING

In this Section, we present the user preferences on product reviews along with the quantitative evidences for their availability in assessing review helpfulness.

4.1 Needs Fulfillment

The hypothesis is that the basic user intention is to acquire the product information from reviews, by which to support the purchase decisions. If the hypothesis is true, a helpful review should have the prerequisite that it can meet the information needs. But the question is what the information needs are? By analyzing the interactive model in commercial websites, it is not hard to find the user behaviors of searching product attributes and functions before purchase. Therefore, we in this paper regard the product attribute and function as two basic information needs, and verify the possibility of using the needs fulfillment to improve review helpfulness assessment.

Table 2: Main types of products on Amazon.com

P1. Unlimited Instant Videos	P9. Movie, Music & Games
P2. MP3s & Cloud Player	P10. Electronics & Computers
P3. Amazon Cloud Drive	P11. Home, Garden & Tools
P4. Kindle	P12. Grocery, Health &Beauty
P5. Appstore for Android	P13. Toys, Kids & Baby
P6. Digital Games & Software	P14. Clothing, Shoes & Jewelry
P7. Audible Audiobooks	P15. Sports & Outdoors
P8. Books	P16. Automotive & Industrial

To prove the feasibility of the needs fulfillment in AHV, we use the items <Content>, <Er>, <Pt> and <Ps> (as shown in Table 1) to calculate the capture rate of the words of product attributes and functions in the <Content> of reviews, and generate the curve diagram of the average rates on 16 types of Amazon products (see the types in Table 2) with up to 20,000 corresponding reviews, by which to compare the distributions of needs fulfillment between the helpful and useless reviews. Thereinto, the item <Pt> provides the main words of product attributes, and the <Ps> offers that of product functions. Besides, because the item <Er> gives the review from the domain expert, we regard the capture rate in <Er> as the best measure of needs fulfillment. We show the curves and the equation of capture rate in Figure 1.

[2] http://www.amazon.com/

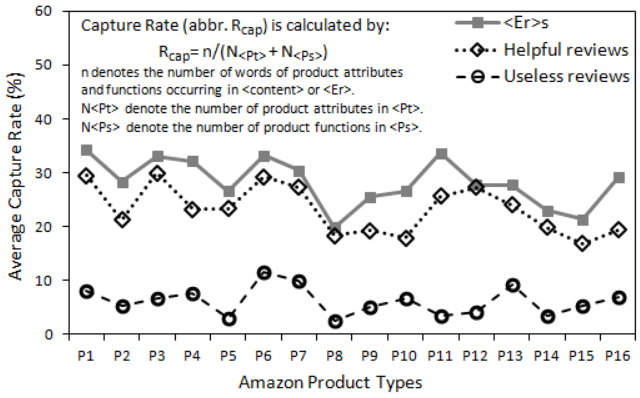

Figure 1. The capture rates of product attributes and functions in Amazon reviews

As shown in Figure 1, it is not hard to find that the capture rate curve of helpful reviews is very close to the curve of <Er>s. On the contrary, the curve of useless reviews is on the bottom of the diagram and far from the curve of <Er>. It illustrates that the helpful reviews, just like the specialist reviews, normally have the capacity to meet the user needs for product information (here only the product attributes and functions). In other words, the needs fulfillment is an important factor for helpfulness assessment. And the capture rate, as the measurement of the needs fulfillment, is a useful feature to divide helpful reviews from useless. But when using the feature in our AHV system, we will experience the problem of extracting the feature without any help from the restricted items <Er>, <Pt> and <Ps>. We will discuss this in Section 5.1.

Table 3: Uncertainties of main volitive auxiliaries

High (uncertainty)	Medium	Low
May, Maybe	Be worth	Must
Can, Could	Easy to	Have, Have to
Will, Would	Convenient to	Be ready to, Be able to
Should, Ought to	Be difficult to	Prefer to, Be willing to
Need, Need to	Hard to	Intend to, Want to, Wish to

4.2 Information Reliability

The hypothesis is that users prefer to vote for the reliable reviews because the reliability of information normally affects the correctness of purchase decision. In this Section, we focus on introducing two evidences of reliability based helpfulness voting: Volitive and Tense.

- **Volitive auxiliary based reliability determination**

At first, we think the volitive auxiliaries (along with the corresponding collocations) can give the evidence to prove the hypothesis. This is because the auxiliaries potentially represent the uncertainty (see Table 3) which can, to some extent, reflect the unreliability of reviews. Consider the reviews below:

(1) *I **prefer to** buy Sony, for it **may** have higher resolution*. (*9/68*)
(2) *It costs more than Nikon*. (*89/129*)

Compared to the review (2), the review (1) looks unreliable because without any definite information. Correspondingly, the review (1) receives a low voting rate. This shows the possibility of using the volitive auxiliary to prove the hypothesis (viz., reliability based helpfulness assessment). Therefore, we regard any sentence, which involves at least one volitive auxiliary, as an uncertainty, and use the number of uncertainties in a review to calculate the reliability score (see the equation in Figure 2). By calculating the average score for each type of Amazon product, we generated the curve diagram of average reliability scores on the 16 types of Amazon products with the corresponding 20,000 reviews, by which to compare the distribution of reliability scores between helpful and useless reviews. We show the curves in sub-figure (1) of Figure 2. Unfortunately, there is no clear discrimination between the curves. This is because the volitive auxiliaries normally have multiple pragmatic functions and senses (see the examples below).

(1) *If asked, the salespersons **have** (**volitive auxiliary**) said that they **had** (**notional word**) the detailed specification*. (Progmatics)
(2) *They **can** (means "**may**") come here to see this **can** (means "**container**") of moldy bacon? It is impossible*. (Sense)

Therefore it is not a proper measure to use the number of volitive auxiliaries to calculate the reliability score without considering pragmatics identification and word sense disambiguation. By simply using the syntactic structure (Stanford Parser[14]) and part-of-speech[15] to filter the notional words in the count of uncertainties, we improve the discrimination between the curves (see sub-figure 2 of Figure 2). Although a little weak, the volitive auxiliary indeed gives the evidence for the availability of information reliability in helpfulness assessment. But when using the feature in our AHV system, we still need to consider the influences of different levels of uncertainty in the measure of reliability. We will discuss this in Section 5.2.

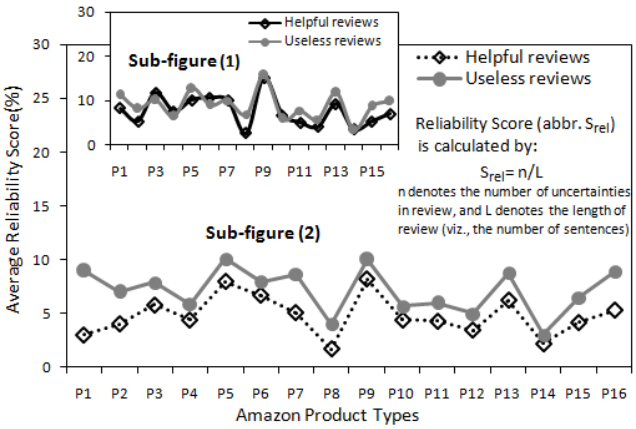

Figure 2. The reliability scores on Amazon reviews

- **Tense based reliability determination**

Secondly, we think the tenses (only past and perfect tenses here) can also give the evidence to prove the hypothesis of reliability based helpfulness voting. This is motivated by the psychology of users normally trusting the experienced reviewers and the habit of the reviewers often using past and perfect tenses in their writing. Actually the habit is reasonable because an experienced reviewer should firstly become an owner of product (normally the buyer in commercial websites) and so the corresponding opinions in reviews were mostly generated from the past experience. The direct evidence is the frequent occurrence of high voting rate in the reviews of buyers. By using the item <Buyer?> (see Table 1), we calculated the proportion of buyers in the reviewers. According to the calculation on 20,000 reviews, approximately 71.29% helpful reviews are written by the reviewers who claimed they were buyers (the item <Buyer?> == "Yes"), and only 39.05% useless reviews by visitors (<Buyer?> == "No").

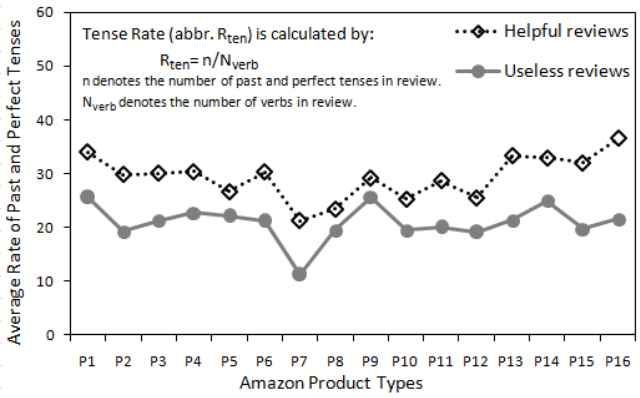

Figure 3. Tense Rate (only past and perfect tenses here) in Amazon reviews

On the basis, we calculated the rate of past and perfect tenses occurring in verbs for each review. And by calculating the average rate for each type of Amazon product, we generated the curve diagram of the average rates on the 16 Amazon products. The diagram shows that the curve of helpful reviews is easily distinguished from the useless (see Figure 3). Therefore, the tenses also give the evidence for the availability of information reliability in helpfulness assessment. As discussed in Section 3.2, we restrict the use of the item <Buyer?> to shield our AHV system from human intervention. So only the tenses, which can be obtained from the item <content>, will be used to measure the information reliability.

4.3 Mainstreaming Opinion

The hypothesis is that users prefer to vote for the reviews which are compatible with the mainstreaming opinions. In part, this is because mainstreaming opinion is normally authoritative and so more convincing.

The direct evidence to support the hypothesis is the low divergence between the star of helpful review and mainstreaming star (based on the data on Amazon.com). Here, the star (see Table 1) is a quantitative measure of product value, e.g. a five-star means "great value" but a one-star means "worthless". And the mainstreaming star is the star which occurs the most frequently in the reviews of specific product. On Amazon.com, a star is given by a reviewer through the interaction interface. Therefore, the star directly represents the opinion of reviewer on product value, and so the mainstreaming star, to some extent, reflects the mainstreaming opinion.

However, the use of the item <Star> is also restricted in our AHV system (see Table 1). The only way is to use the available item <content> to automatically detect the mainstreaming opinion and measure the divergence. The alternative for this is to adopt sentiment analysis to obtain a two-dimension opinion (positive polarity and negative polarity), and calculate the divergence based on the distribution of sentiment words in <content> of reviews. We will discuss this in Section 3.3.

5. FEATURE EXTRACTION

In this paper, we build an automatic helpfulness voting system for the reviews in common commercial websites. The system involves two functions. One is the helpfulness based review classification, and the other is review ranking. Thereinto, the classifier determines whether a review is helpful or useless, and the ranking system generates a ranking list of reviews based on their helpfulness scores. Because we directly use a current Ranking SVM model[12] to implement the review ranking, the remaining problem is only how to extract effective features for the classification. In this Section, we focus on introducing the extraction methods of our user-preference based features.

5.1 Capture Rate of Needs Fulfillment

As discussed in Section 4.1, the needs fulfillment is an important factor for helpfulness assessment, and the capture rate of product attribute and function is an effective measure of needs fulfillment. Therefore we use the rate as a feature in the classification, by which to support our AHV system.

Table 4: A type example of pseudo-feedback of Search Boss

Query_**"Nikon"**

```
<?xml> version="1.0" encoding="UTF-8" standalone="no" ?>
<ysearchresponse xmlns="http://www.inktomi.com/" responsecode ="200">
 <resultset_web count="20" start="0" totalhits ="25574196" deephits=…>
 +<result>
 </result>
 <result>
 …
    <keyterms>
      <terms>
         <term> Nikon Store </term>
         <term> Digital Cameras </term>
         <term> Digital SLR Cameras </term>
         <term> Zoom Lens </term>
         …
    </keyterms>
 …
 </result>
 …
```

The key issue of calculating the capture rate is to know what attributes and functions a product has. However, because the items <Er>, <Pt> and <Ps> are restricted to use (only <Product> and <Content> can be used), it is difficult to obtain the attributes and functions locally. An alternative way is to mine global information about product by using search engine. In this paper, we used the Yahoo!'s open search web services platform, named Search Boss[3], to mine the information. The Yahoo! Search Boss normally provides the keywords of pseudo-relevant feedbacks to a query, and if when the query is a product name, the keywords often contain plenty of product attributes and functions (see Table 4). Thus, according to the item <Product>, which denotes the product name, we can use the Search Boss to obtain the attributes and functions easily.

For each product, we use N_{fbk} top-ranked feedbacks as the global resources for the feature extraction. To obtain the optimal N_{fbk}, we randomly selected 20,000 Amazon products, and use their names as query to retrieve corresponding keywords (viz., the product attributes and functions) on the Yahoo Search Boss. And then we calculated the average overlapping rate of the keywords in editorial review (offered by the <Er> item) at every possible N_{fbk} (see Figure 4) and the average number of keywords increasing along with N_{fbk} (see Figure 5). Under the hypothesis that the editorial reviews (from domain experts) have enough product attributes and functions, by analyzing the trend curves of overlapping rate and keyword's number, we set the optimal number of N_{fbk} as 20 which just occurs at the first degression of the overlapping trend (see Figure 4) and that of the number of keywords. This illustrates that if we use more than 20 feedbacks (viz., N_{fbk}>20) to extract the keywords, we will have more noises in the keywords.

[3] http://developer.yahoo.com/search/boss/

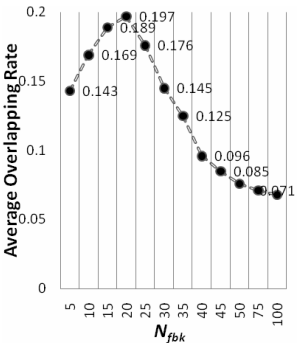

Figure 4. Overlapping trend

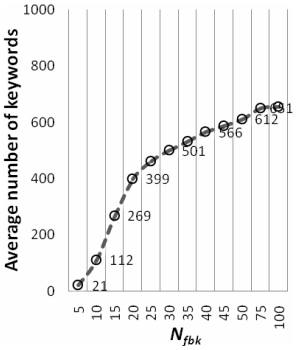

Figure 5. Increasing trend

Besides, the capture rate, as calculated in Section 4.1 (see the equation in Figure 1), neglects the distributions of product attributes and functions in global information resource. This will cause that some common attributes and functions generalize the capture-rate based measure of needs fulfillment. Therefore, according the distributions, we improve the calculation of capture rate by involving a combination weight. For an attribute or function, the weight is calculated as:

$$w_{cmb} = \alpha \cdot w_{key} + (1-\alpha) \cdot IG, \quad 0 \le \alpha \le 1 \tag{1}$$

where, α is a discount factor, IG means the mutual information of the attribute or function in training corpus, w_{key} denotes the weight in keywords. The w_{glb} is calculated as:

$$w_{key} = \frac{df}{N_{key}} \tag{2}$$

where, N_{glb} denotes the number of attributes or functions in global resource (viz., the top N_{fbk} pseudo-feedback of specific product), df denotes the number of pseudo-feedback that involving the attribute or function. On the basis, for a review, we calculate the improved capture rate as follows:

$$R'_{cap} = \frac{\sum w_{cmb} \cdot R_{cap}}{L} \tag{3}$$

where L denotes the length of review (viz., the number of words in review), R_{cap} denotes the original capture rate (see the equation in Figure 1).

In our AHV system, we use the improved capture rate as the measure of needs fulfillment. Besides, we expanded the words of product attribute and functions by using WordNet[4], by which to enhance the possibility of attribute and function word matching. And we also used the OpenNLP parser tool[5] to extract phrases from the item <content> of review.

5.2 Volition and Tense for Reliability

As discussed in Section 4.2, the information reliability is also an effective feature for helpfulness assessment. And both volition based uncertainty factor and tense based experience factor (past and perfect tenses only) can be used as the measures of reliability. In this Section, we respectively show the methods of calculating the factors.

- **Volition based uncertainty factor**

To calculate the factor, the first problem is to obtain enough priori volitive auxiliaries, by which to identify the volitive auxiliaries in content of review. Here, we adopt a cross-language collaborative mining algorithm[16] to obtain the volitive auxiliaries. The algorithm firstly use several English volitive auxiliaries as seeds, and by using a machine translation tool[17] translate the seeds into Chinese; secondly, because Chinese words can be divided into characters, and the character in Chinese volitive auxiliaries normally can combine with other characters to generate new volitive auxiliaries, the algorithm expends the Chinese auxiliaries in this way; at last it translate the new Chinese auxiliaries into English and add them into the list of seeds. The algorithm iteratively executes the operations until the number of English volitive auxiliaries never increases. After checking the auxiliaries manually, we obtained a total of 156 English volitive auxiliaries (including words and phrases).

We use the volition based uncertainty factor as the measure of information reliability, by which to support reliability based helpfulness assessment. The original uncertainty factor (see the equation in Figure 2) is calculate by the rate of sentences which involve at least one volitive auxiliary, within which syntactic structure (Stanford Parser[14]) and part-of-speech[15] is used to avoid the cases of multiple pragmatic functions and ambiguous sense. However, the original uncertainty factor neglects the influence of different uncertainty degrees of volitive auxiliaries. To improve the factor, we roughly divide volitive auxiliaries into three levels of uncertainty degree (see Table 3), and set the levels as the weighting coefficients: high uncertainty corresponds to 3, medium to 2 and low to 1, by which the improved uncertainty factor can be calculated as:

$$U_{vol} = \frac{\sum tf_v \cdot u_{max}}{\frac{n}{N_s}} \tag{4}$$

where N_s denote the total number of sentences in a review, n denotes the number of sentences which involve at least one volitive auxiliaries, u_{max} is the maximum uncertainty degree occurring in a sentence, tf_v denotes frequency of the verb adjacent to volitive auxiliary. By using tf_v, the uncertainty factor has the capacity of determining whether the uncertainty occurs at an event that a review emphasizes on (As the definition of ACE[18], an event normally is triggered by verb).

- **Tense based experience factor**

As discussed in Section 4.2, the experience factor is measured by the rate of past and perfect tenses occurring in verbs because the tenses, to some extent, reflect the possibility of reviewers having corresponding experiences to specific product. However, not all cases have the positive influence for the experience measurement. For example, the review "*I want to buy the camera because my wife* ***said*** *she* ***liked*** *its color*" has two past tenses, but only the verb "liked" can be regarded as the clue of usage experience (because "color" is normally the first user experience). Therefore, to improve the tense based experience factor, we only extract the verbs link to product attributes or functions in syntactic structure. Besides, we involve the frequencies of attributes and functions into the calculation of experience factor:

$$U_{exp} = \frac{\sum tf_p}{\frac{n}{N_v}} \tag{5}$$

where N_v denotes the total number of verbs in a review, n denotes the number of verbs link to attributes or functions in syntactic structure, and tf_p denotes the frequency of product attribute or function in review.

[4] http://wordnet.princeton.edu/
[5] http://incubator.apache.org/opennlp

5.3 Divergence from Mainstreaming Opinion

As discussed in Section 4.3, the divergence from mainstreaming opinion can be used to assess the helpfulness of review. Here, we propose a sentiment based divergence measurement, which only needs to use the item <content> of review.

To calculate the sentiment based divergence, the first issue is to obtain sentiment words. For this, we extract the sentiment words from WordNet Affect[19], which correspond to the WordNet synsets annotated with the six emotions: anger, disgust, fear, joy, sadness, surprise. Secondly, we roughly divided the words into two classes: positive polarity and negative polarity. Thereinto, positive-polarity class involves the words of joy, and negative-polarity class involves the words of anger, disgust, fear and sadness. But the surprise, as neutral emotion, is filtered. On the basis, we estimate the polarity of a review as:

$$P = \begin{cases} positive, \ if \ \dfrac{n_{pos} - n_{neg}}{L} > \theta \\ negative, \ else\ if \ \dfrac{n_{neg} - n_{pos}}{L} > \theta' \\ neutral, \quad else \end{cases} \tag{6}$$

where n_{pos} denotes the number of words of positive polarity, n_{neg} denotes the number of words of negative polarity, L denotes the length of review, θ is the threshold for positive polarity determination, and θ' denotes that for negative polarity determination. By using Benchmark Corpus[20] to train the thresholds, we got their optimal values as: θ equals to 0.02 and θ' equals to 0.015. Within the equation (6), we calculate the quantitative polarity force as:

$$F = \frac{\left| n_{pos} - n_{neg} \right|}{L} \tag{7}$$

After determining the polarity for each review, we detect the mainstreaming opinion for each product. Here, the mainstreaming opinion is either positive polarity or negative polarity. For each product, we respectively count the number of positive reviews and negative reviews, and calculate average polarity force for the two classes of reviews. And then we regard the polarity of the class which involves more reviews, as the mainstreaming opinion, and use the corresponding average polarity force as the force of mainstreaming opinion. On the basis, for each review, we calculate its divergence from mainstreaming opinion by measuring the difference between its polarity force and that of mainstreaming opinion.

6. EXPERIMENT SETUP

In this Section, we firstly introduce the dataset for our experiments, secondly we give the evaluation metrics, and at last we show the AHV systems to be tested.

6.1 Dataset

We focused our experiments on 124,878 reviews associated with Amazon products from the Multi-Domain Sentiment Dataset[6]. The dataset collected most products and reviews released on Amazon.com in 2006.

[6] http://www.cs.jhu.edu/~mdredze/datasets/sentiment/

Table 5. Number of reviews of every product type

Type(P1-P8)	Num	Type(P9-P16)	Num
Un MP3s & Cloud Player	956	Movie, Music & Games	3156
limited Instant Videos	3372	Electronics & Computers	2552
Amazon Cloud Drive	1288	Home, Garden & Tools	4012
Kindle	1296	Grocery, Health &Beauty	1188
Appstore for Android	2188	Toys, Kids & Baby	616
Digital Games & Software	586	Clothing, Shoes & Jewelry	1192
Audible Audiobooks	556	Sports & Outdoors	978
P8. Books	7216	Automotive & Industrial	3116

In most commercial websites like Amazon.com, there are a large number of duplicate reviews, which often negatively influence machine learning algorithms. For this, we use a simple deduplication method to filter the redundant reviews. The method matches bigrams between each pair of reviews. And a pair of reviews is deemed duplicated if they have more than 80% bigram matching. Besides the Amazon.com also have many duplicated products, such as the products can come in black or white models, and so reviews on such product are always duplicated. We filtered out the products whose all reviews are detected to be duplicates. The filtering process discarded a total of 3,404 products and 90,612 reviews. And at last we filtered out the reviews which never received any vote.

The final dataset involves 34,266 reviews on 1289 products. The types of products and the number of reviews in each type are shown in Table 5. We followed Kim et al [1] to set the voting rate 0.6 as the boundary between helpful and useless reviews and labeled the class (helpful or useless) of reviews beforehand. On the basis, we tested our helpfulness classification system on the whole dataset.

For testing AHV systems, we labeled helpfulness ranking for the reviews on the products of Digital Games, Audiobooks, Clothing and Sports (3,311 reviews on 116 products). In evaluation process, we run 5-fold cross validation and each fold use 80% products along with their reviews as training set and the rest as test set. We didn't adopt original Amazon ranking because Amazon ranks the reviews according to the timeliness but not the helpfulness.

6.2 Evaluation Metrics

In our experiments, we used accuracy to evaluate the performance of the helpfulness based review classification, and use the NDCG metric to evaluate the performance of our AHV system.

Here, the accuracy is the rate of the reviews whose helpfulness (helpful or useless) is correctly determined. NDCG@n is widely used to evaluate the performance of pseudo-feedback ranking in the field of information retrieval. In this paper, we translate it into an evaluation of review ranking system. NDCG@n[21] means normalized discount cumulative gain, which can take into account the influence of rank to accuracy. NDCG at rank n is calculated as:

$$NDCG@n = \frac{1}{Z_n} \cdot DCG@n = \frac{\sum_{i=1}^{n} \dfrac{2^{r(u_i)} - 1}{\log(1+i)}}{Z_n} \tag{8}$$

where i is the rank in the ranking list of reviews, Z_n is a normalizing factor and chosen so that for the perfect list DCG at each rank 1, and $r(u_i)$ equals 1 when u_i is a review whose helpfulness is correctly determined, else 0. In our experiments, based on the given voting rate by Amazon.com, we obtain the correct ranking list of reviews, and on the other side, AHV

systems output the ranking list of reviews based on helpfulness. Thus, we can use the NDCG@n to evaluate their performances.

6.3 AHV Systems

In our experiments, we totally built 10 systems: four helpfulness based review classification system, four AHV systems and two improved systems. The classification systems include:

- **System1** is a baseline classification system which roughly regards all reviews in corpus as helpful reviews. Because the average priori probability of helpful review occurring (in the 5 test set) is approximately 0.55, thus the accuracy of the baseline is 0.55.
- **System2** follows the textual features based helpfulness assessment of Kim et al[1]. The system uses the LIBSVM[7] with RBF kernel function (the rest parameters are default) as the classifier, and uses length, unigram and star as the features for classification.
- **System3** follows the work of Liu and Cao et al[5] and uses the same LIBSVM as the classifier but with the features of informativeness, readability and subjectivity.
- **System4** uses our user-preference based helpfulness assessment. The system also uses the LIBSVM as the classifier but with the features of needs fulfillment, information credibility and mainstreaming-opinion divergence. Thereinto, the needs fulfillment is measured by the capture rate (see equation 3), the credibility is measured by uncertainty factors (viz., volition based factor and experience based factor, see equation 4 and equation 5) and the divergence is measured by the difference between polarity forces (see section 5.3). Besides, we trained the discount factor α of the combination weight (see equation 1) in the measure of needs fulfillment, and set its optimal value to be 1.5, by which needs-fulfillment based classification can achieve best performance.

In our experiments, the AHV systems include:

- **System5** is a baseline AHV system which randomly ranks the reviews for each product.
- **System6** is a Ranking SVM based AHV system. Thereinto, Ranking SVM employs support vector machine (SVM) to classify object pairs in consideration of large margin rank boundaries. Here, we take pairs of reviews and their relative helpfulness derived from training data as training instances and apply Ranking SVM for learning better helpfulness assessment functions, by which to obtain optimal ranking list of reviews. Besides, we use the LIBSVM with the features of Kim et al[1] to implement the pair-wise classification.
- **System7** is a Ranking SVM based AHV system. But it uses the LIBSVM with the features of Liu and Cao et al[5] to implement the pair-wise classification.
- **System8** is also a Ranking SVM based AHV system. But it uses the LIBSVM with our features to implement the pair-wise classification.

Finally, by jointly using all of the features of Kim et al, Liu and Cao et al and ours, we implement an improved classification system **(System 9)** and an improved AHV system **(System 10)**.

7 http://www.csie.ntu.edu.tw/~cjlin/libsvm

7. MAIN RESULTS

We firstly run the helpfulness based classification systems. The accuracies of the systems are shown in Table 6. Here, our system (System4) achieved promising performance with the accuracy of 71.91%. Compared to the textual feature extraction of Kim et al (System2) and Liu and Cao et al (System3), our user-preference based feature extraction additionally contributes at least 6.77% correct helpfulness assessment.

Table 6. Accuracies of helpfulness-based review classification

	System1 (baseline)	**System2** (Kim)	**System3** (Liu)	**System4** (Ours)
Accurracy(%)	55.16	61.29	62.85	69.62

We individually used the user-preference based features and their combinations in the classifier, by which to inspect the respective contributions of the features. We show the corresponding accuracies in Table 7, within which "NF" denotes the feature of needs fulfillment, "IC_{vol}" denotes volition based information credibility, "IC_{exp}" denotes experience based information credibility, "MO" denotes divergence from mainstreaming opinion and the sign "↑" means the improvement when adding a feature into classifier. From the performances in Table 7, we can find two issues: one is that the classification accuracy when using the feature of "MO" is very low; two is that the joint use of the features of "MO" and "IC_{vol}" contributes the lowest improvement, and on the contrary, the joint use of "MO" and "IC_{exp}" contributes the most improvement.

Table 7. Respective contributions of user-preference based features and their combinations (accuracy)

Feature	**Accuracy(%)**	**Feature Combination**	**Accuracy(%)**
		NF+ IC_{vol}	61.68(↑3.03)
NF	58.65	NF+ IC_{exp}	64.04(↑5.39)
IC_{vol}	50.03	NF+MO	61.26(↑2.61)
IC_{exp}	51.15	MO+ IC_{vol}	52.66(↑1.51)
MO	43.43	MO+ IC_{exp}	56.19(↑6.16)
		IC_{vol} + IC_{exp}	53.43(↑2.28)

The reason for the first issue is that, in the calculation of divergence from mainstreaming opinion (viz., the feature of "MO"), the correctness of mainstreaming opinion detection is very important, and the correctness mostly relies on whether there are enough reviews to calculate a steady average polarity force. But actually the numbers of reviews for different products have very uneven distribution. Within the test dataset, there are approximately 25% products has only no more than 20 corresponding reviews. The curve of "MO" in Figure 6 illustrates the negative influence of sparse reviews for the "MO" based classifier. In the figure, the horizontal axis corresponds to different ranges of numbers of reviews a product received; the vertical axis corresponds to the average classification accuracies on different ranges. It is not hard to find, compared to other three features, the feature "MO" cannot help the classifier accurately determine the helpfulness of reviews.

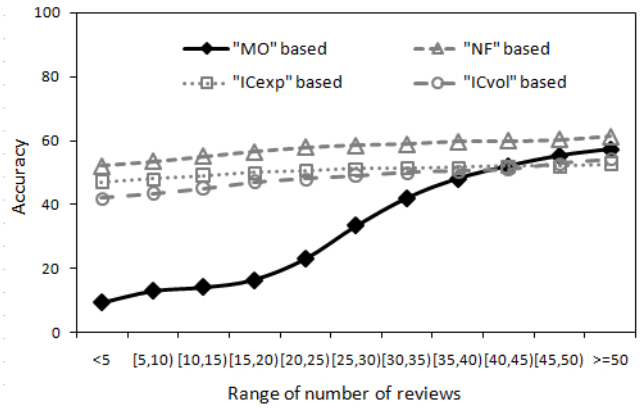

Figure 6. The accuracies of the classification systems in different ranges of numbers of reviews

The reason for the second issue is that the features "MO" and "IC_{vol}" make their classifiers have the same determination on most reviews. By analyzing the results, we found that approximately 79.2% reviews in test dataset receive the same helpfulness determination from "MO" based classifier and "IC_{vol}" based classifier. Thus, when adding the feature "IC_{vol}" into the "MO" based classifier, even if the feature has the ability to make better distinctions on review helpfulness, it has no enough space to offer this advantage. Compared to this, "MO" based classifier and "IC_{exp}" based classifier only have the same determination on 34.3% reviews. Thus the space for their respective superiority is bigger. The second issue also illustrate that the volition and sentiment normally have a compatible effect on reflecting user preference, e.g. people normally show positive sentiment to the things they preferred.

Table 8. NDCG@10 of AHV systems

	System5 (AHV baseline)	**System6** (Kim AHV)	**System7** (Liu AHV)	**System8** (Ours)
NDCG@10(%)	5.6	15.75	16.9	22.25

We secondly run the AHV systems on the test dataset. The performances of the systems are shown in Table 8. Here, NDCG@10 means the NDCG on the top 10 of ranking list, which needs that only a product with at least 10 reviews can be used as test sample. Thus we restricted the participation of products with less than 10 reviews in test, and used the rest (26,962 reviews) to evaluate the AHV systems. The results show that our AHV system achieves the optimal NDCG@10. Compared to the AHV systems of Kim et al and Liu and Cao et al, our AHV system respectively improves 6.5% and 5.35%.

At last, we run the final classification system (System9) and AHV system (System10), both of which use all existing features (including the textual features proposed by Kim et al, Liu and Cao et al, and our user-preference based features). The results show further improvements on helpfulness assessment and ranking (See Table 9).

Table 9. Performances of the improved systems

	System9 (improved classification)	**System 10** (improved AHV)
Accuracy	71.9	-
NDCG@10(%)	-	25.25

As discussed in Section 4, we use parts of items provided by Amazon.com (see Table 1) to illustrate the availability of user-preference based features in helpfulness assessment. Thereinto, the items <Er>, <Pt> and <Ps> are used to measure the feature of needs fulfillment, the item <Buyer?> is used to measure the feature of tense based information credibility, and the item <Star> is used to measure the feature of mainstreaming opinion consistency. And the capacity of the features in dividing helpful and useless reviews is obvious. But the features are restricted to use in AHV system to ensure pure automatic system without human intervention (our systems only use the items <Content> and <Product>). Here, we also use the measures to generate the review classification and AHV system for evaluating the compatibility of our only <Content> based feature extraction to the human intervention based feature extraction.

Table 10. Accuracy comparison of classification

	H(NF)	S(NF)	H(IC)	S(IC)	H(MO)	S(MO)
Accuracy	56.16	58.65	49.34	51.15	44.04	43.43

The performances of the systems are shown in Table 10 and Table 11, where "H" denotes the feature from human intervention, "S" denotes the feature only extracted from contents of review, "NF" denotes the feature of needs fulfillment, "IC" denotes information credibility (only based on tense here) and "MO" denotes mainstreaming opinion consistency. It not hard to find the performances are very compatible. The only issue is the performances of "S(NF)" and "S(IC)" are better than that by human intervention. The reason is that the measure of "NF" and "IC" have been improved based on the language information in <Content>, but it is hard for the items from human.

Table 11. NDCG@10 comparison of AHV

	H(NF)	S(NF)	H(IC)	S(IC)	H(MO)	S(MO)
NDCG@10	11.10	11.52	7.96	8.19	6.26	5.25

8. CONCLUSIONS

In this paper, we focus on discussing how to automatically assess review helpfulness and exploring the possibility of using user-preference based features to improve previous textual feature based helpfulness assessment. We respectively illustrate the availabilities of the features of needs fulfillment, information credibility and divergence from mainstreaming opinion in helpfulness assessment, and give the quantitative measures of the features. By using the features, we respectively build a helpfulness based review classification system and an automatic review ranking system. The test results on a large scale of commercial reviews (from Amazon.com) show the user-preference based features contribute substantial improvements for both review classification and ranking.

The improvements demonstrate that user preference learning is useful to review helpfulness assessment. In future, we will further explore the features that reflect the preference on commercial reviews. For example, users normally only read and vote on the reviews on the top of original ranking list by commercial websites, which will give steady voting rate to the top reviews but unreliable rate to the reviews on the bottom of ranking list. But, the similarity of contents can be used to detect the link between the relevant reviews wherever they locate. Therefore, the

similarity can be used to build a collaborative helpfulness assessment, which use the steady voting rate to estimate the weak rate of the reviews which are seldom or even never seen by users on the bottom of original ranking list.

9. ACKNOWLEDGMENTS

This research is supported by the National Natural Science Foundation of China (No. 60970056, 60970057, 61003152, 90920004), Special fund project of the Ministry of Education Doctoral Program (2009321110006, 20103201110021) and Natural Science Foundation of Jiangsu Province, Suzhou City (SYG201030).

10. REFERENCES

[1] Kim, S.M., Pantel, P., and Chklovski, T., Pennacchiotti, M. 2006. Automatically Assessing Review Helpfulness. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Sydney, July, 423-430.

[2] Titov, I., and McDonald, R. 2008. Modeling online reviews with Multi-grain Topic Model. In *Proceedings of the 17th international conference on World Wide Web*. Beijing, China, April, 111–120.

[3] Goldensohn, S. B., Hannan, K., McDonald, R., Neylon, T., Reis, G. A., and Reynar, J. 2008. Building a sentiment summarizer for local service reviews. 2008. In *Proceedings of WWW2008 Workshop on NLP Challenges in the Information Explosion Era*. Beijing, China, April, 21-25.

[4] Popescu, A. M., and Etzioni, O. 2005. Extracting Product Features and Opinions from Reviews. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. Vancouver, October, 339-346.

[5] Liu, J. J., Cao, Y. B., Lin, C. Y., Huang, Y. L., and Zhou, M. 2007. Low-Quality Product Review Detection in Opinion Summarization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (EMNLP-CoNLL 2007). Prague, Czech Republic, June, 334-342.

[6] Jinal, N., and Liu, B. 2008. Opinion Spam and Analysis. In *Proceedings of the international conference on Web search and web data mining*. Palo Alto, California, February, 219-230.

[7] Cristian, D. N. M., Kossinets, G., Kleiinberg, Jon., and Lee, L. 2009. How Opinions are Received by Online Communities: A Case Study on Amazon.com Helpfulness Votes. In *Proceedings of the 18th international conference on World Wide Web*. Madrid, Spain, April, 141-150.

[8] Tsur, O., and Rappoport, O. 2009. REVRANK: a Fully Unsupervised Algorithm for Selecting the Most Helpful Book Reviews. In *Proceedings of the Third International AAAI Conference on Weblogs and Social Media*. San Jose, California, May, 36–44.

[9] Liu, F., Yu, C., and Meng, W. Y. 2004. Personalized web search for improving retrieval effectiveness. *IEEE Transactions on Knowledge and Data Engineering* 16(1):28–40.

[10] Liu, Y., Huang, X. J., An, A., and Yu, X. H. 2008. Modeling and predicting the helpfulness of online reviews. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*. Pisa, Italy, December, 443–452.

[11] Jo, Y., and Oh, A. 2011. Aspect and Sentiment Unification Model for Online Review Analysis. In *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM Press, New York. February, 815-824.

[12] Cao, Y. B., Xu, J., Liu, T. Y., Li, H., Huang Y. L., and Hon, H. W. 2006. Adapting ranking svm to document retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* . Seattle, Washington, USA, August, 186-193.

[13] Valizadegan, H., Zhang, R., Zhang, R., and Mao, J. 2009. In *Proceedings of the 23rd Neural Information Processing Systems* (NIPS 2009). Lake Tahoe, Nevada, USA, December.

[14] De Marneffe MC., MacCartney B., and Manning CD. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of 5th International Conference on Language Resources and Evaluation*. Genova, May, 449-454.

[15] Giménez, J., and Màrquez, L.2004. SVMTool: A general POS tagger generator based on Support Vector Machines. In: *Proceedings of the 4th International Conference on Language Resources and Evaluation*, Lisbon, May, 43-46.

[16] Zhang, J. F., Hong, Y., Yang, Y. H., Yao, J. M., Zhu, Q. M. 2010. A grammar-based unsupervised method of mining volitive words. In *Proceedings of 2010 International Conference on Asian Language Processing*. Harbin, China, December, 137-141.

[17] Marcu, D., and Wong, W. 2002. A phrase-based joint probability model for statistical machine translation. In *Proceedings of Empirical Methods for Natural Language*. Philadelphia, PA, July, 133– 139.

[18] Doddington, G., Mitchell, A., and Przybocki. 2004. The Automatic Content Extraction (ACE) Program: Tasks, data, & evaluation. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*. In Memory of Antonio Zampolli. Lisbon, May, 837–840.

[19] Strapparava, C., and Valitutti, A. 2004. Wordnet-affect: an affective extension of wordnet. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*. Lisbon, Portugal, May, 1083–1086.

[20] Nasukawa, T., and Yi, J. 2003. Sentiment Analysis: Capturing Favorability using Natural Language Processing. In *Proceedings of Second International conferences on Knowledge Capture*. Sanibel Island, Florida, October, 70-77.

[21] Jarvelin, K., and Kekalainen, J. 2000. IR evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd international ACM SIGIR conference on Research and development in information retrieval*. Athens, July, 41-48.