Information Fusion 23 (2015) 3–15

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

An approach to rank reviews by fusing and mining opinions based on review pertinence

Jun-ze Wang [a,*], Zheng Yan [b,c], Laurence T. Yang [d,e], Ben-xiong Huang [a]

[a] Non-Traditional Security Center, Huazhong University of Science and Technology, China
[b] The State Key Laboratory of ISN, Xidian University, China
[c] The Department of Comnet, Aalto University, Finland
[d] The School of Computer Science, Huazhong University of Science and Technology, China
[e] The Department of Computer Science, St. Francis Xavier University, Canada

ARTICLE INFO

Article history:
Received 15 November 2013
Received in revised form 16 April 2014
Accepted 25 April 2014
Available online 9 May 2014

Keywords:
Review pertinence
Review spam
Retrieval model
Opinion fusion
Opinion mining

ABSTRACT

Fusing and mining opinions from reviews posted in webs or social networks is becoming a popular research topic in recent years in order to analyze public opinions on a specific topic or product. Existing research has been focused on extraction, classification and summarization of opinions from reviews in news websites, forums and blogs. An important issue that has not been well studied is the degree of relevance between a review and its corresponding article. Prior work simply divides reviews into two classes: spam and non-spam, neglecting that the non-spam reviews could have different degrees of relevance to the article. In this paper, we propose a notion of "Review Pertinence" to study the degree of this relevance. Unlike usual methods, we measure the pertinence of review by considering not only the similarity between a review and its corresponding article, but also the correlation among reviews. Experiment results based on real data sets collected from a number of popular portal sites show the obvious effectiveness of our method in ranking reviews based on their pertinence, compared with three baseline methods. Thus, our method can be applied to efficiently retrieve reviews for opinion fusion and mining and filter review spam in practice.

1. Introduction

The fast growth of the Internet has dramatically changed the way that people express their opinions. Nowadays, people can freely post reviews on articles at numerous websites to express their personal opinions. They can also freely share their attitudes and comments in online and mobile social networking. As the reviews express the subjective attitudes, evaluations, and speculations of people in natural language, this kind of contents contributed by Internet users have been well recognized as valuable information. It can be exploited to analyze public opinions on a specific topic or product in order to figure out user like or dislike, etc. Opinion fusion and mining are the methods to analyze and summarize opinions from reviews in order to comprehend public perspectives on a specific topic or an entity.

Research has been conducted in opinion fusion and mining with regard to sentiment analysis and opinion extraction from reviews. For example, Kim and Hovy collected past election prediction messages from the Web and applied an SVM-based supervised learning method to predict election results [1]. Analogously, Lin et al. developed a statistical model to capture how perspectives are expressed at document and sentence levels, and evaluated the model using the articles about Israeli–Palestinian conflict [2].

An important issue that was neglected in the past research is the degree of relevance between a review and its corresponding article. Due to the openness of Internet forums, anyone can write anything on it. The online reviews are mostly not equally relevant to the article. The irrelevant and less relevant reviews are "noisy" to some extend in the collection of reviews. If we can estimate the degree of relevance between a review and its article, we can eliminate the irrelevant reviews and pay little attention to the less relevant ones. As a result, we won't suffer from the negative effect of noisy reviews and can focus on most relevant reviews in opinion fusion and mining, thus the performance of fusion and mining can be greatly improved. However, based on our knowledge, there is no published study on this research topic. The most related work is review spam detection, which simply divides the reviews into two classes: spam and non-spam [3–5]. It is particularly noted that the degrees of relevance between different reviews and the article are in fact different – even for the non-spam reviews.

* Corresponding author. Tel.: +86 15071230213.
E-mail address: wangjunze@mail.hust.edu.cn (J.-z. Wang).

http://dx.doi.org/10.1016/j.inffus.2014.04.002
1566-2535/

In this paper, we propose a notion of "Review Pertinence". It is the degree of relevance of a review to its corresponding article. Obviously, different reviews have different review pertinence. The higher the pertinence, the more the opinion expressed in the review relates the article. Thereby, the reviews with higher pertinence are more useful or valuable for fusing and mining opinions on the topic of the article; while the reviews with lower pertinence are obviously less helpful for this purpose. Particularly, the reviews that have no pertinence to the article are so-called review spam. Obviously, estimating the review pertinence can be used to rank the reviews in order to pick up valuable reviews and eliminate invaluable ones or review spam. It is useful for fusing and mining the reviews in order to analyze and comprehend public opinions on a specific topic and reviewers' personal interests and preferences.

However, review pertinence estimation is not as easy as we thought. First, the reviews are usually short and mostly contain several sentences. Thus, it is hard to capture their intrinsic meanings. Second, the reviews and the article may use different words to present same concepts. Although traditional similarity measures (such as Jaccard Coefficient and Overlap Coefficient [6]) have been widely used to estimate the relevance between documents or sentences, these measures work poorly in a situation that has little word overlap.

This paper proposes a novel method for opinion fusion and mining by considering both the similarity between a review and its corresponding article and the correlation among reviews. Different from the prior art, we consider the correlations between reviews in the review pertinence estimation in order to overcome the shortcoming of similarity measures. We hold such a hypothesis that if review r_1 has high pertinence, review r_2 that is similar to review r_1 should also have high pertinence, even though the degree of similarity between review r_2 and the article is low. The effectiveness of our method is verified through a number of experiments based on real data sets collected from a number of popular portal sites by comparing it with three baseline methods.

The rest of the paper is organized as follows. Section 2 gives a brief review of related work. In Section 3, we analyze the issue of review pertinence estimation in details. Section 4 describes the proposed method. We show our experiment results in Section 5, followed by additional analysis and discussion in Section 6. Finally, conclusions and future work are presented in the last section.

2. Related Work

2.1. Opinion fusion and mining

It has been well recognized that user-generated contents contain valuable information about users. Fusing and mining opinions (positive or negative) from reviews has become a popular research topic in recent years [7–10].

Opinion fusion and mining are applied to extract public opinions on a product or specific topic. Researchers attempted to identify sentiments (i.e., the affective parts of opinions) in reviews [11,12], or classified online product reviews into positive and negative classes [13]. In [14], the authors presented a system that, given a topic, automatically finds the people who hold opinions on that topic and the sentiment of each opinion.

Some researches focused on decomposing or summarizing opinions from reviews. Lu et al. studied the problem of decomposing the overall ratings of a large number of short comments into ratings on some major aspects, so that a user can gain different perspectives of a target product [15]. Wang et al. analyzed opinions on an entity in an online review at the level of topical aspects to discover the latent opinion of each individual reviewer on an aspect, as well as relative emphasis on different aspects when forming the overall judgment of an entity [16]. Hu and Liu adopted semantic analysis techniques to mine and summarize all the customer reviews of a product [8].

The reviews were also used to analyze online public opinions to predict political events. Kim and Hovy presented an election prediction system named Crystal based on user opinions posted on an election prediction website [1]. Given a prediction request, Crystal first identifies which party is requested to predict, and then aggregates a large amount of opinions to provide election prediction.

Many methods have been used in the opinion fusion and mining field. Choi et al. used a global sequence model to classify and assign sources to opinions [17,18]. Mao and Lebanon used a sequential CRF (Conditional Random Fields) regression model to measure the polarity on a sentence level and determine the *sentiment flow* of authors in reviews [19]. Wei and Gulla proposed an approach to label the attributes of a product and their associated sentiments in product reviews through a Hierarchical Learning process with a Sentiment Ontology Tree [20]. Kazutaka et al. employed an unsupervised approach to extract the opinions on the aspects of a product (e.g., comfort and portability) in a summarization process [21]. A graph-based summarization framework (Opinosis) was proposed to generate concise abstractive summaries of highly redundant opinions [22]. A probabilistic rating inference framework, known as *Pref*, was proposed to mine user preferences from reviews and map such preferences into a numerical rating scale [23]. Potthast and Becker introduced OPINIONCLOUD, a technology to summarize and visualize opinions that are expressed in the form of Web comments [24]. Lin et al. investigated the problem of identifying the perspective expressed in a document [2]. They proposed a number of models to learn perspectives from the words used in a document with high accuracy.

Although the above techniques were applied and examined in different domains, an important issue that has been neglected so far is the degree of relevance between a review and its corresponding article. With review pertinence estimation we can improve the performance of opinion fusion and mining by focusing on relevant reviews and at the same time filter potential review spam.

2.2. Review pertinence

For opinion fusion and mining base on reviews, an implicit demand is that the reviews and the corresponding articles should be related.

Review spam detection can be used to filter out unrelated reviews. Review spam is an activity to introduce irrelevant information into reviews. It was firstly introduced by Jindal and Liu in [3]. They presented a supervised learning approach to detect the review spam. Jindal and Liu also studied the problem of opinion spam and the trustworthiness of online opinions in the context of product reviews [5].

However, current researches on review spam detection simply treat this issue as a binary classification task: either spam or non-spam decision is made. Almost all existing methods and models were proposed based on this classification. Few researchers pay attention to the task that rank the reviews that are not spam.

We argue that, even if the reviews are non-spam ones, the degrees of their relevance to the corresponding article are different. Obviously, prior arts lack incisive study on the issue of review pertinence. In this paper, we rank the reviews depending on their relevance with corresponding articles.

A number of researches related to our work have been conducted in the literature. Zhang and Varadarajan identified a research issue in [25]: predicting the utility score of product reviews to indicate their usefulness and merit. They viewed this issue as one type of regression and built a regression model to solve it. But this work only dealt with product reviews and the features of product need to be predefined. Different from this work, we deal with the reviews on news articles written in a nature language, and consider the relationship between a review and its article and the correlation among reviews in opinion fusion and mining. On the other hand, there is a considerable body of work on sentiment/aspect analysis of reviews in the literature, which has moved from binary assessment to continuous assessment. McDonald et al. investigated a structural model for jointly classifying the sentiment of text at varying levels of granularity [26]. In [27], the authors conducted an in-depth analysis on YouTube comments to explore the influence of sentiment expressed in the comments on the ratings of comments.

Collaborative filtering is one of important techniques for recommendations. Typically, a recommender system compares a user profile to some reference characteristics, and seeks to predict the 'rating' that a user would give to an item they had not yet experienced [28]. These characteristics may be from an information item (a content-based approach) or the user's social environment (a collaborative filtering approach) [29]. Our work on review pertinence study could provide a theoretic basis for improving the performance of extracting user preferences, interests and tastes from their reviews on articles (e.g., by avoiding the considerations on unrelated reviews), thus can serve as an input for recommender systems. The article review can be one of the reference characteristics for recommendations.

Little work in the literature conducted continuous assessment of review relevance, as what we study herein. In this paper, we propose the notion of "Review Pertinence" to study the degree of review relevance. We propose a strategy to assess the "Review Pertinence" by taking into account the correlation among reviews and integrating it into a measure based on the similarity between the review and its article. To the best of our knowledge, this has never been studied in the previous work. Our work is a useful supplement of prior arts.

3. Problem definition and analysis

We describe the review pertinence estimation as below:

If A denotes an article, R denotes its reviews, $R = \{r_1, r_2, \ldots, r_n\}$, r_i is the ith review on A. $Rel(r, A)$ denotes the degree of relevance of review r to article A. The review pertinence estimation is to rank the reviews in R according to $Rel(r, A)$. In the rest of this section, we firstly analyze the categories and structure of reviews, and then propose a basic strategy for review pertinence estimation.

3.1. Categories of reviews

We classify reviews into a number of categories for easy analysis.

3.1.1. Common review

Such reviews contain reviewer opinions on the topic of the article. Some of these reviews express a certain polarity (positive or negative).

3.1.2. Reviews in a new topic thread

Such reviews do not focus on the topic of the article. Instead, they start new threads that are inspired by the article. Such new threads may be irrelevant or less relevant to the topic of the article, and do not directly reflect the attitudes of reviewers on the article.

3.1.3. Comment review

Such reviews comment on other reviews, e.g., "I do not agree with the previous review."

3.1.4. Advertisement review

In such reviews, reviewers post some unsolicited advertisements or list some websites links and hope other users to access. The advertisement review could be irrelevant or relevant to the article.

3.1.5. Random text

Such reviews just contain some random text and are completely unrelated to the article.

For illustrating the categorization of reviews, we selected four reviews on an article about "prediction of house price in the next few months". They are originated from a popular website http://www.qq.com. We also extracted real data for our experiment from this website, as described in Section 5.

r_1: The decline of house price may be fantasy ...
r_2: If house price plummets, the price of commodity will inevitably undergo sharp rise.
r_3: I came to see the reviews, not the news.
r_4: XXX is a great product and can be bought on sale with a very cheap price at: YYY.com.

In the above four reviews, r_1 is a common review, r_2 is a review that starts a new thread, r_3 is a random text, and r_4 is an advertisement review. It is easy to figure out that r_3 and r_4 are irrelevant to the article, while both r_1 and r_2 are relevant to the article. But compared with r_2, r_1 is more relevant to the article since it directly expresses the opinion of the reviewer on the topic of the article.

3.2. Structure of review

A review is composed of several parts. We analyze the role of each part in the review pertinence estimation in order to propose an appropriate estimation strategy.

3.2.1. Information of reviewer

The information of reviewer includes the name or mostly nickname, or IP address of the reviewer, and so on. Most of websites show this information as a part of the review.

3.2.2. Review vote

In some websites, users can vote for existing reviews to express their opinions (e.g., for or against, support or oppose). The vote can also be treated as one part of the review.

3.2.3. Content of the review

The content of the review is the most important part. It has freely formatted text, by which users use a natural language to express their views and opinions. In some websites, a review may quote another review, thus form an explicit hierarchical structure in reviews.

Fig. 1 is an example review that contains the parts mentioned above.

3.3. Factors considered in review pertinence estimation

Among all these parts in a review, we analyze and justify appropriate contributors to the estimation of review pertinence.

First, many Internet websites reviewers post reviews anonymously, with nicknames that could be changed frequently. Thus, it is difficult to match a review to a specific reviewer. So we decide to focus on the textual information of the review.

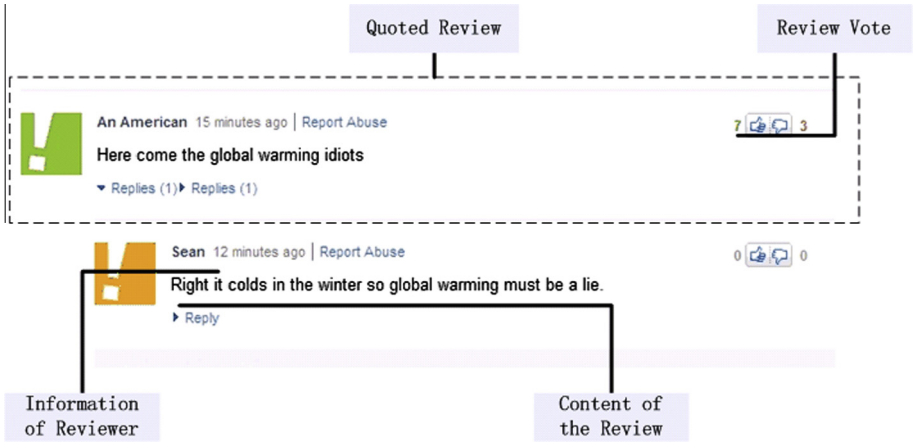

Fig. 1. A review example.

Second, considering the situation that the reviewer creates a new topic thread unrelated to the article, and then this review thread gets many comment reviews. This review – that may be less relevant or irrelevant to the article – could be quoted several times. In this case, quotation and quotation times do not contribute much to the review pertinence estimation. On the other hand, a reviewer could comment a review and create a new topic thread. Thus, it is very possible that the review with high pertinence could be commented by the reviews with low pertinence. Obviously, it is very possible that a review with high pertinence could be quoted by other reviews also with high pertinence. However, refer to Tables 1 and 2 that show the statistical information of our experimental data collected from real websites, only about 25.6% Chinese reviews and 33.8% English reviews have been quoted. These are not big proportions. Thereby, for simplification, we do not use the quotation times of a review to estimate the review pertinence in our proposed method.

Third, review voting reflects the attitudes of users on a specific review (e.g., for or against). As shown in Tables 1 and 2, about 53.7% Chinese reviews and 56.1% English reviews have been voted. But it is inappropriate to estimate review pertinence based on the votes obtained by the review. This is because the number of votes mainly implies the popularity of the opinion expressed in the review, which is not related to the pertinence. Because many websites order reviews based on the number of votes, top-listed reviews could be more easily noticed, thus easily voted by a new reader. Therefore these reviews could have more opportunities to keep their ranks continuously high. This is a kind of Matthew Effect [30]. Furthermore, as indicated in [31], a high proportion of votes to the reviews are actually only provided by a few of reviewers. This phenomenon called as "words of few mouths" implies that the number of votes is not an appropriate measure for the review pertinence estimation.

For further justifying the usefulness of voting number and quotation times in review pertinence estimation, we use the voting number and quotation times of a review as two baseline methods in our experiment to estimate review pertinence. The experimental result shown in Section 5 further proves the above analysis.

Based on the above discussion, we propose using the text content of the review in the review pertinence estimation. Applying this strategy, our method can be flexibly adopted in various situations. Obviously, the text content is the only common part of reviews in various Internet scenarios, while other parts are not always available. Integrating the number of votes to reviews into the final pertinence estimation model might be a more precise method than the one proposed in this paper. But as we discussed above, the number of votes could be affected by some other factors, e.g., the order of reviews. Thus, it could take more efforts to design a suitable integrated model, which could be too complicated to achieve good performance in practice. We leave this study in our future work.

Table 1
Statistical information of DATA-Chi.

Statistical information	
The total number of news articles	240
The total number of reviews	21,938
The average number of words in reviews	10.15
The number of reviews that were quoted	5615
The number of reviews that got votes	9104

Table 2
Statistical information of DATA-Eng.

Statistical information	
The total number of news articles	100
The total number of reviews	31,501
The average number of words in reviews	29.11
The number of reviews that were quoted	10,637
The number of reviews that got votes	17,662

4. Method of review pertinence estimation

A direct method treats the similarity between a review and the article as the review pertinence. Obviously, if a review is related to the article, the review probably contains the same words as those in the article in order to discuss the same topics or represent the same concepts. In this method the review pertinence can be denoted as a value between [0,1] that indicates the probability the review can be generated from the article based on their similarity. Thus, this method can distinguish the degree of relevance, rather than simply classify reviews as spam or non-spam.

In our method, we consider not only the similarity between a review and its corresponding article, but also the correlation among reviews. Suppose a review r is similar to another review that has a high degree of relevance to the article, then the review r should be also relevant to the article, even though it does not

have a high degree of similarity with the article. In what follows, we describe our method in details.

4.1. Similarity between review and article

As discussed above, the similarity between a review and its corresponding article may not comprehensively represent the relevance between them. But the similarity measured based on words overlap is still an important factor that should be considered to estimate the review pertinence. In our proposed method, we take the similarity between the review and the article into account.

Traditionally, the similarity between two documents (e.g., a review r and a corresponding article A) can be computed according to the Vector Space Model (VSM) [32] as the cosine of the inner product between their document vectors:

$$Sim(r,A) = \frac{\sum_i^n c(w_i,r)c(w_i,A)}{\sqrt{\sum_i^n c(w_i,r)^2}\sqrt{\sum_i^n c(w_i,A)^2}}, \quad (1)$$

where function $c(w, r)$ represents the times of term w appearing in r, $c(w, A)$ represents the times of term w appearing in A. $c(w, r)$ and $c(w, A)$ are the weights of terms w in the vector representations of review r and article A, respectively.

In the Vector Space Model, a document D is viewed as a point in a multi-dimensional vector space, denoted as (t_1, w_1; t_2, w_2; …; t_m, w_m). Herein t_i represents the term i appearing in D, and w_i represents the times of term t_i appearing in D, used to evaluate the importance of the term t_i in D. So the vector processing techniques can be used to compute the similarity between documents (i.e., reviews and their corresponding article).

Although the effectiveness of the Vector Space Model has been proved, there are still some shortcomings if using it directly for evaluating the similarity between the review and the article. Here we will discuss the strategy for improving the original model.

Firstly, the weight of term w in article A should be adjusted based on its importance in the article. For calculating the term's importance, we consider not only the position that the term appears in the article: in the title of the article, in the first or last sentence of a paragraph, but also the number of paragraphs that contain the term. It is easy to understand that the terms that distributed widely in the article and/or appear in the title or in the first/last sentence of a paragraph are probably the key terms of the article.

So we calculate the weight of term w in article A with the following method:

$$Weight(w,A) = c(w,A) * M * Pos(w), \quad (2)$$

where $Weight(w, A)$ denotes the weight of term w in article A, $c(w, A)$ represents the times of term w appearing in A, M denotes the number of paragraphs which contain term w. The value of $Pos(w)$ is set depending on the position of w:

$$Pos(w) = \begin{cases} 1.5, & \text{if } w \text{ appears in the title} \\ 1.3, & \text{if } w \text{ in the first or last sentence of a paragraphy, but not in the title} \\ 1, & \text{others} \end{cases} \quad (3)$$

Additionally, we note that this method suffers from zero probability, i.e., if w does not appear in A, $Weight(w, A)$ will be zero. That means if the review r does not share any terms with the article A, the probability of similarity between r and A will be zero. We use a data smoothing method to solve this problem by adding 1 to formula (2) and get the following formula:

$$Weight(w,A) = c(w,A) * M * Pos(w) + 1. \quad (4)$$

Secondly, many semantically similar concepts may be expressed with different words in reviews and their corresponding article, thus direct comparison using word-based Vector Space Model is not effective. In the literature, many existing models have been proposed to measure semantic correlation [33]. Herein, we propose discovering semantically similar terms using HowNet [34] and semantic similarity methods because applying them can calculate the semantic similarity of two terms in a more accurate way than other methods by considering all possible semantic concepts of a term.

HowNet is an authoritative ontology for both Chinese and English nature languages. Each word in HowNet links to several concepts, and each concept is represented by several primitive expressions separated by commas. The similarity between two words is defined as the maximum similarity of their corresponding concepts, and the similarity of two concepts can be calculated based on the similarities of their primitive expressions [35]. Thus, we have the following formula:

$$Semantic(w_1,w_2) = \max Semantic(c_{1i},c_{2j}), \quad (5)$$

where $Semantic(w_1, w_2)$ is the semantic similarity measure of the terms w_1 and w_2; c_{1i} is the concept of w_1, and c_{2j} is the concept of w_2.

$$Semantic(c_1,c_2) = \sum Semantic(p_{1i},p_{2j}), \quad (6)$$

where $Semantic\ (c_1, c_2)$ is the semantic similarity measure of the conceptions c_1 and c_2; p_{1i} is a primitive expression of c_1, and p_{2j} is a primitive expression of c_2. Liu has given a formula to measure the similarity between two primitive expressions [36]:

$$Semantic(p_1,p_2) = \frac{\alpha}{\alpha + dis(p_1,p_2)}, \quad (7)$$

where $Semantic\ (p_1, p_2)$ is the semantic similarity measure of the primitive expressions p_1 and p_2. $dis(p_1, p_2)$ is the semantic distance between two primitives. The semantic distance is defined as the path length of the two primitives in a semantic tree. The semantic tree is constructed according to the hyponymy between primitives. Herein, α is a variable parameter that is set as 0.5 in our experiments.

Based on the above justification, we design a final formula to calculate the similarity between a review r and the corresponding article A:

$$Sim(r,A) = \frac{\sum_i^n \sum_j^n c(w_i,r)Weight(w_j,A)Semantic(w_i,w_j)}{\sqrt{\sum_i^n c(w_i,r)^2}\sqrt{\sum_i^n c(w_i,A)^2}} \quad (8)$$

Formula (8) is an improved Vector Space Model. Here we take two new factors into consideration: the importance of a term in the article and the semantic similarity between two terms.

4.2. Correlation among reviews

The above similarity measure is based on word overlapping. However, it may not completely represent the relevance between the review and the article. Very possibly, a review that has high pertinence does not necessarily have a high degree of similarity with the article. Thereby, estimating review pertinence only based on the similarity measure may not be very accurate.

In order to overcome the above potential shortcomings, we further consider the correlation among reviews in the review pertinence estimation. On the basis of the cosine similarity among reviews, an undirected graph of reviews is constructed. In the graph, each node represents a review; its value denotes the review's pertinence to the article; the weight of the edge between

two nodes denotes the cosine similarity of the two corresponding reviews. If the similarity between two reviews is not zero, the corresponding nodes of the two reviews are connected as neighbors with each other.

Based on the above graph, we can calculate a review r_i's pertinence $Per(r_i, A)$ contributed by the correlation among reviews based on the Random Walk algorithm [37] with the following weighting scheme:

$$Per(r_i, A) = \sum_{r_j \in adj[r_i]} \frac{w(r_j, r_i)}{\sum_{r_k \in adj[r_j]} w(r_j, r_k)} Per(r_j, A), \quad (9)$$

where $adj[r_i]$ denotes the reviews that are neighbors of review r_i. $w(r_j, r_i)$ is the cosine similarity between r_j and r_i.

However, this method does not provide a strategy on how to assign the initial pertinence value of each node. Notably, formula (9) only shows an iteration process. It does not indicate how to end this process. In the best circumstance, after several iterations, the pertinence of each review will reach a static value and the iteration should end. In order to solve these issues, we combine this measure with the similarity measure described in Section 4.1 to get an integrated rational method as described in Section 4.3.

4.3. An integrated method

For constructing a more rational method, we integrate the two measures as described above together. We estimate the review pertinence by considering both the similarity of the review with the article and the correlation among reviews. We achieve an integrated formula as below:

$$Pertinence(r_i, A) = d \times \frac{Sim(r_i, A)}{\sum_{r \in R} Sim(r, A)} + (1 - d)\left[\sum_{r_j \in adj[r_i]} \frac{w(r_j, r_i)}{\sum_{k \in adj[j]} w(r_j, r_k)} Pertinence(r_j, A)\right], \quad (10)$$

where r_i is a review on article A, R is the set of all reviews on A, $Sim(r_i, A)$ denotes the normalized similarity between r_i and A based on (8). $Pertinence(r_i, A)$ denotes the degree of the relevance of r_i to A. Parameter d represents a damping coefficient, which controls the trade-off between the two items in the formula. $adj[r_i]$ and $w(r_j, r_i)$ have the same meanings as in formula (9).

With formula (10) we can iteratively update the pertinence value of each node. In accordance with the description in [38], if the value of $Sim(r_i, A)/\sum_{r \in R} Sim(r, A)$ is not zero, after several iterations the pertinence of each review will reach a static value. Thus, we can get stationary pertinence values of all nodes. As we have explained in Section 4.1, the smoothing technique adopted in the language model guarantees that the value of $Sim(r_i, A)/\sum_{r \in R} Sim(r, A)$ is not zero. This ensures that the iteration process can be terminated. Therefore, the relevance value of each review to the article (i.e., review pertinence) can reach stable after several iterations.

The details of convergence property of this method have been illustrated in [38]. It also provides a simple iterative algorithm, called power method, which can be applied to compute the stationary pertinence values of all nodes. The power method starts with assigning each node a random value. Then, all the node values are updated based on formula (10) during iteration. The detailed process is described in Algorithm 1. In this algorithm, we define its output as a vector p_k, which denotes the stationary pertinence values of all reviews after *kth* iteration. Threshold ε is used to control the termination of iteration. $||p_k - p_{k-1}||$ denotes the difference between p_k and p_{k-1}. If $||p_k - p_{k-1}||$ is smaller than the threshold ε, the iteration will be terminated automatically.

Algorithm 1. Stationary review pertinence computation

Input:
$Sim(r_1, A)$, $Sim(r_2, A)$, ……, $Sim(r_n, A)$: the similarity between reviews and the article A;
$w(r_i, r_j)$, $1 \leqslant i, j \leqslant n$: the cosine similarity among reviews;
ε: the threshold to control the termination of iteration.
Output: vector p_k: the stationary pertinence values of all reviews.
(1) set p_0 with a random vector;
(2) $k = 0$;
(3) **repeat**
(4) $k = k + 1$
(5) calculate the pertinence value of each review using formula (10);
(6) form vector p_k with the above pertinence values;
(7) $\delta = ||p_k - p_{k-1}||$;
(8) **until** $\delta < \varepsilon$
(9) **return** p_k

In Step (6), each review is viewed as a dimension in p_k. The pertinence value of each review represents the value of corresponding dimension.

Notably, the calculation of the correlation among reviews may have a drawback of high complexity, which will influence efficiency under a circumstance of massive data. One approach to solve this problem is to adopt parallel processing distributed to several computers. Canright et al. introduced a fully distributed method to conduct speedy and accurate power method calculation [39]. Previous work has also paid attention to improving the computation performance for similar computation to ours (e.g., for a PageRank model) by reducing the computation of a power method in an iteration process [40,41]. This is very helpful for improving the performance of our method. In the work presented in this paper, we mainly focus on improving the precision of review pertinence calculation. The strategies to speed up the calculation will be further studied in our future work in order to improve the performance of the proposed method in practice.

5. Experimental evaluation

5.1. Experimental data collection and preprocess

There are several popular portal sites in China, e.g., http://www.qq.com, http://www.sina.com, http://www.sohu.com. In these websites users can browse news articles and express their views and opinions. Everyday a large number of reviews are posted on these websites. Some of the reviews are posted anonymously, and some have the information of reviewers. The reviewers can also vote a specific review to express their attitude (e.g., for or against).

We developed a web crawler to obtain web pages from three of the most popular portal sites in China: http://www.qq.com, http://www.sina.com, and http://www.sohu.com. Then we used an html parser to analyze them. We collected 240 news articles and their reviews to form a data set denoted as DATA-Chi, which contains international news (20%), social news in China (35%), articles about military (12%), sports (10%), financial (8%), IT (9%) and other topics (6%). The time span of DATA-Chi collection is from May 1st, 2010 to May 15th, 2010. The reviews of 240 collected articles form a set denoted as Review-Chi. After segmenting Chinese words and removing stop words, we use DATA-Chi to evaluate our proposed method. Table 1 shows the statistical information of DATA-Chi.

We further obtained web pages from Yahoo website (http://news.yahoo.com), and used an English html parser to analyze them. We collected 100 English news articles and their reviews to form a data set denoted as DATA-Eng, which contains news about Internet (40%), politics (40%), technology and science (10%), and other topics (10%). The time span of DATA-Eng collection is from December 1st, 2013 to December 20th, 2013. The reviews of these articles form a set denoted as Review-Eng. Table 2 shows the statistical information of DATA-Eng.

5.2. Baseline methods and evaluation measures

5.2.1. Baseline methods

Since review pertinence has not been studied before, herein we adopted three different baseline methods defined by us in the experiment to show the efficiency and advantages of our method for review pertinence estimation.

(1) The first baseline method, denoted SimRank, views the similarity between review and article as the review pertinence by adopting an improved Vector Space Model to calculate the similarity.

As described in Section 4.1, the Vector Space Model can be used to compute the similarity between the review and its corresponding article to estimate the review pertinence. Briefly, the principle applied in SimRank is:

Principle 1: Consider review r, $Sim(r, A)$ denotes the similarity between review r and article A, and $Pertinence(r)$ denotes the degree of relevance between r and A. SimRank holds such a principle that for review r_1 and r_2, if $Sim(r_1, A) > Sim(r_2, A)$, then $Pertinence(r_1) > Pertinence(r_2)$; if $Sim(r_1, A) = Sim(r_2, A)$, then $Pertinence(r_1) = Pertinence(r_2)$.

(2) We estimate the review pertinence by using the voting number of review. This method is denoted as VotedRank. In most news websites, the reviews are ranked based on their voting numbers. Since we are also interested in whether the voting number can be used to evaluate the review pertinence, we used VotedRank as another baseline method.

In VotedRank, we define the principle of review pertinence estimation as below:

Principle 2: Consider review r, $v(r)$ denotes the voting number of r, $t(r)$ denotes the generation time of r (note that the older review will get a bigger $t(r)$ value), and $Pertinence(r)$ denotes the degree of relevance between review r and article A. VotedRank holds such a principle that for review r_1 and r_2, if $v(r_1) > v(r_2)$, then $Pertinence(r_1) > Pertinence(r_2)$; if $v(r_1) = v(r_2)$ and $t(r_1) < t(r_2)$, then $Pertinence(r_1) > Pertinence(r_2)$.

This principle is based on the following hypotheses: (a) if review r_1 gets more voting number than review r_2, then r_1 has higher pertinence than r_2; (b) if r_1 and r_2 have the same voting numbers, and r_1 is generated later than r_2, then r_1 should have higher pertinence, because r_1 gains the same number of votes in a shorter period of time.

(3) For the similar reason of adopting the VotedRank as a baseline method, we estimate the review pertinence by using the times of review quotation since they could imply the degree of review pertinence with the article. This method is denoted as QuotedRank. The principle of review pertinence estimation in QuotedRank is described below:

Principle 3: Consider review r, $q(r)$ denotes the quotation times of r, $t(r)$ and $Pertinence(r)$ have the same meanings as described above. QuotedRank holds such a principle that for review r_1 and r_2, if $q(r_1) > q(r_2)$, then $Pertinence(r_1) > Pertinence(r_2)$; if $q(r_1) = q(r_2)$ and $t(r_1) < t(r_2)$, then $Pertinence(r_1) > Pertinence(r_2)$.

This principle is based on the following hypotheses: (a) if review r_1 has more times of quotation than review r_2, then r_1 has higher pertinence than r_2; (b) if r_1 and r_2 have the same times of quotation, and r_1 is generated later than r_2, then r_1 should have higher pertinence than r_2, because r_1 gains the same times of quotation in a shorter period of time.

Our proposed method considers both the degree of similarity between the review and its corresponding article and the correlations among reviews. We denote it as Sim + CorRank. The principle applied in our method is:

Principle 4: Consider review r, $Pertinence(r, A)$ denotes the degree of the relevance between review r and article A, calculated based on formula (10). $Pertinence(r)$ has the same meaning as described above. Sim + CorRank holds such a principle that for review r_1 and r_2, if $Pertinence(r_1, A) > Pertinence(r_2, A)$, then $Pertinence(r_1) > Pertinence(r_2)$; if $Pertinence(r_1, A) = Pertinence(r_2, A)$, then $Pertinence(r_1) = Pertinence(r_2)$.

5.2.2. Evaluation measures

We treat the review pertinence estimation as an information retrieval task, which retrieves the reviews with high pertinence given an article. Note that traditional P@N and MAP measures can only handle cases with binary judgment, e.g., "relevant" or "irrelevant". These measures adopted a binary function $pos(i)$ to indicate whether an instance (i.e., review herein) at the rank i is positive (relevant). If the instance at the rank i is positive, $pos(i) = 1$; if not, $pos(i) = 0$. But with our method proposed in this paper, the pertinence value of a review is a numerical rating between [0,1]. So the traditional P@N and MAP measures cannot evaluate the performance of review pertinence estimation well. In this paper, we make use of an improved P@N measure and the Normalized Discount Cumulative Gain (NDCG) measure to evaluate the performance of the above methods. The improved P@N measure is defined as:

$$P@N = \frac{\sum_{i=1}^{N} rel(i)}{N}, \tag{11}$$

where N is the number of the reviews of a specific article. Function $rel(i)$ indicates the pertinence of the review at rank i, the value of $rel(i)$ falls into a scope between 0 and 1.

The *NDCG* measure can handle multiple levels of relevance judgments. It is suitable for evaluating the performance of review pertinence estimation. When evaluating a ranking list, *NDCG* follows two basic rules:

- Highly relevant reviews are more valuable than marginally relevant reviews;
- The lower ranking position a review (at any relevance level) stands, the less valuable it is for usage (e.g., opinion fusion and mining).

According to the above rules, the *NDCG* value is calculated as below when ranking the reviews of an article based on their pertinence:

$$NDCG@N \equiv Z_N \sum_{j=1}^{N} \frac{2^{rel(j)} - 1}{\log(1+j)}, \tag{12}$$

where N is the number of the reviews of a specific article, $rel(j)$ has the same meaning as in formula (11) and Z_N is a normalization constant used to make the NDCG score of a perfect review list be 1. We can see from formula (12) that $2^{rel(j)} - 1$ is the gain (G) of the jth review, $\frac{2^{rel(j)}-1}{\log(1+j)}$ is the discounted gain (DG), $\sum_{j=1}^{N} \frac{2^{rel(j)}-1}{\log(1+j)}$ is the

discounted cumulative gain (DCG) of the review list, and finally $Z_N \sum_{j=1}^{N} \frac{2^{rel(j)}-1}{\log(1+j)}$ is the normalized discounted cumulative gain (NDCG) of the review list, which is denoted as *NDCG@N*.

Obviously, high *P@N* and *NDCG@N* are desirable for retrieving top *N* ranked reviews with high quality, while low *P@N* and *NDCG@N* indicate good performance for retrieving bottom *N* ranked reviews.

5.3. Experiment results

In order to evaluate the effectiveness of the proposed method, we first manually rated the reviews based on their relevance to the article. A 5-likered scale was applied. Eight volunteers manually rated the reviews, indicated their level of agreement on the relevance of a review to the article with a number from 1 to 5 (1: totally not relevant, 3: partially relevant, 5: definitely relevant) respectively. Then we averaged their ratings on each review as its final rate and convert the scale into [0,1]. Thereafter, we ranked the reviews of each article according to their review pertinence calculated with four methods: SimRank, VotedRank, QuotedRank, and Sim + CorRank. We compared the performance of four methods in term of *NDCG@N* and/or *P@N* on all the reviews and top/bottom *N* rated reviews in the Review-Chi and Review-Eng datasets in order to show the effectiveness of our proposed method.

Particularly, Experiment 1 was designed to evaluate the overall performance of the above four methods. Experiment 2 and Experiment 3 were respectively conducted in order to evaluate the performance of our method with regard to two useful applications of review pertinence estimation. One is extracting the reviews with high pertinence for efficient and accurate opinion mining. The other is eliminating the reviews with low pertinence in order to filter review spam.

5.4. Experiment 1

This experiment was conducted to compare the performance of four methods in term of *NDCG@N*, with the datasets Review-Chi and Review-Eng. Through this experiment, we tested whether our proposed method Sim + CorRank can effectively rank the reviews according to the review pertinence. We also attempted to study the influence of the damping coefficient *d* on the performance of Sim + CorRank. Parameter *d* is applied to balance the contributions of the two items in the formula (10) for review pertinence computation.

The *P@N* measure was not used to compare performance in this experiment. This is because the sum of pertinence values of all the reviews on a given article is a constant. So we got the same *P@N* with regard to the whole reviews in Review-Chi or Review-Eng when using different methods.

5.5. Results based on dataset Review-Chi

Fig. 2 shows the results of average *NDCG@N* measures of all the articles in DATA-Chi, based on SimRank, VotedRank, QuotedRank, and Sim + CorRank. Herein, *N* is a variable that denotes the total number of reviews on a given article.

We observe that the Sim + CorRank method generally performs better than any of the other three methods. Fig. 2 also indicates the influence of parameter *d* on the performance of Sim + CorRank. For *NDCG@N*, Sim + CorRank performs better than the baseline methods when *d* is between 0.5 and 0.9, while Sim + CorRank performs best when $d = 0.7$. We also find that VotedRank and QuotedRank are not suitable for review pertinence estimation and QuotedRank performs the worst in all methods. This result supports our analysis presented in Section 3.3.

5.6. Results based on dataset Review-Eng

Fig. 3 shows the results of average *NDCG@N* measures of all the articles in DATA-Eng, based on SimRank, VotedRank, QuotedRank, and Sim + CorRank. Herein *N* has the same meaning as in Fig. 2.

We observe that the Sim + CorRank method generally performs better than VotedRank and QuotedRank, but does not show obvious advantages when comparing with SimRank method. Fig. 3 also indicates the influence of parameter *d* on the performance of Sim + CorRank. For *NDCG@N*, Sim + CorRank performs better than all the baseline methods only when *d* is 0,5, or between 0.7 and 0.9, while Sim + CorRank performs best when $d = 0.7$.

We also found that VotedRank and QuotedRank perform better when processing English reviews. The possible reason could be that the quality of reviews posted to Yahoo news website is better, and the reviews got more number of votes and more quotation than in Chinese news websites (refer to Table 2).

5.7. Experiment 2

This experiment was conducted to evaluate the performance of our method for retrieving the reviews with high pertinence.

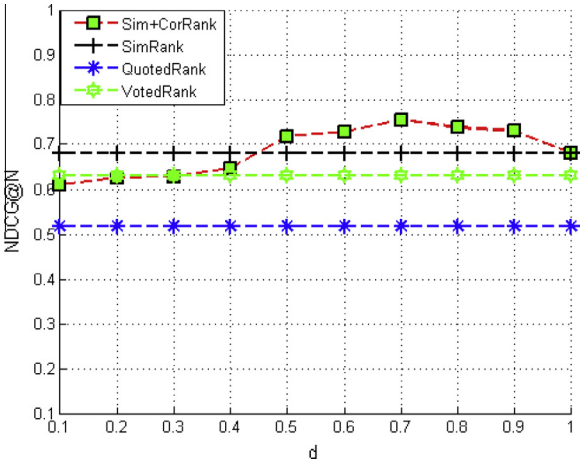

Fig. 2. Comparison of different methods in term of average "*NDCG@N*" of all the articles in DATA-Chi in retrieving reviews with high pertinence.

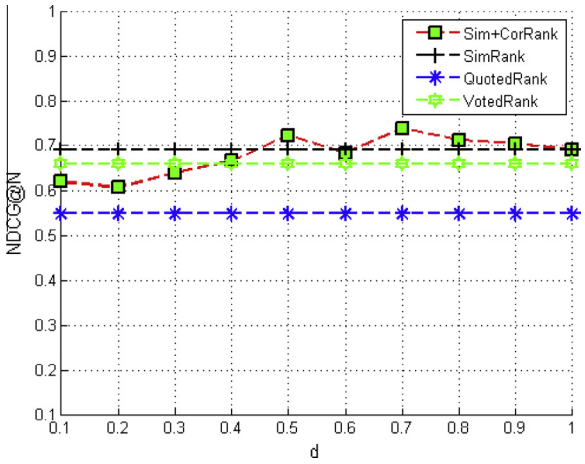

Fig. 3. Comparison of different methods in term of average "*NDCG@N*" of all the articles in DATA-Eng in retrieving reviews with high pertinence.

In Experiment 2, we got the top 20 reviews according to their pertinence for each article with four different methods: SimRank, Sim + CorRank, VotedRank and QuotedRank.

The reason for us to pick up 20 reviews for analysis lies in the fact that in most of the portal sites, such as www.qq.com, www.sina.com, www.sohu.com, www.ifeng.com, www.people.com.cn, www.china.com.cn, and even in www.weibo.com, 20 reviews are showed in the first web page of reviews and the users usually pay attention to the reviews in the first web page.

The top 20 reviews in Review-Chi formed a dataset denoted as Review-Chi-Top, and the top 20 reviews in Review-Eng formed a dataset denoted as Review-Eng-Top. Notably, the reviews in the two above datasets were also manually rated by four volunteers. Herein, we calculated the improved *P@20* and *NDCG@20* for Review-Chi-Top and Review-Eng-Top.

5.8. Results based on dataset Review-Chi-Top

Fig. 4 shows the results of *P@20* and *NDCG@20* regarding the top 20 reviews in Review-Chi, using the four methods. We can see that Sim + CorRank generally performs better than any of the other three methods. Fig. 3 also indicates the influence of parameter *d* on the performance of Sim + CorRank. We observe that Sim + CorRank performs better than the baseline methods when *d* is between 0.4 and 0.9 for *P@20* and between 0.4 and 0.9 for *NDCG@20*. We also observe that the Sim + CorRank performs best when *d* is around the scope [0.7,0.8], for both *P@20* and *NDCG@20*. So in both cases, a relative broad scope of suitable parameter values is observed.

5.9. Results based on dataset Review-Eng-Top

Fig. 5 shows the results of *P@20* and *NDCG@20* regarding the top 20 reviews in Review-Eng, using the four methods. We can see that Sim + CorRank performs better than QuotedRank, and Sim + CorRank, SimRank and VotedRank perform similarly. We observe that Sim + CorRank performs better than the baseline methods when *d* is between 0.6 and 0.7 for *P@20* and between 0.6 and 0.9 for *NDCG@20*. We also observe that Sim + CorRank performs best when *d* is around the scope of [0.6,0.7] for both *P@20* and *NDCG@20*. Obviously, VodedRank performs better for Review-Eng-Top than Review-Chi-Top.

We conclude this experiment that the reviews in Review-Chi with high pertinence can be effectively retrieved by the Sim + CorRank method. Although it does not perform as good as when processing Review-Chi, Sim + CorRank can retrieve the reviews with high pertinence in Review-Eng with at least similar performance to SimRank and VotedRank.

Based on Experiment 2, we suggest setting the parameter *d* around 0.7 for retrieving the reviews with high pertinence. This implies that we should pay more attention to the review similarity than the review correlation, but should not neglect the review

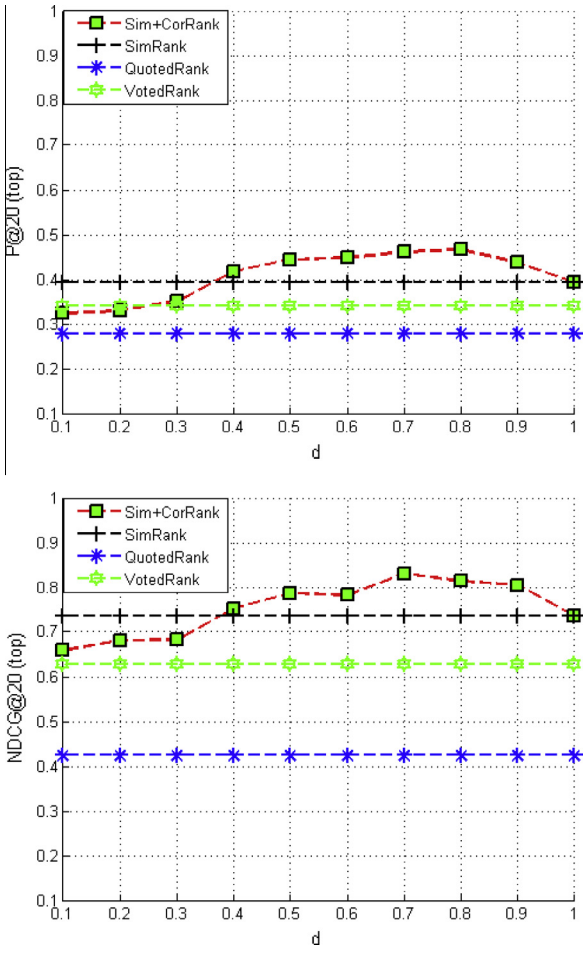

Fig. 4. Comparison of different methods in terms of average "*P@20*" and "*NDCG@20*" of all the articles in DATA-Chi, in retrieving reviews with high pertinence.

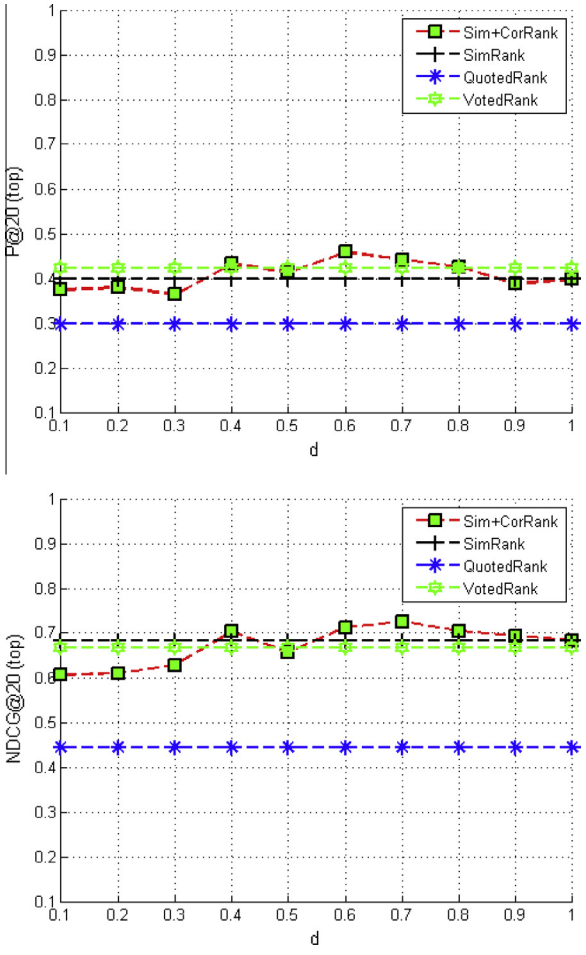

Fig. 5. Comparison of different methods in terms of average "*P@20*" and "*NDCG@20*" of all the articles in DATA-Eng, in retrieving reviews with high pertinence.

correlation in this case. The review correlation analysis can obviously improve the performance of retrieving the reviews with high pertinence.

5.10. Experiment 3

We conducted Experiment 3 to show that the reviews with low pertinence can also be effectively retrieved by the Sim + CorRank method.

Similar to the process described in Experiment 2, we got the bottom 20 reviews according to their pertinence. These reviews formed two datasets denoted as Review-Chi-Bottom and Review-Eng- Bottom. We calculated *P@20* and *NDCG@20* for the two datasets.

5.11. Results based on dataset Review-Chi-Bottom

Fig. 6 shows the results. Note that since we consider the bottom 20 ranked reviews, the lower the values of *P@20* or *NDCG@20*, the better the performance. From Fig. 6, we can see that Sim + CorRank generally performs much better than QuotedRank and VotedRank. The influence of parameter *d* on the performance of Sim + CorRank is also indicated in Fig. 6. Considering both *P@20* and *NDCG@20*, Sim + CorRank performs better than the baseline method SimRank when *d* is between 0.6 and 0.9, and Sim + CorRank performs not good when *d* is between 0.1 and 0.5. This implies that the review correlation should be considered less than the review similarity for retrieving the reviews with low pertinence. Based on Experiment 3, we suggest setting the parameter *d* around 0.7 and 0.8 for retrieving the reviews with low pertinence.

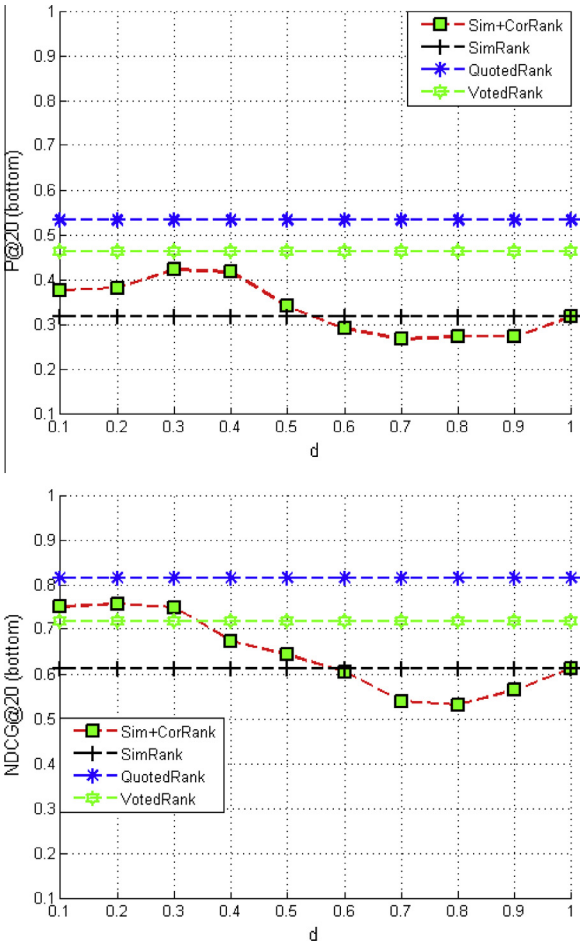

Fig. 6. Comparison of different methods in terms of average "*P@20*" and "*NDCG@20*" of all the articles in DATA-Chi, in retrieving reviews with low pertinence.

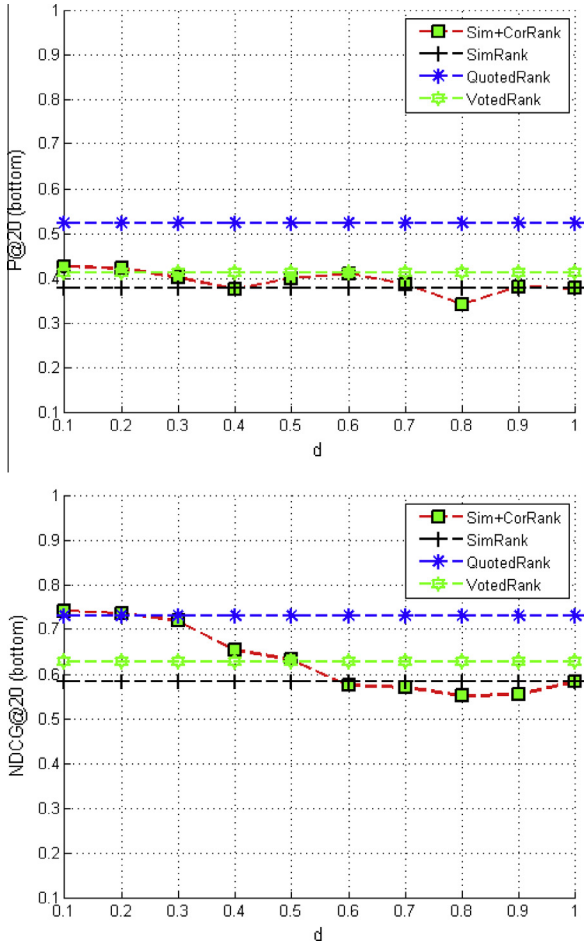

Fig. 7. Comparison of different methods in terms of average "*P@20*" and "*NDCG@20*" of all the articles in DATA-Eng, in retrieving reviews with low pertinence.

5.12. Results based on dataset Review-Eng-Bottom

From Fig. 7, we can see that Sim + CorRank generally performs better than QuotedRank and VotedRank, but worse than SimRank. Only when *d* is around 0.8, Sim + CorRank performs better than the other three methods. Considering both *P@20* and *NDCG@20*, Sim + CorRank performs not as good as when processing Review-Chi-Bottom. We will give more discussion on this issue in Section 6.

Based on Experiment 3, we suggest setting the parameter *d* around 0.8 and 0.9 in Sim + CorRank for retrieving the reviews with low pertinence. Further considering all above experimental results, we suggest setting the value of parameter *d* around 0.7 and 0.8 in practice.

6. Analysis and discussion

6.1. Experiment result analysis

Based on the experiment results, we find that Sim + CorRank may not perform well enough to precisely eliminate the reviews with low pertinence (i.e., review spam). Based on our observation, the reason could be (a) the reviews in a new topic thread share common words with the article, so they are similar with each other to some degree. In this case, Sim + CorRank gives high pertinence

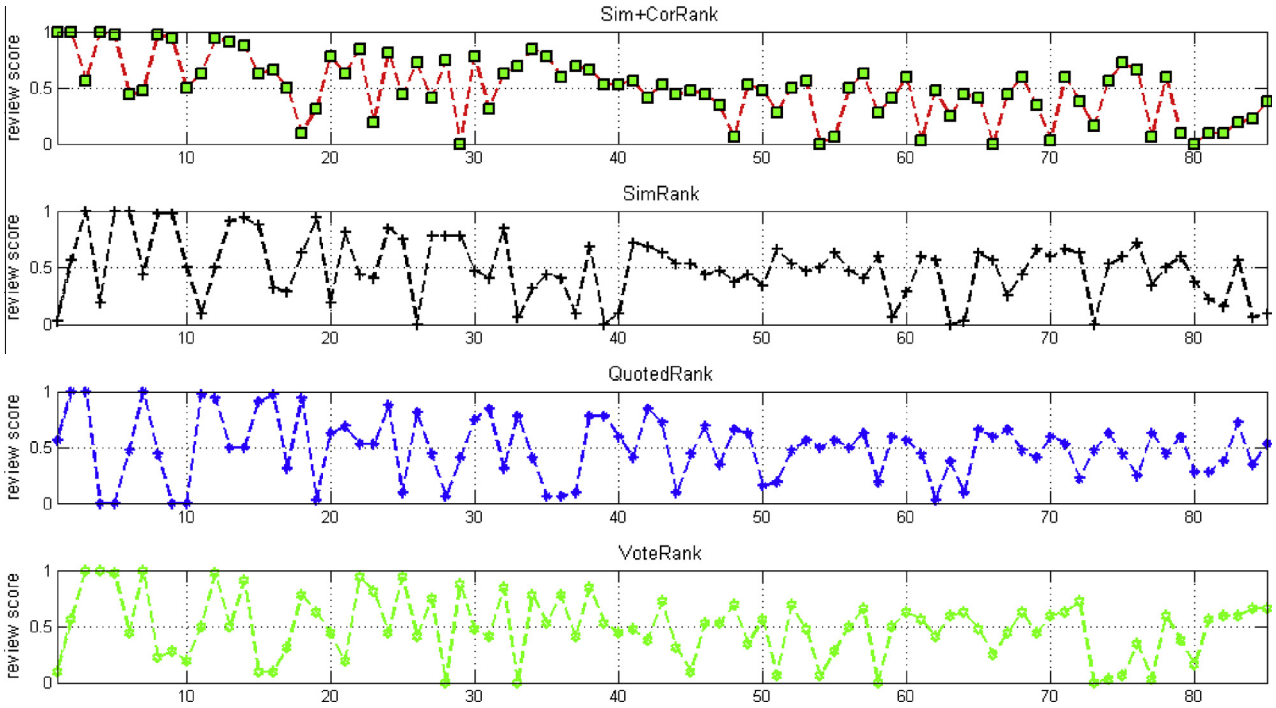

Fig. 8. An example of review pertinence estimation.

to this kind of reviews although they are not relevant to the article; (b) some advertisement reviews share common words with the article, thus this kind of reviews are given high pertinence by Sim + CorRank; and (c) some reviews are relevant to the corresponding article, but not similar to other reviews. Sim + CorRank may give low pertinence to this kind of reviews by mistake. How to improve the accuracy of Sim + CorRank to overcome the above issues is our future work.

Comparing Figs. 4 and 5 with Figs. 6 and 7, we find that Sim + CorRank performs better in experiment 2 than 3, with regard to both the *P@20* and *NDCG@20* measures. This implies that the Sim + CorRank method performs better when extracting the reviews with high pertinence than eliminating the reviews with low pertinence. With the following example we can observe this result clearly. We picked up one article from DATA-Chi, and respectively ran four methods: SimRank, Sim + CorRank, VotedRank and QuotedRank to rank its reviews (with a total number 85). Fig. 8 shows the result when $d = 0.7$. The x-axis stands for the position of the review ranked by the given method; the y-axis stands for the average rating of a review, rated by the four volunteers.

From Fig. 8, we can see that with the Sim + CorRank method, the reviews that were manually rated higher were basically ranked with high pertinence. But some of the highly rated reviews were ranked low. This implies that the Sim + CorRank method performance better when "extracting the reviews with high pertinence" than "eliminating the reviews with low pertinence". This is exactly the reason why Sim + CorRank performs better in experiment 2 than in experiment 3, with regard to both the *P@20* and *NDCG@20* measures. In addition, we also observe from Fig. 8 that Sim + CorRank performs better than other three baseline methods.

In all the experiments, Sim + CorRank performs worse when processing English reviews. We analyzed the reasons as below.

First, it is likely to use pronouns to represent nouns in English (such as subject and object). There are more short sentences and fewer pronouns in reviews presented in Chinese than in English. Our algorithm is based on the similarity computation between reviews and the similarity between reviews and their corresponding article. The similarity computation relies on the common terms used in documents. In English reviews, different nouns could be presented with the same pronouns, which could affect the computation on similarity.

Second, people tend to use different terms or words in English to express the same or similar meanings. But people rarely do so when writing in Chinese. One reason could be it is more readable to write in English with this way since English sentences are generally longer than Chinese ones. This could be a reason to impact the accuracy of similarity computation for English documents.

Finally, the word-formation in Chinese is different from English. Taking the term "biotechnology" as an example, it is related to terms "biology" and "technology". In English it is hard to find this kind of relationships. But in Chinese, we can easily find the relationship between "生物技术 (biotechnology)" and "生物 (biology)", or the relationship between "生物技术 (biotechnology)" and "技术 (technology)". Thus similarity computation is more accurate in terms of Chinese documents than English ones.

6.2. Further discussion

The experiment results show that the method proposed in this paper performs better than the baseline methods. It proves our hypothesis: the review pertinence is at least determined not only by the similarity between the review and its article, but also the correlation among the reviews.

The method (e.g., SimRank) based on a language model measures the similarity between the review and its article, but neglects the pertinence information that can be retrieved from the correlation among the reviews. Our method overcomes this shortcoming by integrating the above two aspects together.

Due to the "word mismatch" problem, if we use the traditional similarity measure models to measure the review pertinence, the results will not be very good. But applying the method proposed in this paper, even though a review r does not share any common words with the corresponding article A, it is still possible to connect it to other reviews that share common words with both

r and A. With this connection, the semantic relationship between r and A could be measured.

The ranking on reviews based on the review pertinence introduces a new way to order the reviews. The ordering could be different from a chronological order. In practice, ordering reviews based on pertinence except for the chronological order could affect user experience when reading the ranked reviews. Studying this effect could be an interesting research topic. More importantly, our method can be applied into extracting the reviews with high pertinence to analyze public opinions on an article: like or dislike and the arguments on a specific topic; eliminating the reviews with low pertinence to filter potential review spam. Thus, breaking the natural time sequence of the reviews will not affect the usability of these applications. Indicating the review pertinence in a usable way is a UI design and usability issue that is worth our further study.

Regarding the applications and usage of our study, our work can be applied into the area of Internet of Things [42]. For example, accurate opinion mining on some web articles and reviews about a product can provide a valid input to evaluate its quality and popularity, as well as find and group people with similar opinions on the same objects. Furthermore, non-spam reviews can also be used for analyzing user interests, preferences and taste, and thus it is possible to provide personalized recommendations in a precise way.

7. Conclusion and future work

In this paper, we proposed a novel method to estimate the review pertinence by considering both the similarity between the review and its article and the correlation among reviews. We further collected articles and their reviews in both Chinese and English from a number of popular websites to form our experimental datasets. Based on the datasets we compared the effectiveness of our method with three baseline methods. The experimental results show the advance of our method regarding ranking the reviews based on their pertinence. In addition, our method is the most efficient one in retrieving the reviews with high pertinence. It also performs well in retrieving the reviews with low pertinence. We also found that our method performs better in processing Chinese reviews than English ones. About the reasons, we proposed our justification.

Our work contributes to the literature in four folds: (a) introduced the notion of "review pertinence" to describe the degree of relevance between a review and its article. To the best of our knowledge, this is almost the first work to study this issue; (b) proposed a novel method to estimate the review pertinence; (c) designed and conducted a number of experiments based on real datasets in Chinese and English to show the effectiveness and advantages of our method and provide design implications on the parameter d; and (d) further analyzed the performance of our method, justified the reasons of occasional mistakes and discussed its suitability for different usage.

Regarding the future work, we will analyze the effect of other factors in the review pertinence estimation, especially the reasons that could cause estimation inaccuracy in some specific situations, as analyzed in Section 6.1. We will improve our method to overcome current shortcomings towards practical usage by not only improving computation efficiency but also applying it into services with significant business potential.

Acknowledgements

This work is sponsored by the PhD grant (JY0300130104) of Chinese Educational Ministry, the initial grant of Chinese Educational Ministry for researchers from abroad (JY0600132901), the grant of Shaanxi Province for excellent researchers from abroad (680F1303), and the fundamental research funds for the Central Universities of China (2013WQ035).

References

[1] S.M. Kim, E. Hovy, Crystal: analyzing predictive opinions on the web, in: Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural, Language Learning, 2007, pp. 1056–l064.
[2] W.H. Lin, T. Wilson, J. Wiebe, et al., Which side are you on? identifying perspectives at the document and sentence levels, in: Proceedings of the Conference on Natural, Language Learning, 2006, pp. 109–116.
[3] N. Jindal, B. Liu, Analyzing and detecting review spam, in: Proceedings of the 7th International Conference on Data Mining, 2007, pp. 547–552.
[4] N. Jindal, B. Liu, Review spam detection, in: Proceedings of the 16th International Conference on the, World Wide Web, 2007, pp. 1189–1190.
[5] N. Jindal, B. Liu, Opinion spam and analysis, in: Proceedings of the 1st Conference on Web Search and Data Mining, 2008, pp. 219–230.
[6] C.D. Manning, H. Schutze, Foundations of Statistical Natural Language Processing, Cambridge, The MIT Press, Massachusetts, 1999.
[7] K. Dave, S. Lawrence, D.M. Pennock, Mining the peanut gallery: opinion extraction and semantic classification of product reviews, in: Proceedings of the 12th International Conference on World Wide Web, 2003, pp. 519–528.
[8] M. Hu, B. Liu, Mining and summarizing customer reviews, in: Proceedings of the 10th International Conference on Knowledge Discovery and Data Mining, 2004, pp. 168–177.
[9] H. Tang, S. Tan, X. Cheng, A survey on sentiment detection of reviews, Expert Syst. Appl. 36 (7) (2009) 10760–10773.
[10] B. Liu, Sentiment Analysis and Subjectivity, second ed., Handbook of Natural Language Processing, 2010.
[11] P.D. Turney, Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews, in: Proceedings of the 40th Annual Meeting of the Association for, Computational Linguistics, 2002, pp. 417–424.
[12] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up? Sentiment classification using machine learning techniques, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2002, pp. 79–86.
[13] H. Cui, V. Mittal, M. Datar, Comparative experiments on sentiment classification for online product reviews, in: Proceedings of the 21st National Conference on Artificial Intelligence, 2006, pp. 1265–1270.
[14] S.M. Kim, E. Hovy, Determining the sentiment of opinions, in: Proceedings of the 20th International Conference on Computational Linguistics, 2004, pp. 1367–1373.
[15] Y. Lu, C. Zhai, N. Sundaresan, Rated aspect summarization of short comments, in: Proceedings of the 18th International Conference on World Wide Web, 2009, pp. 131–140.
[16] H. Wang, Y. Lu, C. Zhai, Latent aspect rating analysis on review text data: a rating regression approach, in: Proceedings of the 16th International Conference on Knowledge Discovery and Data Mining, 2010, pp. 783–792.
[17] Y. Choi, C. Cardie, E. Riloff, et al., Identifying sources of opinions with conditional random fields and extraction patterns, in: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2005, pp. 355–362.
[18] Y. Choi, E. Breck, C. Cardie, Joint extraction of entities and relations for opinion recognition, in: Proceedings of Empirical Methods in Natural Language Processing, 2006, pp. 431–439.
[19] Y. Mao, G. Lebanon, Isotonic conditional random fields and local sentiment flow, in: Proceedings of, Advances in Neural Information Processing Systems, 2007, pp. 961–968.
[20] W. Wei, J.A. Gulla, Sentiment learning on product reviews via sentiment ontology tree, in: Proceedings of the 48th Annual Meeting of the Association for, Computational Linguistics, 2010, pp. 404–413.
[21] S. Kazutaka, M. Yamaumi, R. Tadano, et al., Interactive aspect summarization using word-aspect relations for review documents, in: Proceedings of the 5th International Conference on Soft Computing and Intelligent Systems and 11th International Symposium on Advanced Intelligent Systems, 2010, pp. 183–188.
[22] K. Ganesan, C. Zhai, J. Han, Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions, in: Proceedings of the 23rd International Conference on Computational Linguistics, 2010, pp. 340–348.
[23] C.W. Leung, S.C. Chan, F. Chung, et al., A probabilistic rating inference framework for mining user preferences from reviews, World Wide Web 14 (2) (2011) 187–215.
[24] M. Potthast, S. Becker, Opinion summarization of web comments, in: Proceedings of Advances in, Information Retrieval, 2010, pp. 668–669.
[25] Z. Zhang, B. Varadarajan, Utility scoring of product reviews, in: Proceedings of the Conference on Information and Knowledge Management, 2006, pp. 51–57.
[26] R. McDonald, K. Hannan, T. Neylon, et al., Structured models for fine-to-coarse sentiment analysis, in: Proceedings of the 45th Annual Meeting of the Association of, Computational Linguistics, 2007, pp. 432–439.
[27] S. Siersdorfer, S. Chelaru, J.S. Pedro, How useful are your comments? analyzing and predicting YouTube comments and comment ratings, in: Proceedings of the 19th International Conference on World Wide Web, 2010, pp. 891–900.

[28] J.T. Hancock, C. Toma, N. Ellison, The truth about lying in online dating profiles, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2007, pp. 449–452.
[29] X. Su, T.M. Khoshgoftaar, A survey of collaborative filtering techniques, Adv. Artif. Intell. (2009).
[30] R.K. Merton, The matthew effect in science, Science 159 (1968) 56–63.
[31] R. Zhang, ·T. Tran, Y. Mao, Opinion helpfulness prediction in the presence of words of few mouths, World Wide Web 15 (2) (2012) 117–138.
[32] G. Salton, Automatic Text Processing: the Transformation, Analysis, and Retrieval of Information by Computer, Addison-Wesley, 1989.
[33] Y. Neuman, D. Assaf, Y. Cohen, Fusing Distributional and Experiential Information for Measuring Semantic Relatedness, Inf. Fusion 14 (3) (2012) 281–287.
[34] Z. Dong, Q. Dong, About hownet, <http://www.keenage.com>, visited on 08.06.13.
[35] Y. Guan, X. Wang, X. Kong, et al., Quantifying semantic similarity of Chinese words from HowNet, in: Proceedings of the International Conference on Machine Learning and Cybernetics, 2002, pp. 234–239.
[36] L. Qun, L. Sujian, Lexical semantic similarity computation based on HowNet, Computational Linguistics and Chinese Language Processing, 7(2) (2002) 59–76.
[37] L. Page, S. Brin, R. Motwani, et al., The PageRank citation ranking: bringing order to the web, 1999.
[38] S. Brin, L. Page, The anatomy of a large-scale hypertextual Web search engine, Comput. Networks ISDN Syst. 30 (1) (1998) 107–117.
[39] G. Canright, K. Engo-Monsen, M. Jelasity, Efficient and robust fully distributed power method with an application to link analysis, Technical Report, <www.cs.unibo.it/bison/publications/2005-17.pdf>.
[40] K. Sankaralingam, S. Sethumadhavan, J.C. Browne, Distributed PageRank for P2P Systems, in: Proceedings of the 12th IEEE International Symposium on High Performance Distributed Computing, 2003, pp. 58–68.
[41] C. Kohlschutter, P.A. Chirita, W. Nejdl, Efficient parallel computation of pagerank, Adv. Inf. Retrieval (2006) 241–252.
[42] Z. Sheng, S. Yang, Y. Yu, et al., A survey on the ietf protocol suite for the internet of things: standards, challenges, and opportunities, IEEE Wirel. Commun. 20 (6) (2013) 91–98.