Information Sciences 460–461 (2018) 519–540

Contents lists available at ScienceDirect

Information Sciences

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

How "small" reflects "large"?—Representative information measurement and extraction☆

Guoqing Chen[a], Cong Wang[a], Mingyue Zhang[a,b,*], Qiang Wei[a], Baojun Ma[c]

[a] China Retail Research Center, School of Economics and Management, Tsinghua University, Beijing 100084, China
[b] International Business School, Beijing Foreign Studies University, Beijing 100089, China
[c] School of Economics and Management, Beijing University of Posts and Telecommunications, Beijing 100876, China

ARTICLE INFO

Article history:
Received 31 March 2017
Revised 1 August 2017
Accepted 31 August 2017
Available online 5 September 2017

Keywords:
Representative
Coverage
Redundancy
Consistency
Compactness
Extraction

ABSTRACT

While web services avail a rapid growth of data volume for use, identifying helpful information is of great value, especially when users face with an unwilling glut of information. Thus, it is deemed relevant and meaningful to provide users with a representative subset (i.e., small set) that could well reflect the original information corpus (i.e., large set). In such a large–small context, this paper addresses the issues of representativeness in light of measurement and extraction by reviewing our previous efforts. Specifically, we first discuss various metrics from different perspectives of representativeness, then present a series of related representativeness extraction methods. Finally as a supplement and extension, a recent effort is introduced, which aims to take information quality into account in deriving a ranked subset. The proposed extraction method is justified by extensive real-world data experiments, showing its superiority to others in both effectiveness and efficiency.

1. Introduction

Increasing popularity of web services, such as search engines, product review platforms, and academic databases, has resulted in the so-called information explosion, especially in the forms of web texts [1]. For example, Google may return millions of relevant results that match the keywords a user inputs; a product may receive thousands of online reviews from past buyers; the academic databases may provide an abundance of articles within a certain research topic; and so on. Usually, users have limited time and processing capacity to read all pieces of information [2]. Moreover, the device functionality also restricts the presentation and navigation of the whole information set, such as limited screen size of mobile phones. Thus, only a small subset (i.e., "small") of the original information (i.e., "large") could be browsed and absorbed by users [3,4]. Without proper ways to measure and extract the subset of information, it may misguide users to have an incomplete understanding of original document set, giving rise to decisions of bad quality [5].

Therefore, it is desirable to provide users with a representative subset to grasp the main ideas of the original set of information. It motivates us to ask the following two questions with respect to the representativeness: (1) how to measure

☆ Partly supported by the National Natural Science Foundation of China (71490724/71372044/71402007/71110107027), and the MOE Project of Key Research Institute of Humanities and Social Sciences at Universities of China (12JJD630001).

* Corresponding author at: China Retail Research Center, School of Economics and Management, Tsinghua University, Beijing 100084, China.
E-mail addresses: chengq@sem.tsinghua.edu.cn (G. Chen), wangc3.14@sem.tsinghua.edu.cn (C. Wang), zhangmy@bfsu.edu.cn (M. Zhang), weiq@sem.tsinghua.edu.cn (Q. Wei), mabaojun@bupt.edu.cn (B. Ma).

http://dx.doi.org/10.1016/j.ins.2017.08.096
0020-0255/

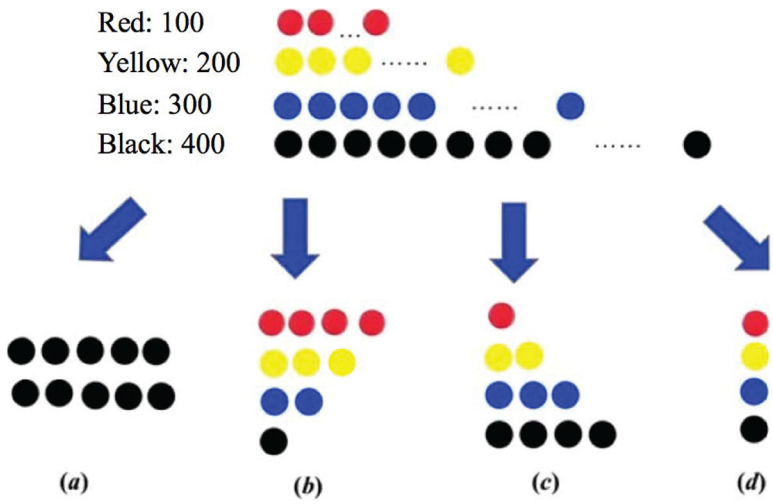

Fig. 1. Example of information extraction. (For interpretation of the references to colour in the text, the reader is referred to the web version of this article.)

the representativeness of a subset; and (2) how to extract a representative subset. In other words, the two questions can be described as an issue of how "small" reflects "large" with the measurement and extraction.

Literally, *representativeness* can be defined as the degree to which a small set of items reflects the diverse contents/semantics of a large information base [6]. In web search or information querying, though some ranking criteria (e.g., PageRank value, popularity, usefulness, or release time) are useful and widely adopted by online platforms and services [7–9], the resultant top-ranked outcomes are often found limited in effectively and sufficiently reflecting the diversity of all texts/documents. On one hand, the extracted subset should be as close as possible to the original set of information from the representativeness perspective of interest. On the other hand, redundant items in the extracted subset shall be restricted to an extent that conforms to the metrics of concern.

For illustrative purposes, let us consider an example. Given 1000 homogenous balls with four different colors, and there are 100 red ones, 200 yellow ones, 300 blue ones, and 400 black ones. Suppose that the balls in the same color are identical (i.e., their similarities equal 1) and the balls in different colors are completely different (i.e., their similarities equal 0). Fig. 1 gives four extracted subsets with different strategies. In result (a), only black balls are extracted. Such a subset is often observed in practice since the majority of the content is covered. However, it is not "similar" to the original set due to a failure of capturing other content information. In result (b), though it includes four kinds of colors, the proportion of each color is different from the original set. In result (d), only one ball in each color is extracted, which makes the set compact but not able to capture the structure information. The result (c) is considered appreciate since it covers not only the sufficient content information of the 1000 balls (i.e., all four colors), but also the structure information (i.e., similar distributions on different colors). It is worth mentioning that, here in Fig. 1, the representativeness problem is simplified such that the balls in the same color are identical. This assumption leads to some of the items in result (c) duplicated. However, things are more complicated in practice. For instance, suppose the original set consists of 1000 product reviews with 100 being positive, 200 being negative and 700 being neutral. If a user could only browse 10 reviews due to time or patience limit, then a subset with 1 positive, 2 negative and 7 neutral ones are preferred. In this case, the reviews in the same sentimental polarity are generally not identical but mutually similar in a degree. Thus, extracting multiple reviews in each polarity is meaningful.

In our previous studies [6,10–13], we have introduced several evaluation metrics from both the closeness and duplication perspectives. For closeness, three measurements are designed to assess the degree of information load, namely, information content coverage, information structure coverage and consistency. For duplication, two measurements are designed, namely, redundancy and compactness. Subsequently, representative information extraction methods have been developed to optimize the different metrics.

In sum, to address the issue of how small reflects large (also interchangeably referred to as the small–large issue), this paper will first discuss a number of existing metrics on representativeness including those introduced by the authors, as well as several extraction methods proposed by the authors with respect to certain metrics. Then, a new effort will be discussed, where a novel extraction method is presented for extending the consistency perspective in a more comprehensive manner in terms of information quality. For clarity, Table 1 highlights the respective metrics and methods that are covered in this paper. The reminder of this paper is organized as follows. Section 2 describes various major measurements of information representativeness, including the metrics proposed in our prior studies. Section 3 discusses three representative information extraction methods in light of information coverage, diversity, and consistency, respectively. Section 4 presents

Table 1
Summary of the measurements and methods.

Aspects	Metrics	Methods
Closeness	Content coverage	Cov_{C+S}-*Select*; $CovRed_{C+S}$-*Select*
	Structure coverage	Cov_{C+S}-*Select*; $CovRed_{C+S}$-*Select*
	Consistency	eSOP
Duplication	Redundancy	REPSET; $CovRed_{C+S}$-*Select*
	Compactness	$Comp_{fuzzy}$-*Select*
Quality (newly proposed)	Quality-aware consistency	$QCRR_{df}$

a new extraction method incorporating quality factor with respect to consistency measurement. Finally, Section 5 concludes the work and highlights future directions.

2. Measurement of representativeness

In light of representativeness, two aspects of a small set could be considered, namely, closeness and duplication. First of all, information in the extracted small set should be sufficient to reflect the information load of the original set. That is to say, a good representative subset achieves a high degree of closeness with the original set of information in terms of both content and structure. Moreover, highly duplicated information in the extracted small set can remarkably reduce users' satisfaction with information search services [14], and shall be restricted. Overall, as far as the metrics are concerned, the small set of representative results should have high closeness with the original set while possessing limited information duplication. In this section, some related metrics will be discussed including the ones introduced by the authors.

2.1. Existing metrics

From the closeness aspect, Pan et al. [15] proposed a metric called "coverage" which measures the percentage of classes that are covered by the representative set. A class is covered if and only if at least one of the items in the extracted subset belongs to that class. Let $C(S)$ be the distinct number of class labels covered by representative set S and $|C|$ be the total number of predefined class labels in the original set, then coverage is defined as:

$$coverage = \frac{C(S)}{|C|} \tag{1}$$

Zhai et al. [16] presented a subtopic retrieval problem and designed an $S-recall@K$ metric to evaluate the information representativeness. It is calculated with the percentage of the number of subtopics in all documents with respect to the original set. Specifically, suppose we have a topic T with n_A subtopics $A_1, \cdots, A_{n_A}$, and a ranking $d_1, \cdots, d_n$ of n documents. Let $subtopics(d_i)$ be the set of subtopics to which d_i is relevant. The *subtopic recall (S-recall) at rank K* is defined as:

$$S-recall@K = \frac{|\cup_{i=1}^{K} subtopics(d_i)|}{n_A} \tag{2}$$

The main limitation of the above two metrics is that they are dependent on predefined class labels, which is not always feasible in real-world settings. Further, Zhuang et al. [17] put forward two metrics to quantify the information extraction quality in the context of blog profiling. Though their metrics are free of predefined class labels, the information structure is still not taken into consideration.

From the duplication aspect, several approaches have been used to evaluate the duplication of one object with respect to a set, such as maximum similarity based [18], minimum similarity based [15], and average similarity based [17]. However, these approaches could hardly provide a solution that measures the overall duplication degree inside the extracted subset. Concretely, the duplication degree for a set (i.e., D) could be defined as the average value in terms of all objects (i.e., d_i) in the set. Thus, they can be respectively described as follows.

- Maximum similarity based:

$$Duplication(D) = \frac{1}{|D|}\sum_{d\in D}\max_{d_i\in D-\{d\}}\{sim(d,d_i)\} \tag{3}$$

- Minimum similarity based:

$$Duplication(D) = \frac{1}{|D|}\sum_{d\in D}\min_{d_i\in D-\{d\}}\{sim(d,d_i)\} \tag{4}$$

- Average similarity based:

$$Duplication(D) = \frac{1}{|D|(|D|-1)}\sum_{d\in D}\sum_{d_i\in D-\{d\}}\{sim(d,d_i)\} \tag{5}$$

Note that these metrics may cause some distortion. First, the maximum or minimum metrics to reflect the duplication degree of a set could be easily affected by a single outlier. Second, even if the average metric is adopted, the metric value would still be inconsistent with intuition in certain special cases. For instance, given an n-size set with n_a objects a and $n - n_a$ objects b, if n is large, $n_a = n - n_a = n/2$, the intuitively perceived duplication degree should be $\frac{n-2}{n}$, which is near 100%, whereas the average metric value is $\frac{n_a-1}{n-1}$, which is near 50%. Thus, it is considered necessary and meaningful to design more appropriate metrics to measure the duplication degree for a set.

2.2. Our introduced metrics

2.2.1. Closeness aspect

For closeness, we proposed several metrics, including content coverage, structure coverage and consistency [6,10].

(1) Content coverage

Suppose there is an original set $D = \{d_1, d_2, \cdots, d_n\}$ of n objects and an extracted small set $S = \{s_1, s_2, \cdots, s_k\}$ with size k (e.g., *tuples* in database or *documents* in web search results). Given two objects $d \in D$ and $s \in S$, s covers the content of d with degree $sim(s, d)$ and vice versa, where $sim(s, d)$ refers to the similarity measurement. The degree of a subset S covering the content of an original object d is determined by the object in the subset most similar to d, namely, $\max_{s \in S} sim(s, d)$. To measure the overall content coverage of small subset S with respect to original large set D, denoted as $Cov_C(S, D)$, we can use the arithmetic average operator to aggregate the content coverage degree of all objects in D.

$$Cov_C(S, D) = \frac{\sum_{d \in D} \max_{s \in S}\{sim(s, d)\}}{|D|} \tag{6}$$

Clearly, $Cov_C(S, D)$ possesses some useful properties. First, it is in the range [0, 1] and is reflexive, we have $Cov_c(D, D) = 100\%$ since S is able to cover all the information inherent in d if d appears in S. Second, it is monotonic, i.e., if $S' \subseteq S$, then $Cov_C(S', D) \leq Cov_C(S, D)$. Third, $k/n \leq Cov_C(S, D) \leq 1$ because at least k objects in D appear in S.

(2) Structure coverage

The structure coverage is modeled with information entropy [19] . Suppose that the n objects in original set D are classified into m distinct groups, where the number of objects in each group is n_j with $j = 1, 2, \cdots, m$ and $n_1 + n_2 + \cdots + n_m = n$. Then the information structure of set D can be expressed as its average distribution of information load via "information entropy":

$$-\frac{1}{\log m} \sum_{j=1}^{m} \frac{n_j}{n} \log\left(\frac{n_j}{n}\right) \tag{7}$$

where the default log base is 2. It is worth noting that when $n_1 = n_2 = \cdots = n_m$, we have the maximum value of overall information entropy, i.e., $\log m$. Thus, we divide $\log m$ to get the average information entropy in Eq. (7).

The intuitive idea to measure structure coverage of subset S to original set D is to calculate their information entropy with Eq. (7) respectively and compare them. However, this requires high computational complexity and also pre-determined class labels for each set. Thus, our previous work [6] simplifies the calculation using an assignment operation which can appropriately measure the information distribution in S with respect to that in D. Concretely, each object s in S could be treated as an virtual class label, resulting in k natural classes. Then, each d in D could be assigned into the corresponding class s if $d = s$ in a crisp sense. In real applications, the objects in a set could be search results, online reviews or other general documents. Thus, the assignment operation needs to be extended since the similarity of any two objects is in the range of [0, 1] rather than binary.

Basically, each d in D could be deemed to assign into the class with a label m where $m = \arg\max_{j=1,2,\cdots,k}(sim(s_j, d))$ with $m = 1, 2, \cdots, k$. Thus, the n objects in D could be assigned into k classes, denoted as D_1, D_2, $\cdots$, D_k, respectively. If document d is assigned to D_j, then d belongs to class D_j with degree of $sim(s_j, d)$, i.e., reflecting the extent of information of d covered in D_j where s_j is the natural label. Thus, the total "information load" in D_j is $\sum_{d \in D_j} sim(s_j, d)$, denoted as n_j^v, which is the cardinality with respect to the pieces of information covered in D_j to reflect the information in the original set D. Moreover, the total information load of all objects in D is $\sum_{j=1,2,\cdots,k} n_j^v$, denoted as n^v [20,21]. In this regard, the information structure coverage of subset S with respect to D, denoted as $Cov_S(S, D)$, can be constructed in spirit of information entropy, as illustrated below:

$$Cov_S(S, D) = \begin{cases} 1 & if\ k = 1 \\ -\frac{1}{\log k} \sum_{j=1}^{k} \frac{n_j^v}{n^v} \log\left(\frac{n_j^v}{n^v}\right) & if\ k > 1 \end{cases} \tag{8}$$

Note that we define the structure coverage with 1 when there is only one extracted object in the subset.

$Cov_S(S, D)$ also exhibits some useful properties. First, it is in the range (0, 1] and reflexive, i.e., $Cov_S(D, D) = 100\%$, since D can fully capture its own structure information. Second, if the information load in D could be conveyed with the equivalent distribution into S, then S preserves the best information structure. Third, if the information load in D could be assigned in a manner of more proximate distribution into S, the value of $Cov_S(S, D)$ would be higher. This property is very useful and important for designing better strategies for extracting a subset with higher structure coverage.

Furthermore, the content coverage metric and structure coverage metric could be aggregated to measure the information coverage from a combined perspective.

Information coverage. Given an original set D with n objects and an extracted subset S with k objects, where $S \subseteq D$, the information coverage of S in regard to D is defined as

$$Cov(S,D) = \begin{cases} Cov_C(S,D) = \dfrac{1}{n}\sum_{d\in D} sim(s_1,d) & if\ k=1 \\ Cov_C(S,D) \times Cov_S(S,D) = & \\ \dfrac{1}{n}\sum_{d\in D}\max_{s\in S}\{sim(s,d)\} \times \left\{-\dfrac{1}{\log k}\sum_{j=1}^{k}\dfrac{n_j^v}{n^v}\log\left(\dfrac{n_j^v}{n^v}\right)\right\} & if\ k>1 \end{cases} \tag{9}$$

where $n_j^v = \sum_{d\in D_j} sim(s_j,d)$ and $n^v = \sum_{j=1,2,\cdots,k} n_j^v$.

Here, the pairwise similarity metric (i.e., $sim(s,d)$) in the above definition should be carefully selected depending on different contexts. When only considering literal content for clustering, Cosine similarity metric is commonly used [3]. When the feature information is incorporated, Euclidean distance or Kullback–Leibner divergence-type metrics will be more suitable [3]. Furthermore, when sentiments or topics are considered in measuring the similarities of product reviews, some sentiment analysis methods with topic modeling [22] could be integrated.

(3) Consistency

The "consistency" metric is to measure the similarity between two sets when considering the feature and sentiment information in the documents. Given a set of documents $D = \{d_1, d_2, \cdots, d_n\}$, the corresponding set of features $F = \{f_1, f_2, \cdots,\}$ in the whole corpus and the set of sentiment orientations $SO = \{so_1, so_2, \cdots\}$ with respect to the features can be derived using available feature extraction and sentiment analysis methods, respectively [23–25]. The described settings can be easily found in real contexts, for instance, online reviews for products. In this regard, a document d, $d \in D$, can be represented as a set of feature-sentiment orientation tuples [23,26], namely, $d = \{(f,so)|f \in F, so \in SO\}$. Suppose we extracted a subset of the documents, denoted as S, $S \subseteq D$ (e.g., $S = \{d_1, d_3\}$), let $F_S \in F$ denote the features commented on in S. Apparently, we have $F_D = F$. For a feature f in F_S, let S_f denote the subset of S composed of the documents that have comments on feature f, i.e., $S_f = \{d|d \in S, f \in F_{\{d\}}\}$. Moreover, let S_f^{so} denote the subset of S_f with sentiment orientation so associated with feature f in each document, $S_f^{so} = \{d|d \in S_f, (f,so) \in d\}$.

Hence, the distribution of opinions regarding feature f on S can be represented as a vector, $d(f,S) = (|S_f^{so_1}|, |S_f^{so_2}|, \cdots)$, where $|S_f^{so}|$ is the number of documents in S_f^{so}. Furthermore, because *positive* and *negative* are the two most widely used sentiment orientations [23,25,27–29], here we define the set of sentiment orientations as $SO = \{+,-\}$, where + and − refer to the positive and negative sentiment orientations, respectively. Thus, the distribution information of a feature f on set S is represented by the proportions of positively opinioned documents among all opinioned ones on f, i.e., $|S_f^+|/|S_f|$. The consistency $cons_f(S,D)$ between S and D on feature f can be viewed as the consistency between two distributions, which can be measured with the absolute error of the proportions, namely, $1 - ||S_f^+|/|S_f| - |D_f^+|/|D_f||$. In addition, the weight of f can be measured by $|D_f|/|D|$ as the frequency of f being commented in all documents [26,30], which is to reflect the importance of providing consistent information on the feature f. Thus, the consistency on f can be formulated as

$$cons_f(S,D) = \begin{cases} 0 & if\ |S_f| = 0 \\ \frac{|D_f|}{|D|} \times \left(1 - \left|\frac{|S_f^+|}{|S_f|} - \frac{|D_f^+|}{|D_f|}\right|\right) & if\ |S_f| > 0 \end{cases} \tag{10}$$

Thus, we have the notion of consistency $Cons(S,D)$ between S and D formulated as the sum of consistency on all features:

Information consistency. Suppose the original set of documents is D and an extracted subset is S where $S \subseteq D$, the feature set mentioned in documents is denoted as $F = \{f_1, f_2, \cdots\}$ and corresponding sentiment orientation set is $SO = \{+,-\}$, the information consistency of S in regard to D is defined as

$$\begin{aligned} Cons(S,D) &= \sum_{f\in F} cons_f(S,D) \\ &= \sum_{f\in F_S} \frac{|D_f|}{|D|} \times \left(1 - \left|\frac{|S_f^+|}{|S_f|} - \frac{|D_f^+|}{|D_f|}\right|\right) \end{aligned} \tag{11}$$

Here, 1 is the reward for providing information on f, $||S_f^+|/|S_f| - |D_f^+|/|D_f||$ is the punishment for its inability to provide accurate information, and $|D_f|/|D|$ is the weight of f. We also have $0 \le Cons(S,D) \le \Sigma_{f\in F}|D_f|/|D|$, $Cons(D,D) = \sum_{f\in F}|D_f|/|D| = 1$ and $Cons(\varnothing, D) = 0$.

2.2.2. Duplication aspect

For duplication, two measurements are introduced: redundancy and compactness. In contrast to the "coverage" and "consistency" metrics, which compare two sets (i.e., they compare the original set D and extracted subset S), the "redundancy" and "compactness" metrics consider the objects and related relationships in a single set, such as S.

(1) Redundancy

Given two objects s_1 and s_2 in the extracted subset S, s_1 is called redundant with respect to s_2 with degree of $sim(s_1, s_2)$ because part of the information about s_2 has been duplicated by s_1 and vice versa. Further, we can use this idea to measure the extent to which an object s_1 is redundant in the set S. Formally, the degree to which s_1 is redundant in S, denoted as $Red(s_1, S)$ where $s_1 \in S$, can be represented as follows.

$$Red(s_1, S) = 1 - \frac{1}{\sum_{s \in S} sim(s_1, s)} \tag{12}$$

Concretely, $\Sigma_{s \in S} sim(s_1, s)$ in Eq. (12) refers to the total amount of redundant information associated with s_1 in S. Therefore, its reciprocal (i.e., $\frac{1}{\sum_{s \in S} sim(s_1, s)}$) represents the proportion of s_1's information in $\Sigma_{s \in S} sim(s_1, s)$. Then $Red(s_1, S)$ is the proportion of other objects' information that is duplicated by s_1. Thereafter, the degree of redundancy in the whole set S, denoted as $Red(S)$, could be described as follows.

Information redundancy. Given a set S with k objects, the information redundancy of S is defined as

$$\begin{aligned} Red(S) &= \frac{\sum_{s_i \in S} Red(s_i, S)}{|S|} \\ &= \frac{1}{k} \times \sum_{s_i \in S} \left(1 - \frac{1}{\sum_{s \in S} sim(s_i, s)}\right) \end{aligned} \tag{13}$$

$Red(S)$ also possesses some useful properties. First, $0 \le Red(S) < 1$ when S is not empty. Second, if $k = 1$, which means there is only one object in S, $Red(S) = 0$ indicating there is no redundant information. Third, if any two objects in S are mutually different, that is, for any $s_1, s_2 \in S$, $sim(s_1, s_2) = 0$, then we also have $Red(S) = 0$. Forth, if all the objects in S are identical, then $Red(S) = 1 - 1/k$, meaning that there is only $1/k$ information load in S that is not redundant.

As discussed in the previous section, high representativeness means high closeness and low duplication. Hence, we also proposed a measure *diversity* combining the "coverage" and "redundancy". In sprit of recall, precision and F_β [31], the combined measure $F_\beta(S, D)$ is defined as follows:

$$\begin{aligned} F_\beta(S, D) &= \frac{1}{\alpha / Cov(S, D) + (1 - \alpha)/(1 - Red(S))} \\ &= \frac{(\beta^2 + 1) Cov(S, D) \times (1 - Red(S))}{\beta^2 \times Cov(S, D) + (1 - Red(S))} \end{aligned} \tag{14}$$

where $\beta^2 = (1 - \alpha)/\alpha, \alpha \in [0, 1], \beta \in [0, \infty)$. $F_\beta(S, D)$ is a weighted harmonic mean of coverage rate and redundancy rate, where α or β reflects users' preference on coverage and non-redundancy. If $0 \le \alpha < 0.5 (\beta > 1)$, it means that users prefer more on non-redundancy than coverage for the extracted subset, and if $0.5 < \alpha \le 1 (0 \le \beta < 1)$, it means the opposite. If $\alpha = 0.5 (\beta = 1)$, it means that users treat coverage and non-redundancy equally. In addition, the combined measure has two properties. First, $0 \le F_\beta(S, D) \le 1$. Second, given a certain $\alpha(\beta)$, $F_\beta(S, D)$ increases monotonously with $Cov(S, D)$'s increase and $Red(S, D)$'s decrease.

(2) Compactness

This metric is introduced mainly for tuples in the context of structured data formats such as relational databases. Though semi-structured and unstructured data becomes dominant recently, sometimes we still need to transform the data into structured one to conduct further analysis. Thus, investigating the properties of structured data with traditional relational databases is relevant and useful. The compactness of a relation refers to the degree of non-redundancy. In the following, we discuss "compactness" in both the classical relations and possibilistic databases.

Similarly to the discussion about "structure coverage", we also use information entropy to define the set compactness. Suppose there is an extracted subset of relation S consisting of k tuples, which can be classified into m distinct groups, and the number of tuples in each group is k_j with $j = 1, 2, \cdots, m$ and $k_1 + k_2 + \cdots + k_m = k$. The probability of a random tuple belonging to the jth class, as well as the probability of jth class in S is k_j/k. The expected information entropy for classifying this given relation S is

$$-\sum_{j=1}^{m} \frac{k_j}{k} \log \frac{k_j}{k} \tag{15}$$

In classical relation $S = \{s_1, s_2, \cdots, s_k\}$, a distinct tuple can be viewed as a class with only one element, any tuples that are identical to each other could be assigned into the same class. Here, we have relation compactness as follows to describe the degree of a given relation being non-duplicated.

Relation compactness. Let $S = \{s_1, s_2, \cdots, s_k\}$ be a classical relation with k tuples which can be viewed as an extracted subset, S be divided into m classes according to tuple identity (i.e., every tuple in the same class is identical to each other), and k_j be the number of tuples in the jth class. Then the relation compactness of S is defined as

$$Comp_{classical}(S) = -\frac{1}{\log k} \cdot \sum_{j=1}^{m} \frac{k_j}{k} \log \frac{k_j}{k} \tag{16}$$

The degree of compactness reflects the extent to which the tuples of S are not duplicated, measured by the "amount of information" in S. In other words, the higher $Comp_{classical}(S)$ is, the less redundant the tuples in S, meaning that the more information S contains. Moreover, $Comp_{classical}(S)$ possesses some useful properties.

(1) We have $Comp_{classical}(S) = 1$ if and only if all the tuples in S are mutually distinct, i.e., $m = k, k_1 = k_2 = \cdots = k_m = 1$. This indicates that there is zero redundancy when all the tuples are unique.
(2) if and only if all the tuples in S are identical, i.e., $m = 1, k_1 = k$, we have $Comp_{classical}(S) = 0$.
(3) $0 \le Comp_{classical}(S) \le 1$.
(4) if m is fixed and $k_1 = k_2 = \cdots = k_m$, then $Comp_{classical}(S)$ decreases whenever k increases.
(5) if k is fixed, then $Comp_{classical}(S)$ decreases when any two of the classes merged into one class.

In addition, the fuzzy extension of compactness can be described as follows. Suppose there are k tuples with fuzziness involved in S, where these tuples can be not only identical/distinct to each other, but also similar to each other. A tuple s may not totally belong to the jth class, but belong to the jth class at a certain degree, denoted as $O_j(s) \in [0, 1]$. Thus, the concept of relation compactness can be extended by using $\Sigma_{s \in S} O_j(s)$ instead of k_j, denoted as k_j^v.

Fuzzy relation compactness. Given a fuzzy relation $S = \{s_1, s_2, \cdots, s_k\}$ of k tuples, where S can be divided into m classes, $O_j(s)$ represents the degree that tuple s belongs to the jth class ($j = 1, 2, \cdots, m$), and $k_j^v = \sum_{s \in S} O_j(s)$ is the Σcount operation as the "effective number" of tuples in the jth class. Let $k^v = \sum_{j=1,2,\cdots,m} k_j^v$, the fuzzy relation compactness in S is

$$Comp_{fuzzy}(S) = -\frac{1}{\log k^v} \sum_{j=1}^{m} \frac{k_j^v}{k_v} \log\left(\frac{k_j^v}{k^v}\right) \tag{17}$$

3. Representative information extraction methods

3.1. Coverage-oriented extraction

Aiming at providing as much information as possible in the extracted subset, we designed a heuristic method based on the idea of optimizing the information coverage measure as discussed in Section 2.2.1 [11]. Specifically, a two-step optimization strategy has been devised: (1) content coverage maximization along with an algorithm called Cov_C-*Select*, and (2) total information coverage (i.e., both content and structure) maximization along with an algorithm called Cov_{C+S}-*Select*. Moreover, we also incorporated a fast approximation strategy to improve the computation efficiency and proposed a corresponding algorithm, called $FastCov_{C+S}$-*Select*.

Concretely, first, we formulated the content coverage maximization problem and then designed a greedy algorithm to find a solution.

The maxContCov(k) problem. Given an original set $D = \{d_1, d_2, \cdots, d_n\}$, the similarity between any two documents in D (i.e., $sim(d_i, d_j)$) and an integer k (i.e., $1 < k < n$), the content coverage maximization problem (i.e., *maxContCov*(k)) is to find a subset of documents $S^* \subseteq D$ with $|S^*| = k$ such that

$$\begin{aligned} Cov_C(S^*, D) &= \max_{S \subseteq D, |S|=k} \{Cov_C(S, D)\} \\ &= \max_{S \subseteq D, |S|=k} \left\{ \frac{\sum_{d \in D} \max_{s \in S}\{sim(s, d)\}}{n} \right\} \end{aligned} \tag{18}$$

Though the desired objective of *maxContCov*(k) is a NP-hard problem, we can use a greedy strategy since it possesses the property of submodularity [32], which is stated below:

Submodularity. Given a finite ground set N, a set function $f : 2^N \to \mathbb{R}$ is submodular if and only if for all sets S, $T \subseteq N$, such that $S \subseteq N$ and $d \in N \backslash T$, $f(S + d) - f(S) \ge f(T + d) - F(T)$.

Thereafter, a naïve greedy algorithm called Cov_C-*Select* is proposed for computing a solution to *maxContCov*(k). In the algorithm, given the original set $D = \{d_1, d_2, \cdots, d_n\}$, let $S = \varnothing$ and D initially be the candidate set $D_{candidate}$. Each extraction step derives a result s^* in $D_{candidate}$ into set S, which makes the current value of $Cov_C(S \cup s^*, D)$ maximum and thus contributes to the largest marginal value of information content coverage. The procedures are repeated until $|S| = k$. Since the objective *maxContCov*(k) is submodular and the algorithm is in a greedy manner, we reach the conclusion that Cov_C-*Select* algorithm is a $(1 - 1/e)$-approximation algorithm for *maxContCov*(k). In addition, the computation complexity is $O(k^2 n^2)$.

Second, we describe the total information coverage maximization problem (i.e., *maxCov*(k)) based on the combined information coverage metric.

The maxCov(k) problem. Given an original set $D = \{d_1, d_2, \cdots, d_n\}$, the similarity between any two documents in D (i.e., $sim(d_i, d_j)$) and an integer k (i.e., $1 < k < n$), the total information coverage maximization problem (i.e., *maxCov*(k)) is to find

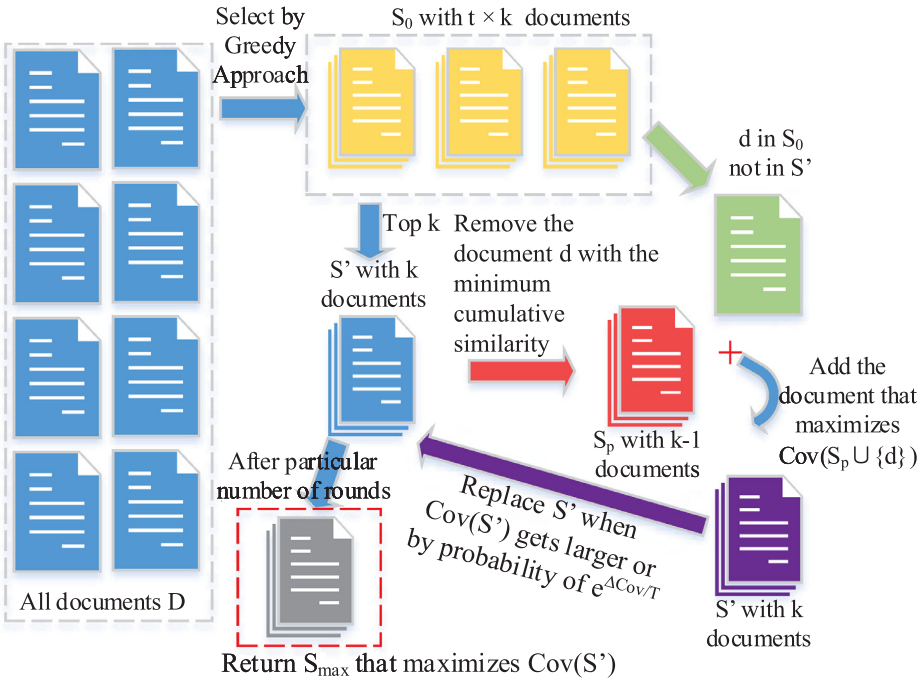

Fig. 2. Procedures of Fast-Cov method.

a subset of documents $S^* \subseteq D$ with $|S^*| = k$ such that

$$Cov(S^*, D) = \max_{S \subseteq D, |S|=k} \{Cov_C(S, D) \times Cov_S(S, D)\}$$
$$= \max_{S \subseteq D, |S|=k} \left\{ \frac{\sum_{d \in D} \max_{s \in S}\{sim(s, d)\}}{n} \times \left\{ -\frac{1}{\log k} \sum_{j=1}^{k} \frac{n_j^v}{n^v} \log\left(\frac{n_j^v}{n^v}\right) \right\} \right\} \quad (19)$$

The function of $Cov(S, D)$ is not submodular which could be proved by contradiction. Thus, we use the idea of simulated annealing to find a solution. It normally begins from a very high initial cooling temperature and makes use of stochastic search strategy as the temperature drops to reach global optimization. We proposed a heuristic algorithm called *FastCov*$_{C+S}$*-Select* to efficiently solve the *maxCov*(k) problem whose procedures are presented in Fig. 2. The core idea lies in that, it is more likely to obtain the extracted subset with higher total information coverage in optimizing the structure coverage on the basis of a subset with significantly high content coverage. Given the original set D of size n, the extracted size k, the initial solution S_0, a set of $t \times k$ documents from the output of *Cov*$_C$*-Select*($t \times k$), where t is a predefined small integer, the similarity measurement between any two documents $sim(d_i, d_j)$ as well as the initial cooling temperature and final temperature, the memorial variable S_{max} is used to record the best solution with the highest value of coverage at the time after each iteration. Particularly, each iteration procedure is composed of four steps in the algorithm *FastCov*$_{C+S}$*-Select*: (1) New solution generation from current state; (2) Calculating difference of *Cov* values between new solution and current status; (3) Judgment on whether to accept the new solution; (4) Updating relevant variables. In this way, the computational complexity is $O(tk^2n^2) + O(T_0tk^2n)$ with $t \times k \ll n$. The effectiveness and efficiency of *Cov*$_C$*-Select* and *FastCov*$_C$*-Select* as well as their outperformances over other related methods have also been justified with extensive real-world data experiments (i.e., 3,500,000 snippets from Google search results) and human evaluations.

3.2. Diversity-oriented extraction

Here we discuss a method named REPSET(REPresentative SET) for identifying a small set of documents which largely represent the diversified content information in light of *coverage* and *redundancy*. Particularly, REPSET incorporates a novel document clustering technique followed by an extraction procedure, which consists of two major steps: (1) categorizing all the documents based on their similarities of content between each other; and (2) selecting a good representative document for each category. Accordingly, the overall framework of REPSET is shown in Fig. 3. As illustrated, in the first step, the degrees of similarity between documents are calculated. Next, a clustering algorithm is used to cluster the documents into a document dendrogram with different granularities of clusters. Finally, for all given levels in the document dendrogram, a representative document is selected from each cluster according to the similarity.

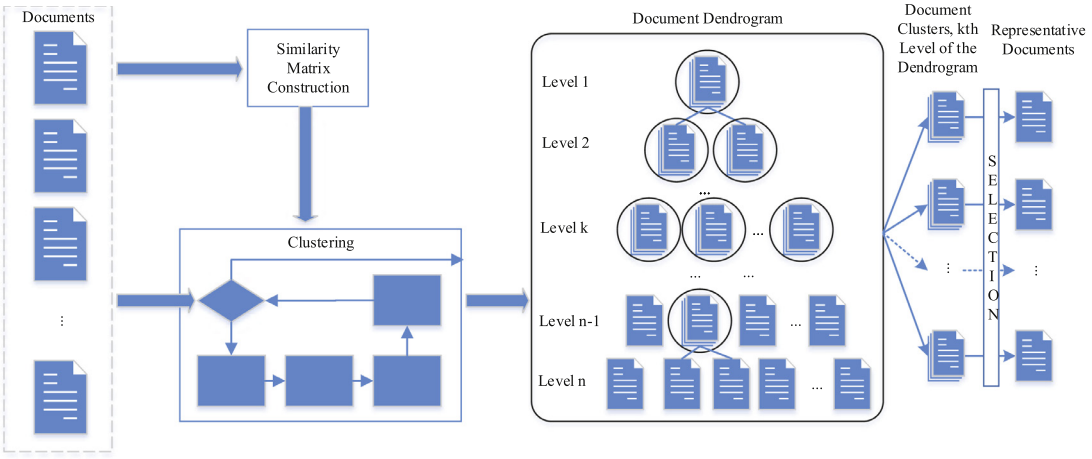

Fig. 3. Procedures of REPSET method.

(1) Similarity matrix construction

Given a set D with n documents, each document d is mapped as a keyword vector denoted as $[w_1, w_2, \cdots, w_p]$ with the TF/IDF model [33], then the cosine similarity between any two documents could be calculated, resulting in an $n \times n$ similarity matrix M. Specifically, the cosine similarity between document $d_1 = [w_{11}, w_{21}, \cdots, w_{p1}]$ and $d_2 = [w_{12}, w_{22}, \cdots, w_{p2}]$ in a p-dimensional vector space can be calculated as follows.

$$sim(d_1, d_2) = \frac{d_1 \bullet d_2}{\|d_1\| \times \|d_2\|} = \frac{w_{11} \times w_{12} + w_{21} \times w_{22} + \cdots + w_{p1} \times w_{p2}}{\sqrt{w_{11}^2 + w_{21}^2 + \cdots + w_{p1}^2} \times \sqrt{w_{12}^2 + w_{22}^2 + \cdots + w_{p2}^2}} \tag{20}$$

(2) Clustering

We designed a clustering algorithm which fulfills three principal requirements: (1) the accuracy of clustering results should be high; (2) the number of clusters should not be arbitrarily determined in advance, so that the system can respond immediately with different numbers of representative documents; and (3) there should be a multi-level structure in the clustering results, so that users can "drill down" on representative documents with different information granularity.

Concretely, each document d in the original set D is initially considered to be a single cluster, resulting in n clusters in the first stage, denoted as $C_1^1, C_2^1, \cdots, C_n^1$. Let C_j^k denote the jth cluster in the kth stage, then the average similarity of each pair of clusters C_m^k, C_n^k is:

$$sim(C_m^k, C_n^k) = \frac{1}{n_m^k n_n^k} \sum_{d_x \in C_m^k} \left(\sum_{d_y \in C_n^k} sim(d_x, d_y) \right) \tag{21}$$

where n_m^k and n_n^k represent the number of documents in clusters C_m^k and C_n^k, respectively. Based on the idea of hierarchical clustering, at stage k, two clusters with the maximum average similarity, C_p^k and C_q^k, will be merged to a new cluster, C_{pq}^k.

$$sim(C_p^k, C_q^k) = \max_{C_m^k, C_n^k \in \{C_1^k, C_2^k, \cdots, C_{n-k+1}^k\}} (sim(C_m^k, C_n^k)) \tag{22}$$

Given a threshold $\lambda (0 \le \lambda \le 1)$, the documents in the new cluster C_{pq}^k meeting the following condition will be marked as $d_{boundary}$:

$$\frac{1}{n_{pq}^k} \sum_{d_x \in C_{pq}^k} sim(d_{boundary}, d_x) < \lambda, \lambda \in [0, 1] \tag{23}$$

Since the value of λ may affect the accuracy of the clustering algorithm, we use the following measurement to tune the value of λ:

$$CS = \frac{\sum_{p=1}^{k} \left\{ \frac{1}{|C_p|} \sum_{d_i \in C_p} \{sim(d_i, O_p)\} \right\}}{\sum_{p=1}^{k} \left\{ \max_{q=1,2,\cdots,k, p \neq q} \{sim(O_p, O_q)\} \right\}} \tag{24}$$

In the equation, the numerator represents the aggregate in-cluster similarity after clustering where C_p denotes a cluster and O_p is the centroid document of C_p. The denominator represents the aggregate between-cluster similarity where O_p and O_q are the centroid documents of two clusters. Thus, the clustering result is better when the value of CS is higher, indicating a more appropriate value of λ. Moreover, since the expression of CS does not explicitly include λ, we use stepwise self-tuning to seek the optimal λ. Then the marked boundary documents will be re-allocated to some new cluster to achieve highest average similarity, which is called a backward strategy. Specifically, at stage k, there are $n-k+1$ clusters, denoted as $C_1^k, C_2^k, \cdots, C_{n-k+1}^k$. REPSET first calculates the average similarities between $d_{boundary}$ and the $n-k+1$ clusters, i.e., $sim(d_{boundary}, C_j^k) = \frac{1}{n_j^k}\sum_{d_x \in C_j^k} sim(d_{boundary}, d_x)$. If $sim(d_{boundary}, C_j^k)$ is the largest among all the clusters, then $d_{boundary}$ is re-allocated to the cluster C_j^k.

In each iteration of updating clusters, two clusters with the maximum similarity between each other are merged into one, and the documents near boundaries are also appropriately processed. Thus, after n iterations, the clusters will converge to 1 and the whole dendrogram will be well generated.

(3) Representative document selection

As stated in [34], the document with the highest average similarity to the other documents will offer the highest content coverage in the set. Thus, at each level of the dendrogram, REPSET extracts the corresponding representative documents in the clusters. Therefore, through one time running, REPSET can provide representative subsets at all levels of the dendrogram with high information content coverage. In addition, the low between-cluster similarity guarantees its low redundancy.

Overall, the proposed method REPSET is proved to be effective with respect to the measures *content coverage* and *redundancy*, as well as F_1 discussed in Section 2. This method has been applied and verified with huge data experiments and human evaluations in benchmarking comparison as well as in an organizational intra-blogging platform at a large Chinese mobile firm with around 80 million customers. In sum, the extraction method and system enables the managers to locate representative articles that are helpful for understanding the hot topics, prevailing thoughts, and emerging opinions among the employees.

3.3. Consistency-oriented extraction

In the context of text ranking such as review ranking in e-commerce, we have formulated the *consistent review ranking* (CRR) problem, aiming at providing consumers with concrete review ranking results that are *consistent* with the corresponding review summaries. Furthermore, a heuristic stepwise optimization approach has been proposed to maximize the expected consistency between the entire set of reviews and the ranking list of reviews which is a subset that a consumer would read.

Concretely, given a set of documents (i.e., reviews here) $D = \{d_1, d_2, \cdots, d_n\}$ and a ranking list of all these reviews $L = (d_{l_1}, d_{l_2}, \cdots, d_{l_n})$ where d_{l_i} denotes the ith review in the list, the consumers tend to read the reviews sequentially and may break at any position [35]. Suppose a consumer stops reading at position i, then the subset of reviews he/she has read consists of top i reviews, $S_i = \{d_{l_1}, d_{l_2}, \cdots, d_{l_i}\}$. The consistency of information read by the consumer is $Cons(S_i, D)$ which is described in Section 2.2.1. Given the probability distribution of reading breaking positions—$P = \{p_1, p_2, \cdots, p_n\}$, the expected consistency between D and the ranking list L is denoted as $expCons(L, D) = \sum_{i=1}^{n} p_i \cdot Cons(S_i, D)$.

The consistent review ranking (CRR) problem. Given an original set of reviews for a product $D = \{d_1, d_2, \cdots, d_n\}$, rank all these reviews to form a ranking list L such that the expected consistency between L and D, i.e., $expCons(L, D)$, is maximized. Let S_i denotes the subset of reviews in L that consumers stop reading at position i with probability p_i.

$$\begin{aligned} &\max\ expCons(L, D) = \sum_{i=1}^{n} p_i \sum_{f \in F_{S_i}} \frac{|D_f|}{|D|} \times \left(1 - \left|\frac{|S_{if}^+|}{|S_{if}|} - \frac{|D_f^+|}{|D_f|}\right|\right) \\ &s.t.\ S_i = \{d_{l_1}, d_{l_2}, \cdots, d_{l_i}\}, i = 1, 2, \cdots, n. \end{aligned} \tag{25}$$

Since the CRR problem is NP-hard, it is infeasible to find an exact solution and therefore approximation approaches are considered. Heuristically, the expected consistency, $expCons(L, D)$, is maximized at each step. Because $L_{i-1} = (d_{l_1}, d_{l_2}, \cdots, d_{l_{i-1}})$ is determined in previous steps, $\sum_{j=1}^{i-1} p_j \cdot Cons(S_j, D)$ is fixed. Thus, selecting a review, $d_{l_i} \in \{d | d \in D, d \notin S_{i-1}\}$, to maximize the expected consistency is just to maximize the consistency between D and $S_{i-1} \cup \{d_{l_i}\}$, i.e., $Cons(S_{i-1} \cup \{d_{l_i}\}, D)$.

First, we proposed an intuitive approach named stepwise optimization procedure (SOP) based on the above idea. At the beginning, an empty ranking list L_0 and its corresponding set $S_0 = \varnothing$ are initialized. Then one review is added to them in each iteration until all reviews have been added. In the ith iteration, all possible lists are generated based on L_{i-1} preserved at the previous iteration, denoted as SL_i. Later, a list $L_i \in SL_i$, with its corresponding S_i having the maximized $Cons(S_i, D)$, is preserved at the list for the next iteration. After n iterations, L_n is just the required ranking list L. The SOP method is greatly advantageous over the exact methods which need to enumerate all possible ranking lists in efficiency. However, SOP only considers one list in an expected consistency maximization manner that it might not be so effective. This motivates us to investigate the balance of effectiveness and efficiency. Further, we proposed an enhanced approach named enhanced SOP (eSOP) which preserves certain lists that performs well on $expCons(L, D)$ instead of only preserving the one with the maximum value in each iteration. The illustrative procedures of eSOP are shown in Fig. 4.

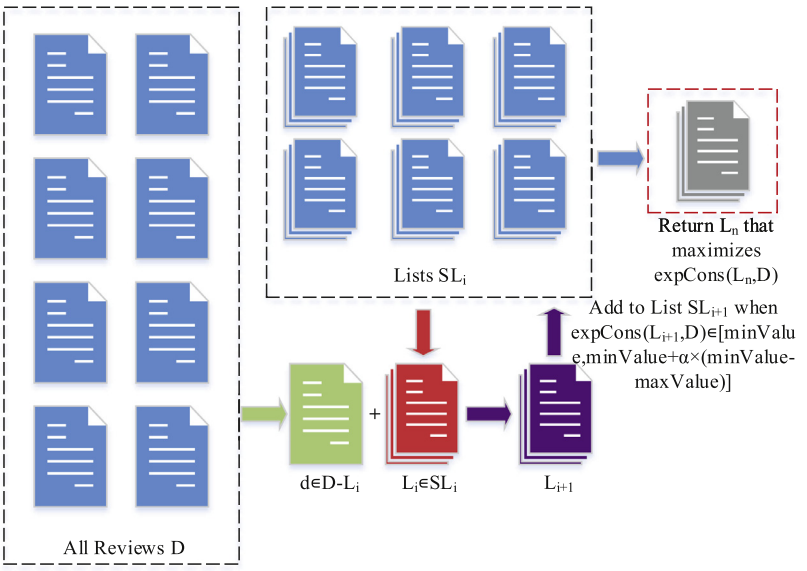

Fig. 4. Procedures of eSOP method.

The input for eSOP includes the review set $D = \{d_1, d_2, \cdots, d_n\}$ with all reviews structured as a set of feature-sentiment orientation tuples $d = \{(f, so) | f \in F, so \in SO\}$, the probability distribution $P = \{p_1, p_2, \cdots, p_n\}$ where p_i denotes the probability that a consumer stops reading at the ith review, and a threshold α to select the list generated in the same iteration with high expected consistency values. Same with SOP, an empty ranking list is initialized, $L_0 = ()$, with a ranking list set that only contains the empty list, $SL_0 = \{L_0\}$. In the ith iteration, all lists in the list set preserved from previous iteration, $L_{i-1} \in SL_{i-1}$, are selected to generate possible lists, $SL_i = \{L_i | L_{i-1} \cdot AddAtTheEnd(d), L_{i-1} \in SL_{i-1} \text{ and } d \in \{d | d \in D, d \notin S_{i-1}\}\}$. Later, the maximum and minimum expected consistency values of these list (i.e., $L_i \in SL_i$) are calculated, denoted as *maxValue* and *minValue*, respectively. Then the list with $expCons(L_i, D)$ falling in the interval $[minValue + \alpha \times (maxValue - minValue), maxValue]$ are preserved as the lists for the next iteration. After n iterations, the list in set SL_n with the maximum expected consistency value is finally chosen as the ranking list L.

The eSOP method has been verified with extensive real-world data experiments (such as the data from Amazon.com and Tmall.com) and human evaluations, showing its effectiveness and advantage over other related methods. Notably, the proposed method, eSOP, can be easily extended to more general contexts whenever considering the feature and sentiment information in the documents. By optimizing a ranking list of the original document set with respect to *consistency* measurement, we are able to obtain a consistent subset with any size $k (0 < k < n)$.

3.4. Other related work

In regard to the duplication measurements (i.e., *redundancy* and *compactness*), several related work has also been proposed. First, on the basis of Cov_{C+S}-*Select* which optimizes the information *coverage* measure, we also considered the influence of *redundancy* (i.e., $Red(S)$) on the extraction results. Generally, there are some trade-offs between information coverage and redundancy. That is to say, with the increase of extraction scale, the information coverage of subset is increasing, but along with the surging of redundancy. Thus, we treated it as a Multi-Objective Optimization problem to optimize *coverage* and *redundancy* simultaneously. Specifically, we designed a new objective function $Div(\lambda)$ where λ is the penalty parameter of information redundancy [36].

$$Div(\lambda) = Cov(S, D) - \lambda \cdot Red(S) \tag{26}$$

Analogously, we also used the idea of simulated annealing to design a heuristic iterative algorithm $CovRed_{C+S}$-*Select*. Particularly, the penalty factor λ is pre-determined and the initial solution D_0 also derives from the output of Cov_C-*Select*. In each iteration, the document with the highest information redundancy in the current solution is removed, and replaced with the one with the highest objective function value among the remaining documents. The replacement will be accepted if the objective function increases after the operation. Otherwise, an acceptance probability will be found by using the idea of simulated annealing. Theoretically, this algorithm approaches to the optimal solution with probability 1 when the iterations are large enough.

Second, a center-based method $Comp_{fuzzy} - Select$ has been proposed in our previous work for extracting the representative tuples in terms of *compactness*. Given a fuzzy relation $S = \{s_1, s_2, \cdots, s_k\}$ and threshold $\lambda = \{\lambda_1, \lambda_2, \cdots, \lambda_g\}$ with equivalence classes of $(M^+)_{\lambda i}, i = 1, 2, \cdots, k$, two tuples $t_i = \{\pi_{i1}, \pi_{i2}, \cdots, \pi_{ig}\}$ and $t_j = \{\pi_{j1}, \pi_{j2}, \cdots, \pi_{jg}\}$ are called λ-*close* if for

all $u \in \{1, 2, \cdots, g\}$, π_{iu} and π_{ju} are in the same equivalence class. Conceptually, all the tuples in a same equivalence class are regarded to express approximately the same information and therefore one of them could be extracted to represent the class. Suppose there are h λ-*close* tuples in class $C = \{s_1, s_2, \cdots, s_h\}$ of a relation S, $C_i = C - \{s_i\}, 1 \le i \le h$, and the relation compactness of C and C_i are SC and SC_i respectively. Then, tuple $t_p (1 \le p \le h)$ will be retained if and only if $SC_p = \max_{i=1}^{h} SC_i$. After repeating the extraction process for each class in the fuzzy relation, we can get the subset with minimum compactness.

4. Quality-aware review consistency ranking problem

This section presents our new effort in the small–big problem by incorporating a new perspective of quality. To be more specific, we focus on an extension to the consistency review ranking (CRR) problem (see Section 3.3), where reviews are deemed differently according to their quality with respect to features. In above-mentioned efforts [10,28,29], all reviews are treated equally when forming feature sentiment distributions. For instance, all positive opinions on service are considered in the same way without distinguishing them from each other such that "Service was good." and "The service was quite good, carefully answered my questions, especially the return policies. They also called me a week later to ask my experience of using the products" are regarded indifferent. Here, though both are positive reviews on the service feature, the latter seems more informative (e.g., more detailed and helpful, therefore of higher quality) than the former. Thus, it is desirable and meaningful to attach different weights to various quality degrees when forming the statistical feature sentiment distributions [37–39]. Apparently, taking such a quality-aware view into account conforms to representativeness and could help enrich the horizon of consistency.

4.1. Problem definition

In expressing the quality of the reviews with weights, a family of functions $\mathcal{H} = \{q : D \to \mathbb{R} | 0 \le q(d) \le 1\}$ could be introduced to map a specific review d to a real-numbered quality score $q(d)$ in the interval [0, 1]. The quality scores as weights could be either continuous or discrete, depending on the measurement of review quality. Concretely, if the reviews are classified into different quality groups [38,40], the quality scores can be set as a group-dependent discrete numbers; if the quality scores are derived through a continuous way, e.g., through model training and predicting using some machine learning methods [41,42], the quality scores could be set as continuous real numbers accordingly. With function q, such quality-aware weights can be assigned to reviews, so as to reflect their relative significances in consistency. In this way, the statistical feature sentiment could be derived as the weighted sum of reviews. For simplicity, let $w_{d_i f}$ denote the weight of review i on feature f, W_{D_f} denote the total weight of set R on feature f. Thus, $w_{d_i f}$ and W_{D_f} can be calculated by Eqs. (27) and (28). Likewise, the weight of positive-oriented review ($d_i^+{}_f$) and review set (D_f^+), denoted as $w_{d_i^+ f}$ and $W_{D_f^+}$ can be calculated as Eqs. (29) and (30).

$$w_{d_i f} = \begin{cases} q(d_i) & \text{if } d_i \text{ contains } f \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

$$W_{D_f} = \sum_i w_{d_i f} \tag{28}$$

$$w_{d_i^+ f} = \begin{cases} w_{d_i f} & \text{if } d_i \text{ contains positive opinion of } f \\ 0 & \text{otherwise} \end{cases} \tag{29}$$

$$W_{D_f^+} = \sum_i w_{d_i^+ f} \tag{30}$$

Thus the quality-aware sentiment orientation distribution deviation between review corpus D and subset S can be derived as $\left| \frac{W_{D_f^+}}{W_{D_f}} - \frac{W_{S_f^+}}{W_{S_f}} \right|$.

Apart from the sentiment orientation distribution divergence, the relative importance of feature f, i.e., the multiplier $\frac{|D_f|}{|D|}$ in Eq. (10), should also be extended when concerning the quality factor of reviews. Intuitively, if a feature is frequently mentioned by high-quality reviews, it would be very inappropriate to exclude it in the extracted subset, and it should be assigned a greater weight compared with other features of low frequency or mentioned in low-quality reviews. Here, review quality and feature frequency are two major factors of concern when deriving the relative significance of a feature. Then, the relative significance of feature f could be calculated as $\frac{W_{D_f}}{W_D}$, where W_{D_f} is calculated as Eq. (28), W_D is the quality sum of the all the reviews in D, i.e., $W_D = \sum_{d_i \in D} q(d_i)$, which serves as a normalizer to map the relative significance of feature f into a real number in [0, 1].

Formally, quality-aware review consistency metric can be defined as follows.

Definition 1. Quality-aware consistency metric. Suppose the original review set is D, and an extracted subset is S, i.e., $S \subseteq D$, the feature set commented on in D is denoted as $F = \{f_1, f_2, \cdots\}$ and the corresponding sentiment orientation set

Table 2
Example of reviews of different quality scores.

Review	Contents	Feature-SO tuples	Estimated quality score
1	Nice looking cover for the price, it fits very well.	(looking, +; fitting, +)	0.381
2	I keep my bike in the garage and needed a decent cover without spending a couple of hundred dollars. When I opened the package of this very inexpensive cover I was expecting the worst, but I was impressed. My bike is the full size 2014 Harley Ultra Limited "full dresser" with the new fairing, saddle bags and back box with luggage rack - just about as big as Harley's come. I was happy to see it fit very nicely and it looks nice on the bike. The only issue was my antennas on the back box would be too big for any cover. You have two choices, you can remove them when you cover your bike or you can cut two small holes in the cover to accommodate the antennas. I chose to cut the holes as I am using the cover for inside use only. Overall, it is a great cover for the money. I am ordering one in a smaller size for another motorcycle I have. If you decide to cut the holes, I would recommend you first remove the antennas, put you cover on perfectly, mark where you holes should be, remove your cover, cut small holes where your marks are and return the cover. In my case, since I replaced my whip antennas with smaller ones, I no longer have to remove them when installing the cover.	(looking, +; fitting, +)	0.885

is $SO = \{+, -\}$. Let W denote the quality-aware weight of review set, which is specified by Eqs. (27)–(30). The quality-aware consistency of review subset S in regard to D is defined as

$$QCons(D, S) = \sum_{f \in F_S} \frac{W_{D_f}}{W_D} \times \left(1 - \left|\frac{W_{D_f^+}}{W_{D_f}} - \frac{W_{S_f^+}}{W_{S_f}}\right|\right) \tag{31}$$

Guided by the quality-aware consistency metric, an extended representative information extraction problem with respect to online texts/reviews can be formulated. Given a large review set $D = \{d_1, d_2, \cdots, d_n\}$, our major task is to provide consumers with a subset S of high-quality reviews that could achieve high consistency with D. In order to further depict the pattern and uncertainty of consumers' reading behaviors(i.e., consumers tend to read the reviews sequentially and may stop at any position [35]), following [10] a distribution of possible reading stop positions could be incorporated, i.e., $P = \{p_1, p_2, \cdots, p_n\}$, where p_i indicates the probability that a consumer stops reading at review i, which could be captured by various ways, including eye tracking [35] and log file analysis [43] technologies. Then the quality-aware consistent review ranking problem could be formally defined as follows.

Definition 2. The quality-aware consistent review ranking (QCRR) problem. Given an original set of reviews for a product $D = \{d_1, d_2, \cdots, d_n\}$, rank all these reviews to form a ranking list L such that the expected quality-aware consistency between L and D, i.e., $expQCons(L, D)$, is maximized. Let $S_i = \{d_{l_1}, d_{l_2}, \cdots, d_{l_i}\}$ denote the subset of reviews in L that consumers stop reading at position i with probability p_i. The QCRR problem could be defined as

$$\begin{aligned} &\max\ expQCons(L, D) = \textstyle\sum_{i=1}^{n} p_i \sum_{f \in F_{S_i}} \frac{W_{D_f}}{W_D} \times \left(1 - \left|\frac{W_{D_{i\,f}^+}}{W_{D_f}} - \frac{W_{S_{i\,f}^+}}{W_{S_f}}\right|\right) \\ &s.t.\ S_i = \{d_{l_1}, d_{l_2}, \cdots, d_{l_i}\}, i = 1, 2, \cdots, n. \end{aligned} \tag{32}$$

It is worth mentioning that when the weights are all set to 1, the QCRR problem degenerates to the CRR problem, since W_{D_f} ($W_{D_f^+}$) becomes $|D_f|$ ($|D_f^+|$) when the quality of reviews on features is not distinguished (or treated equally). In other words, the CRR problem is a special case of the QCRR problem.

To better illustrate the different treatment for review quality, an example of two reviews for a motorcycle cover[1] from a real-world review set is shown in Table 2. It could be seen that both of them possess positive opinions in terms of looking and fitting. The first review puts up the opinions directly without any elaboration, whereas the second one provides detailed information about the fitting of the cover and even offers specific advice for people to better utilize the cover. Intuitively, the second review is more informative compared to the first one, as it is richer in review semantics, e.g., it contains the subjective feeling of the reviewer (happy), it has greater content richness, etc. Furthermore, the difference of the two reviews could be well reflected by their quality scores, which is measured through the process discussed in Section 4.2, i.e., the quality score of the second review is 0.885, which is far greater than that of the first one. Thus, although both reviews contain the same feature-SO tuples, their relative contribution to the intra-set consistency could be differentiated by their quality scores, which is intuitively appealing.

4.2. Review quality measurement

As typical text-form information, the quality of online reviews could be analyzed by referring text-based information quality analysis framework. According to [44], which surveyed twenty information quality research models, the informa-

[1] https://www.amazon.com/Classic-Accessories-73807-MotoGear-Motorcycle/dp/B000NNB1GG/.

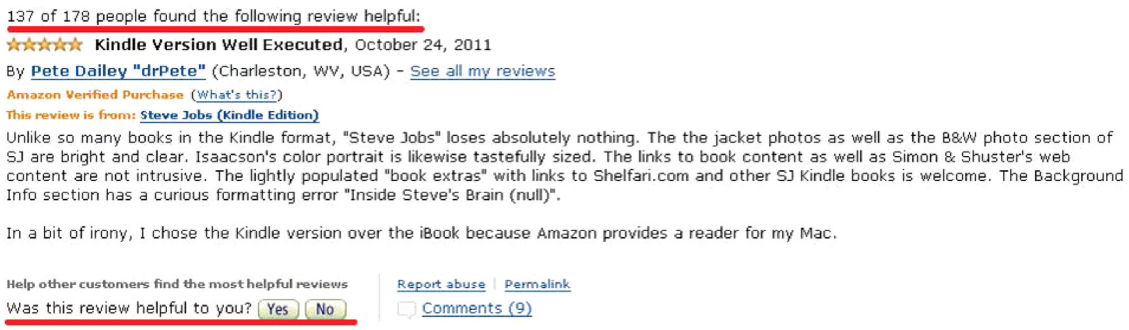

Fig. 5. Example of helpfulness voting mechanism on Amazon.

tion quality framework proposed by [45] not only strikes a balance between theoretical consistency and practicability but also is applicable to various domains. Concretely, the hierarchical text-based information quality framework proposed by [45] was constructed with four major dimensions, i.e., intrinsic data quality, contextual data quality, representational data quality as well as accessibility data quality, along with a few sub-dimensions. Under this framework, some recent efforts on online reviews [37,40] have identified the related facets and specific characteristics that impact the review quality. Moreover, some other related research analyzes online review quality from a text mining perspective, and incorporates finer-grained review characteristics into the quality framework [9,38,46,47].In addition, various characteristics of online reviews could be extracted and used for measuring the review quality. For instance, the review quality measurement has been dealt with human-annotated quality class labels [37,40], or as a prediction problem with the ground truth set as the human-voted quality score [42,48]. Here, the review quality was reflected from a human perception perspective. In practice, many e-commerce sites, e.g., amazon.com, tmall.com etc., have established mechanisms for consumers to evaluate the quality of reviews by encouraging them to vote on whether a review is helpful(see Fig. 5), which provides a natural benchmark for review quality measurement. Note that since not all reviews could get sufficient votes required to measure review quality, a predictive model could be trained by reviews with enough votes to further predict the quality of those reviews without sufficient votes.

In our work, we first construct a comprehensive quality framework by summarizing various facets and corresponding characteristics in literature (see Table 3), and use it to analyze the quality measurement through a procedure of model training and predicting. Take the running example in Table 2 for instance, in order to measure the quality of the two reviews, their characteristics are first analyzed in light of the quality framework shown in Table 3. Then, a prediction model should be trained by the reviews with quality scores already (sufficient quality votes in this case), and further adopted to predict the quality score of the two reviews according to their characteristics, which serve as the input of the model. In our work, an SVR model was trained and used for further prediction. For more details of the SVR model, please refer to Section 4.5.1. It could be well inferred that the second review possesses more characteristics compared to the first one and could probably be predicted with a higher quality score, which is verified by the predicted quality score shown in Table 2.

4.3. Complexity analysis

Before figuring out the possible solutions to the QCRR problem, the computational complexity of the problem needs to be examined first. Suppose that the quality of the reviews are pre-acquired, it can be proved that the QCRR problem is NP-hard since a classic NP-hard problem (namely, weighted maximum coverage problem (w-MC)), is reducible to QCRR. More specifically, it can be shown that any instance of w-MC can be transformed to a particular instance of QCRR in polynomial time and that the solution for the instance of QCRR can be transformed to the solution for the instance of w-MC in polynomial time.

Theorem 1. *The QCRR problem is NP-hard.*

Proof. An instance of w-MC consists of a set of elements $E = \{e_1, e_2, \ldots e_m\}$, the weight of each element is denoted as w_{e_j}, a collection of subsets $\{E_1, E_2, \ldots, E_n\}$ (where each subset contains some elements of E), and an integer k. The objective of solving the w-MC problem is to select no more than k subsets to maximize the total weight of these subsets. Given an instance of w-MC, we can transform it to a particular instance of QCRR. First, subsets E_i, elements e_j and element weights w_{e_j} can be represented as reviews r_i, features f_j and feature weights w_{f_j}, respectively. In this case, every subset E_i is represented by a review r_i . Then we can design an instance of QCRR on the review set R by setting $p_i = 1$ for $i = k$ and $p_i = 0$ for $i \neq k$. It is easy to see that this instance can be constructed in polynomial time.

In this particular instance of QCRR, we have $\frac{W_{D_i^+ f}}{W_{D_f}} = \frac{W_{S_i^+ f}}{W_{S_f}}$ for all feature f_j, because in the constructed instance all opinions are positively oriented. The objective is to maximize $p_k \times \sum_{f \in F_{S_k}} \frac{w_{f_j}}{W}$ (W is a normalizer irrelevant of f), which is the

Table 3
Online review quality framework.

Facet	Characteristic description	Number	Source
Stylistic	Ratio of uppercase and lowercase characters in review text	2	[46]
Stylistic	Ratio of 1-letter words in review text	1	[47]
Stylistic	Ratio of 2 to 9-letters words in review text, respectively	8	[47]
Stylistic	Ratio of 10 or more-letter words in review text	1	[47]
Stylistic	Review length measured by the number of sentences, words and characters in review text and title, respectively	6	[9,37,40,46,47]
Stylistic	Average length of words in review text and title measured by the number of characters, respectively	2	[47]
Stylistic	Average length of sentences in review text and title measured by the number of words and characters, respectively	4	[37,38,40,47]
Stylistic	Ratio of capitalized sentences in review text and title, respectively	2	[41]
Stylistic	Ratio of words that only used once in review text and title, respectively	2	[41]
Stylistic	Number of spelling errors	1	[9,37,40]
Stylistic	Ratio of nouns, adjectives, verbs, adverbs, punctuations ,symbols, numbers, comparative words and non-English words, respectively	9	[40,41]
Content richness	Number, ratio and times of product attributes mentioned	3	[38,40]
Content richness	Average tf-idf values	1	[37,40]
Content richness	Readability index of review text	1	[9,40]
Subjectivity	Number of positive and negative words	2	[41]
Subjectivity	Number of opinionated sentences	1	[38,40]
Subjectivity	Number of sentences containing positive, negative, and neutral opinions	3	[37,40]
Emotion	Number of words that express emotions, i.e., anger, disgust, fear, joy, sadness, and surprise, respectively	6	[49]
Product-related	number of reviews that the product has received	1	[49]
Product-related	average and standard deviation of review ratings for the product	2	[49]
Reviewer expertise	Number of words which represent various time tags, i.e., date, duration, set and time, respectively	4	[50]
Comparison with other reviews	Average cosine similarity between the tf-idf vectors of the review and its previous posted reviews	1	[37,40]
Comparison with other reviews	deviation between the rating and average rating	1	[37,40,47]
Comparison with other reviews	Kullback–Leibler Divergence between unigram model of the review and its previous posted reviews	1	[41]
Rating	Rating assigned to the product	1	[40,49]

same as maximizing the total weight of features in the top k reviews. Both instances have the same objective, which is to select k reviews (subsets) to maximize the total weights contained in them. Thus, there is a direct correspondence between the solution of the QCRR instance and the solution of the w-MC instance, which is obviously a simply transformation in polynomial time.

Therefore, we have proved that w-MC is reducible to QCRR, meaning that QCRR is NP-hard. □

4.4. Algorithm

As an NP-hard problem, QCRR cannot be solved directly in polynomial time, thus approximate methods should be exploited. A depth-first method named QCRR_{df} is proposed, with the corresponding algorithmic details as shown in Algorithm 1. First, a sequence denoted as C_{init} is initialized to store the maximum consistency currently found for each possible size of the ranking list. $f(R)$: $R \rightarrow L_{cur}$ is a function used to generate a fast approximate sequence, which is set as a greedy approach in our method. Each $c_i \in C_{init}$ is initialized by the subsequence of the first i elements in L_{cur} respectively. L_{best} is recorded accordingly when a better solution is found, and the initial value of L_{best} is set identical as L_{cur}, i.e., L_{cur} is treated as the current best result. The function *SEARCH* that receives the current ranking list l is the core step (lines 5–27) of the algorithm. To be more specific, the finale (lines 6–12) should be defined at first for a recursive algorithm, which saves the current best solution in L_{best} when there are no more candidates and returns. The candidates are generated into L_{new} and sorted by quality in descending order(lines 13–14). Next, every review L_{new} is added to the end of L, the maximum and minimum consistency scores $c_{\max}$, $c_{\min}$ are calculated with their corresponding reviews $d_{\max}$, $d_{\min}$ (lines 15–18). If $c_{\max}$ is greater than the current best consistency score found, the score will be updated correspondingly (lines 19–21). Finally, only the lists whose consistency score is greater than $c_{\max} - (1-\gamma)(\sum_{i=|L_{last}|}^{N} p_i)(c_{\max} - c_{\min})$ (lines 22–27) is searched. A parameter γ is introduced here to help controlling the cardinality of candidate lists. When $\gamma = 1$, only the local optimum is considered, therefore the algorithm degenerates to a greedy approach. When $\gamma = 0$, every candidate is considered, the algorithm is equivalent to the enumerating approach. The setting of γ will be further discussed in the experiment section. Moreover, the sum of stop reading probabilities for each remaining position ($\sum_{i=|L_{last}|}^{N} p_i$) is multiplied to $(1-\gamma)$ to further shrink the candidate lists.

Algorithm 1 QCRR_{df}.

Input: Review set D, parameter γ, stop reading probabilities $P = \{p_1, p_2, \ldots, p_{|R|}\}$, heuristic approach function $f(D) : D \rightarrow L_{cur}$ (Greedy in our implement)

Output: The tuple$(c_{|D|}, L_{best})$ containing the maximum consistency c_N and the corresponding ranking list L_{best}

$L_{cur} \leftarrow f(D)$, $L_{best} \leftarrow L_{cur}$
$C_{init} = \{expQCons(D, \{s_1, \ldots, s_j\}) | 1 \leq j \leq |D|, s_j \in L_{cur}\}$ // C_{init} is a sequence, let $C_{init} = \{c_1, c_2, \ldots, c_{|D|}\}$
SEARCH($\emptyset$)
return $c_{|D|}, L_{best}$
function SEARCH(l)
    $Depth \leftarrow |l|$
    if $Depth = |D|$ **then**
        if $expQCons(D, l) >= c_{|D|}$ **then**
            $L_{best} \leftarrow l$
            return
        end if
    end if
    $L_{new} \leftarrow D - l$
    Sort L_{new} by quality in descending order
    $d_{\max} \leftarrow \text{argmax}_{d \in L_{new}} expQCons(d, l \cup \{d\})$
    $c_{\max} \leftarrow expQCons(D, l \cup \{d_{\max}\})$
    $d_{\min} \leftarrow \text{argmin}_{d \in L_{new}} expQCons(D, l \cup \{d\})$
    $c_{\min} \leftarrow expQCons(D, l \cup \{d_{\min}\})$
    if $c_{|l|+1} < c_{\max}$ **then**
        $c_{|l|+1} \leftarrow c_{\max}$
    end if
    for $d \in L_{new}$ **do**
        if $expQCons(D, l \cup \{r\}) \geq c_{\max} - (1 - \gamma)(c_{\max} - c_{\min})(\sum_{i=|l|}^{|D|} p_i)$ **then**
            SEARCH($l \cup \{d\}$)
        end if
    end for
end function

4.5. Experiments

4.5.1. Experimental setup

In order to evaluate the performance of our proposed method QCRR_{df}, intensive experiments were conducted on real-world review data to compare it with other baseline methods in terms of both effectiveness and efficiency. The experimental environment was on a PC with 24G RAM and i5 4690K CPU. All the methods were implemented by Python 2.X version and run on CPython 2.7.6 64-bit interpreter.

The experiments were conducted on a real-world review dataset crawled from Amazon.com, which is frequently used as data source in related research [10,28], during the period of May–July 2014. The range of products included all the items that appeared in the main page of all directories shown in the "Full Store Directory"[2]. The quality voting information of the reviews was also recorded to measure the quality of the reviews. The dataset consisted of 2251 products with a total of 300,004 reviews. The reviews were pre-processed into structured form as sets of feature-sentiment orientation tuples to derive review summaries using mature techniques proposed in feature extraction and sentiment analysis fields [23,27]. Notably, as a pre-processing step, detailed discussion of these techniques would not be elaborated in this study. It can be calculated after pre-processing that the average number of features per product was 28.11, and the corresponding standard deviation was 18.36.

As mentioned in Section 4.2, the quality of the reviews needs to be measured before performing the review ranking algorithm. Following the framework in Table 3, the characteristics of the reviews were first extracted. Then, the quality score of each review with sufficient quality votes was calculated as the percentage of people who voted the review helpful. With all the scores serving as the training labels, together with the extracted review characteristics, an SVR model, which is a commonly adopted supervised machine learning model with associated learning algorithms that analyze data used for regression analysis, was trained to predict the quality of reviews without enough votes. In our experiments, the SVR model was trained using a training dataset containing 50,887 reviews which had received enough votes (more than 10 votes). The parameters of SVR were tuned with 5-fold cross validation and the best set of parameters made our regression model reached a mean square error at 0.0855, which was quite good. The corresponding best parameters were RBF kernel, $C =$

[2] https://www.amazon.com/gp/site-directory.

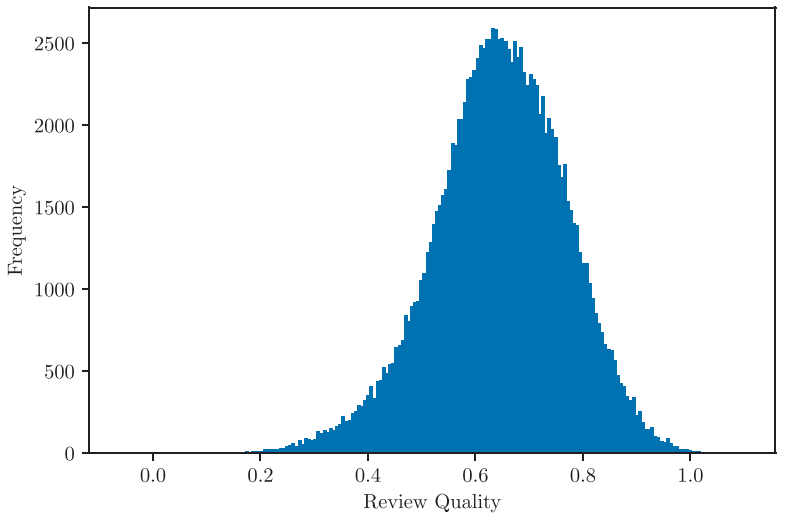

Fig. 6. Review quality distribution.

0.1162, $\gamma = 0.0082$ (γ here is a parameter in SVR model, which is not the same parameter as that in QCRR_{df}), respectively. The distribution of the predicted quality is shown in Fig. 6. With the predicted quality scores, various weights could be assigned to the reviews and thus the QCRR_{df} algorithm could be performed.

Moreover, the probability distribution of consumers breaking positions, i.e., distribution of p_i, is the necessary input for the QCRR_{df} algorithm and other methods that aimed at the proposed objective function. As discussed in [10], the distribution of p_i was set as uniform distribution, which is the same for all the related methods.

4.5.2. Tuning for parameter γ

To better understand the impact of parameter γ on the performance of QCRR_{df}, the experiments were conducted under different γ settings first. It can be seen from Section 4.4 that the smaller γ is set, the more results are considered in further consideration. Specifically, when $\gamma = 1$, the method is equivalent to an extreme greedy strategy, which is fast but less effective; when $\gamma = 0$, it is equivalent to brute-force searching, which yields a great performance for effectiveness but requires intensive computation. In other words, γ serves as an indicator to balance the effectiveness and efficiency of the method. To derive the best value of γ, γ was tuned from 0 to 1 stepped by 0.05 with a randomly chosen subset of the review data. The expected consistency score and the running time were recorded. The relative expected consistency scores *expQCons*(*D, L*) of different parameter settings, with the result of pure search ($\gamma = 0$) as the basis score, were shown in Fig. 7. It can be seen that equivalent performance could be achieved when γ was set smaller than 0.85. Furthermore, the ratio of effectiveness improvement over efficiency drops, i.e., $\frac{\Delta expQCons(D,L)}{\Delta t}$ was calculated for each γ setting, as shown in Fig. 8. From Fig. 8, it could be concluded that optimal balance between effectiveness and efficiency could be reached when γ was set as 0.85. Therefore, 0.85 was used for γ in the follow-up experiments.

4.5.3. Performance comparison

Notably, as an NP-hard optimization problem, QCRR could be solved by various classical heuristic methods, among which the most well-known are greedy, simulated annealing, stepping forward of approximation dynamic programming. Thus, these three were included as the representatives of classic heuristic methods in baselines. Moreover, some state-of-the-art methods were also incorporated as baseline methods, including two methods proposed in prominent work [28], i.e., greedy-based and integer regression-based method, as well as one method proposed in our previous research [10] for consistent review ranking, i.e., enhanced stepwise optimization procedure, denoted as eSOP. Apart from these methods, random and default ranking on the website were also considered as two baselines.

Concretely, the greedy method, denoted as Greedy, pursues maximum expected consistency at each iteration. Stepping forward approach of approximation dynamic programming, denoted as SF, is a greedy-based dynamic programming method that gradually selects the review that maximizes the future expected consistency, which is estimated with greedy strategy. It was applied to solve the proposed QCRR problem by making review selection decisions from the first review to the last one to maximize the expected consistency. In the ith iteration, given the current state S_{i-1}, the reward of decision d_{l_i} is estimated by considering both the current reward $p_i QCons(D, S_{i-1} \cup d_{l_i})$ and the future possible reward $\sum_{j=i+1}^{|D|} p_j QCons(D, S)$, where the future states are predicted in a greedy manner; i.e., $S_j = S_{j-1} \cup d_{l_i}, d_{l_i} = \max_{d \in \{d | d \in D, d \notin S_{j-1}\}} \{QCons(D, S_{j-1} \cup \{d\})\}, j = i+1, i+2, \cdots, n-1$. Simulated annealing, denoted as SA, is a well-known probabilistic metaheuristic method to provide sufficiently good approximations for optimization problems in large search spaces. The experiments adopted the algorithm

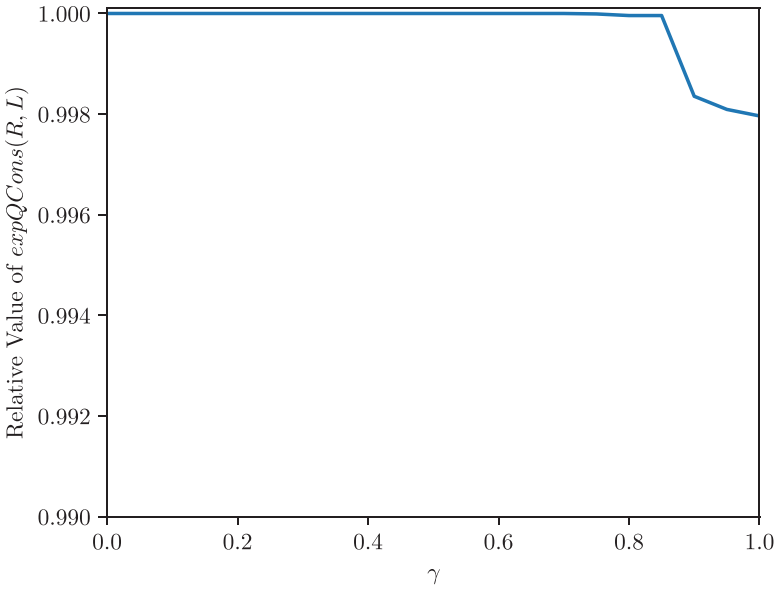

Fig. 7. Tuning parameter γ.

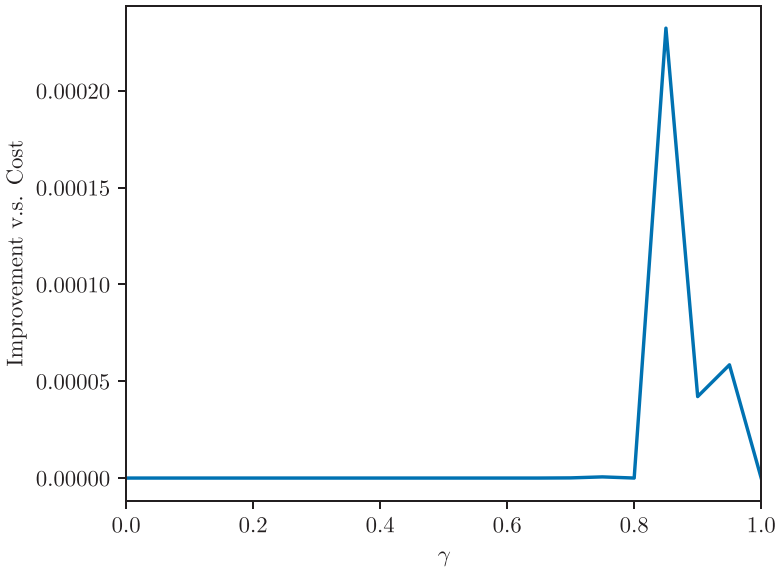

Fig. 8. Performance improvement versus efficiency cost.

proposed in [51] with its default parameters, where a new state was created by randomly permuting two reviews from the current state.

Moreover, the greedy-based method proposed in [28] when solving the characteristic review selection (CRS) problem, denoted as Greedy-CRS, works iteratively to minimize the inconsistency in each iteration until the size of the selected review set reaches the predefined number k, which is set as the number of the reviews to be ranked in our scenario. The integer-regression based method proposed in [28], denoted as IR-CRS first transforms the problem into a continuous regression one and obtains a nonnegative real valued solution and then generates the discrete solution that is closest to the continuous one, thus yielding a ranked result of the selected reviews, and the expected quality-aware consistency could be calculated as $\sum_{i=1}^{n} p_i QCons(D, S_i)$, where S_i represents the resultant set produced by IR-CRS given $k = i$. The eSOP method proposed in our previous effort [10] is a heuristic method that combines the strengths of greedy and exact permutation methods with a parameter α, ($\alpha \in [0, 1]$). eSOP works iteratively to select reviews to the ranking lists by maintaining a list set. In the ith iteration, all lists in the list set preserved from the previous iteration are selected to generate possible lists by appending a not yet selected review $d, d \in \{d | d \in D, d \notin S_{i-1}\}$ at the last of the list. The maximum and minimum expected consistency values of these lists are calculated and the lists whose expected consistency falls in the interval $[min + \alpha \times (max - min)]$

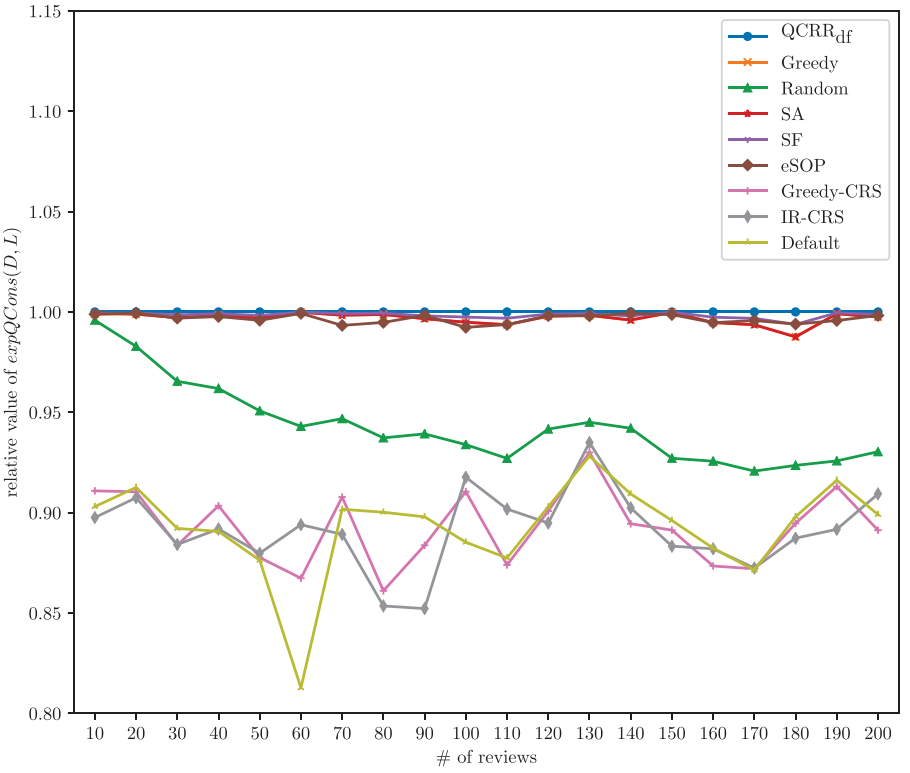

Fig. 9. Relative effects of different methods varying cardinality.

Table 4
Summary for effects of each method.

Method name	QCRR$_{df}$	SA	SF	Greedy	Random
expQCons(D,L)	2.0749	2.0703***	2.0724***	2.0700***	1.9959***
Standard deviation	0.6256	0.6251	0.6253	0.6248	0.6229
Method name	eSOP	Greedy-CRS	IR-CRS	Default	
expQCons(D,L)	2.0695***	1.8588***	1.8514***	1.8557***	
Standard deviation	0.6248	0.6055	0.6008	0.5854	

***: $p < 0.01$, **: $p < 0.05$, *: $p < 0.1$.

are preserved for next iteration. After n iterations, the list with the maximum expected consistency is finally chosen as the output ranking list. In our experiments, the parameter α was set identically as that in [10], i.e., $\alpha = 0.8$. The random method, denoted as Random, was designed to select the best result from 1000 random permutations of reviews. Additionally, the expected consistency of the default ranking on Amazon was also calculated as a baseline denoted as Default.

These methods were compared in terms of effectiveness and efficiency. The effectiveness was measured by the expected quality-aware consistency score, i.e., $expQCons(D, L)$. As the absolute values might vary greatly with the number of reviews, the relative performance of the baseline methods were calculated by treating the expected consistency score of QCRR$_{df}$ as the basis, i.e., for any baseline method i, the relative performance could be calculated as $\frac{expQCons(D,L)_{\texttt{method}_i}}{expQCons(D,L)_{\texttt{QCRR}_{df}}}$.

Thus, method i would outperform our proposed method if the ratio was greater than 1, otherwise our method outperformed method i. The relative performance of each method was shown in Fig. 9, with the x-axis representing review binning and the y-axis representing relative performance. It could be seen from Fig. 9 that all the ratios were smaller than 1, indicating that our proposed method outperformed all the baseline methods. Furthermore, a summary of the effects of different methods measured by the absolute expected quality-aware consistency score ($expQCons(D, L)$) is available in Table 4. Moreover, paired t-tests were conducted between our method and other baseline methods, with the significance levels reported in Table 4 with stars, from which it can be seen that our proposed method could achieve a significantly greater consistency score than other baseline methods with significance level $p < 0.01$. Notably, the default review order shown on the Amazon had the worst quality-aware consistency, which further emphasized the importance of our work.

The efficiency of each method was measured by running time. The results with varied review cardinalities were shown in Fig. 10. It can be seen that Greedy and Random were the two methods that could generate review ranking lists fastest,

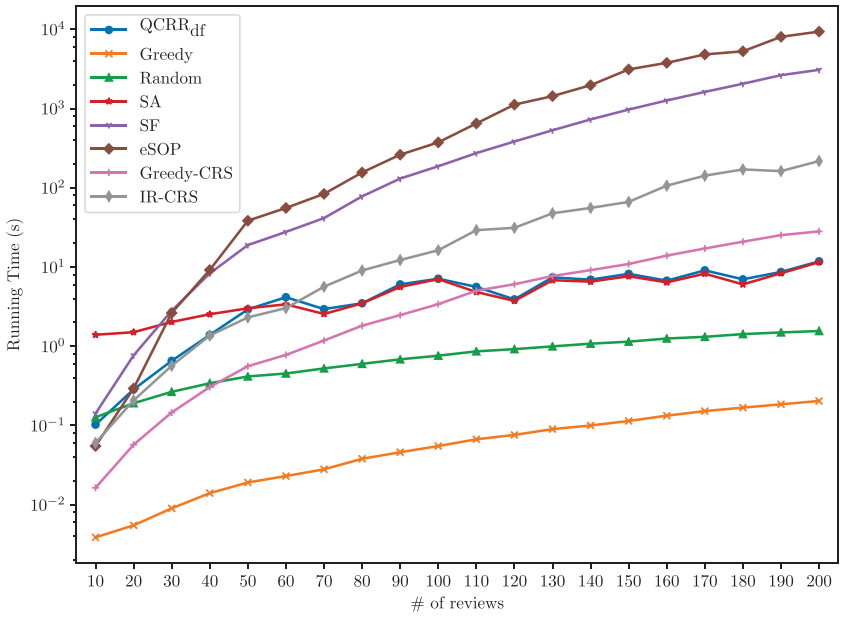

Fig. 10. Time efficiency of different methods varying cardinality.

Table 5
Summary for efficiency of each method.

Method name	QCRR$_{df}$	SA	SF	Greedy
Average running time (s)	2.4175	3.0799***	227.0430***	0.0315***
Standard deviation	2.8896	2.1824	617.2627	0.0473
Method name	Random	Greedy-CRS	IR-CRS	eSOP
Average running time (s)	0.4478***	2.6121**	17.7637**	649.8529***
Standard deviation	0.3677	6.0534	45.1284	1830.8308

***: $p < 0.01$, **: $p < 0.05$, *: $p < 0.1$.

which is in line with our estimation that Greedy is a heuristic method that performs with linear complexity. On the other hand, eSOP and SF were the two methods that worked with lowest efficiency, which fell in our expectation as the theoretical time complexity of SF could reach $O(n^4)$ in the worst case, and part of eSOP would degenerate to pure-enumeration with very high computational complexity. As for our method, the running time did not grow much as the number of review grew, showing a good scalability. Furthermore, the summarized results of efficiency were listed in Table 5 and the results of paired t-tests were reported by the stars along with the running time of each method, which showed that the average running time of QCRR$_{df}$ was significantly less than those of eSOP and SF at a significance level of $p < 0.01$. Therefore, it could be concluded that our proposed method outperformed other methods when considering both effectiveness and efficiency comprehensively. In other words, the experiments showed the superiority of our method in achieving great effectiveness without sacrificing the efficiency.

Finally, for illustrative purposes, let us revisit the review example shown in Table 2. As the CRR problem modeling holds an indifferent view toward the reviews once they include the same feature-sentiment orientation tuples, the two reviews in Table 2 would be treated equally when forming the consistent review ranking list. For instance, the eSOP method would come across a tie when dealing with the two reviews, and it would select one according to natural order in the original review corpus. In other words, the low-quality review may be selected if it happens to be placed in front due to the natural order in data storage for the original review corpus. Hence, the review subset read by consumers would include the low-quality review instead of the high-quality one, which is not desirable. As for the QCRR problem modeling that takes review quality into account, the second review would be given a greater weight (i.e., quality score) and thus extracted with priority compared to the first one.

5. Conclusion

This paper has addressed an important issue of extracting representative information (via query, search, and data analytics) for a small subset from a large original data corpus of big size, which is deemed particularly relevant and meaningful in the context of big data. Then, a number of related metrics including those formulated in our previous efforts have been

discussed from different perspectives of representativeness. Subsequently, several representative subset extraction methods with respect to various representativeness metrics have been elaborated. Furthermore, an extended effort has been made to formulate the quality-aware consistent review ranking (QCRR) problem, along with a framework that takes an array of facets and characteristics for describing review quality, as well as a heuristic method (namely, $QCRR_{df}$) for solving the problem. Extensive real-world data experiments have been conducted for justifying the effectiveness and efficiency of $QCRR_{df}$ compared with the baselines. Future work may center on extending $QCRR_{df}$ to other domains of applications where texts pertain, so as to provide users with representative subsets of original texts in support of their decision processes. Another ongoing exploration is to reflect the extents to which the sentimental orientations are modeled with degrees, so as to represent positive, negative as well as neutral opinions in QCRRdf with multi-level linguistic terms and hedges (e.g., strong and weak positive/negative/neutral), which are essentially fuzzy extensions.

Acknowledgments

Hereby, the authors highly appreciate the support from and friendship with Janusz Kacprzyk over many years, and would like to join with other colleagues worldwide in respectfully acknowledging his great contributions to the fields in both theoretical and applied ways.

References

[1] E. Brynjolfsson, Y. Hu, M.D. Smith, Consumer surplus in the digital economy: estimating the value of increased product variety at online booksellers, Manage. Sci. 49 (11) (2003) 1580–1596.
[2] F. Branco, M. Sun, J.M. Villas-Boas, Too much information? Information provision and search costs, Market. Sci. 35 (4) (2015) 605–618.
[3] B. Liu, Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data, Springer Science & Business Media, 2007.
[4] A. Spink, D. Wolfram, M.B. Jansen, T. Saracevic, Searching the web: the public and their queries, J. Assoc. Inf. Sci. Technol. 52 (3) (2001) 226–234.
[5] Y.C. Chen, R.A. Shang, C.Y. Kao, The effects of information overload on consumers subjective state towards buying decision in the internet shopping environment, Electron. Commer. Res. Appl. 8 (1) (2009) 48–58.
[6] B. Ma, Q. Wei, Measuring the coverage and redundancy of information search services on e-commerce platforms, Electron. Commer. Res. Appl. 11 (6) (2012) 560–569.
[7] P. Boldi, M. Santini, S. Vigna, Pagerank: functional dependencies, ACM Trans. Inf. Syst. (TOIS) 27 (4) (2009) 19.
[8] S. Fox, K. Karnawat, M. Mydland, S. Dumais, T. White, Evaluating implicit measures to improve web search, ACM Trans. Inf. Syst. (TOIS) 23 (2) (2005) 147–168.
[9] A. Ghose, P.G. Ipeirotis, Estimating the helpfulness and economic impact of product reviews: mining text and reviewer characteristics, IEEE Trans. Knowl. Data Eng. 23 (10) (2011) 1498–1512.
[10] Z. Zhang, G. Chen, J. Zhang, X. Guo, Q. Wei, Providing consistent opinions from online reviews: a heuristic stepwise optimization approach, INFORMS J. Comput. 28 (2) (2016) 236–250.
[11] B. Ma, Q. Wei, G. Chen, J. Zhang, X. Guo, Content & structure coverage: extracting a diverse information subset, INFORMS J. Comput. 29 (4) (2017) 660–675.
[12] X. Guo, G. Chen, Q. Wei, J. Zhang, D. Qiao, Extracting representative information on intra-organizational blogging platforms, MIS Quart. (2017) http://www.misq.org/skin/frontend/default/misq/pdf/Abstracts/13229_RA_GuoWeiAbstract.pdf.
[13] J. Zhang, G. Chen, X. Tang, Extracting representative information to enhance flexible data queries, IEEE Trans. Neural Netw. Learn. Syst. 23 (6) (2012) 928–941.
[14] Y. Bernstein, J. Zobel, Redundant documents and search effectiveness, in: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, ACM, 2005, pp. 736–743.
[15] F. Pan, W. Wang, A.K. Tung, J. Yang, Finding representative set from massive data, in: Fifth IEEE International Conference on Data Mining, IEEE, 2005, p. 8pp.
[16] C.X. Zhai, W.W. Cohen, J. Lafferty, Beyond independent relevance: methods and evaluation metrics for subtopic retrieval, in: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2003, pp. 10–17.
[17] J. Zhuang, S.C. Hoi, A. Sun, On profiling blogs with representative entries, in: Proceedings of the Second Workshop on Analytics for Noisy Unstructured Text Data, ACM, 2008, pp. 55–62.
[18] J. Carbonell, J. Goldstein, The use of MMR, diversity-based reranking for reordering documents and producing summaries, in: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 1998, pp. 335–336.
[19] C.E. Shannon, A mathematical theory of communication, ACM SIGMOBILE Mobile Comput. Commun. Rev. 5 (1) (2001) 3–55.
[20] D. Dubois, H. Prade, Fuzzy cardinality and the modeling of imprecise quantification, Fuzzy Sets Syst. 16 (3) (1985) 199–230.
[21] F. Herrera, L. Martínez, A 2-tuple fuzzy linguistic representation model for computing with words, IEEE Trans. Fuzzy Syst. 8 (6) (2000) 746–752.
[22] F. Li, M. Huang, X. Zhu, Sentiment analysis with global topics and local dependency., in: AAAI, 10, 2010, pp. 1371–1376.
[23] Q. Miao, Q. Li, D. Zeng, Fine-grained opinion mining by integrating multiple review sources, J. Assoc. Inf. Sci. Technol. 61 (11) (2010) 2288–2299.
[24] Q. Su, X. Xu, H. Guo, Z. Guo, X. Wu, X. Zhang, B. Swen, Z. Su, Hidden sentiment association in Chinese web opinion mining, in: Proceedings of the 17th International Conference on World Wide Web, ACM, 2008, pp. 959–968.
[25] J. Yi, W. Niblack, Sentiment mining in webfountain, in: 21st International Conference on Data Engineering, 2005. ICDE 2005. Proceedings, IEEE, 2005, pp. 1073–1083.
[26] N. Archak, A. Ghose, P.G. Ipeirotis, Deriving the pricing power of product features by mining consumer reviews, Manage. Sci. 57 (8) (2011) 1485–1509.
[27] M. Hu, B. Liu, Mining and summarizing customer reviews, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2004, pp. 168–177.
[28] T. Lappas, M. Crovella, E. Terzi, Selecting a characteristic set of reviews, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2012, pp. 832–840.
[29] P. Tsaparas, A. Ntoulas, E. Terzi, Selecting a comprehensive set of reviews, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2011, pp. 168–176.
[30] L. Zhang, B. Liu, S.H. Lim, E. O'Brien-Strain, Extracting and ranking product features in opinion documents, in: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, Association for Computational Linguistics, 2010, pp. 1462–1470.
[31] D.H. Kraft, A. Bookstein, Evaluation of information retrieval systems: a decision theory approach, J. Assoc. Inf. Sci. Technol. 29 (1) (1978) 31–40.
[32] G.L. Nemhauser, L.A. Wolsey, M.L. Fisher, An analysis of approximations for maximizing submodular set functions—I, Math. Program. 14 (1) (1978) 265–294.
[33] G. Salton, The Smart Retrieval System Experiments in Automatic Document Processing, 1971.
[34] X. Liu, W.B. Croft, Evaluating text representations for retrieval of the best group of documents, in: European Conference on Information Retrieval, Springer, 2008, pp. 454–462.

[35] E. Cutrell, Z. Guan, What are you looking for?: an eye-tracking study of information usage in web search, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2007, pp. 407–416.
[36] B. Ma, Q. Wei, G. Chen, Extracting a diverse information subset by considering coverage and redundancy simultaneously, Working Paper (2017).
[37] C.C. Chen, Y.D. Tseng, Quality evaluation of product reviews using an information quality framework, Decis. Support Syst. 50 (4) (2011) 755–768.
[38] J. Liu, Y. Cao, C.-Y. Lin, Y. Huang, M. Zhou, Low-quality product review detection in opinion summarization., in: EMNLP-CoNLL, 7, 2007, pp. 334–342.
[39] N. Tian, Y. Xu, Y. Li, G. Pasi, Quality-aware review selection based on product feature taxonomy, in: Asia Information Retrieval Symposium, Springer, 2015, pp. 68–80.
[40] X. Zheng, S. Zhu, Z. Lin, Capturing the essence of word-of-mouth for social commerce: assessing the quality of online e-commerce reviews by a semi-supervised approach, Decis. Support Syst. 56 (2013) 211–222.
[41] Y. Lu, P. Tsaparas, A. Ntoulas, L. Polanyi, Exploiting social context for review quality prediction, in: Proceedings of the 19th International Conference on World Wide Web, ACM, 2010, pp. 691–700.
[42] X. Yu, Y. Liu, X. Huang, A. An, Mining online reviews for predicting sales performance: a case study in the movie domain, IEEE Trans. Knowl. Data Eng. 24 (4) (2012) 720–734.
[43] M. Kamvar, M. Kellar, R. Patel, Y. Xu, Computers and iPhones and mobile phones, oh my!: a logs-based comparison of search users on different devices, in: Proceedings of the 18th International Conference on World Wide Web, ACM, 2009, pp. 801–810.
[44] M.J. Eppler, D. Wittig, Conceptualizing information quality: a review of information quality frameworks from the last ten years, in: IQ, 2000, pp. 83–96.
[45] R.Y. Wang, D.M. Strong, Beyond accuracy: what data quality means to data consumers, J. Manage. Inf. Syst. 12 (4) (1996) 5–33.
[46] M.P. OMahony, B. Smyth, A classification-based review recommender, Knowl. Based Syst. 23 (4) (2010) 323–329.
[47] Q. Cao, W. Duan, Q. Gan, Exploring determinants of voting for the helpfulness¥ of online user reviews: a text mining approach, Decis. Support Syst. 50 (2) (2011) 511–521.
[48] Y. Liu, X. Huang, A. An, X. Yu, Modeling and predicting the helpfulness of online reviews, in: Eighth IEEE International Conference on Data Mining, 2008. ICDM'08, IEEE, 2008, pp. 443–452.
[49] D. Yin, S.D. Bond, H. Zhang, Anxious or angry? Effects of discrete emotions on the perceived helpfulness of online reviews, MIS Quart. 38 (2) (2014) 539–560.
[50] H. Min, J.C. Park, Identifying helpful reviews based on customers mentions about experiences, Expert Syst. Appl. 39 (15) (2012) 11830–11838.
[51] J. Vandekerckhove, General simulated annealing algorithm, MATLAB Central File Exchange, 2006.