

The Pareto Principle is Everywhere: Finding Informative Sentences for Opinion Summarization through Leader Detection

Linhong Zhu¹, Sheng Gao², Sinno Jialin Pan², Haizhou Li², Dingxiong Deng³, and Cyrus Shahabi³

Abstract Most previous works on opinion summarization focus on summarizing sentiment polarity distribution towards different aspects of an entity (e.g., battery life and screen of a mobile phone). However, users' demand may be more beyond this kind of opinion summarization. Besides such coarse-grained summarization on aspects, one may prefer to read detailed but concise text of the opinion data for more information. In this paper, we propose a new framework for opinion summarization. Our goal is to assist users to get helpful opinion suggestions from reviews by only reading a short summary with a few informative sentences, where the quality of summary is evaluated in terms of both aspect coverage and viewpoints preservation. More specifically, we formulate the informative-sentence selection problem in opinion summarization as a community-leader detection problem, where a *community* consists of a cluster of sentences towards the same aspect of an entity and *leaders* can be considered as the most informative sentences of the corresponding aspect. We develop two effective algorithms to identify communities and leaders. Reviews of six products from Amazon.com are used to verify the effectiveness of our method for opinion summarization.

1 Introduction

Nowadays, opinion data can be widely found on the Web, such as product reviews, personal blogs, forums and news groups. Such information is highly valuable to e-commerce users (e.g., manufacturers, customers, online adver-

Information Sciences Institute, USA¹, Institute for Infocomm Research, Singapore²,
University of Southern California, USA³
linhong@isi.edu¹, {jspan, gaosheng, hli}@i2r.a-star.edu.sg², {dingxiong.deng,
shahabi}@usc.edu³

tisers, and etc.). For example, travelers may rely on comments about hotels on Tripadvisor¹ to book an appropriate resort.

However, the flourish of online opinions is a double-edged sword, which provides useful information meanwhile poses challenges in digesting all the massive information. For instance, in Amazon, some popular products may get hundreds even thousands of reviews, which makes it difficult for potential customers to go through all the reviews to make an informed decision on purchase. Furthermore, some reviews are noninformative and may even mislead customers. To address these issues, most online portals provide two services: aspect summary and review helpfulness rating. Accordingly, various amount of research has been conducted on aspect-based opinion summarization [16, 28, 29, 34, 43, 47] and review quality evaluation [7, 20, 23, 26].

Aspect-based opinion summarization aims to identify aspects of a given entity, and summarize the overall sentiment orientation towards each aspect. For example, for a mobile phone product, aspect-based opinion summarization may return the following information “battery life (three stars); screen (five stars); sound quality (five stars)”, where *battery life*, *screen* and *sound quality* are three of the aspects of a mobile phone, and the numbers of stars denote the corresponding overall sentiment orientation towards the aspects summarized from existing reviews. This kind of summarization is useful for consumers. However, it may lose some detailed information, which is also important for consumers to make decisions. For example, travelers may prefer to get information on suggested traveling routines in detail instead of only summarizing which tourist spots are good or bad.

In some scenarios, opinion summarization by selecting informative reviews is more desirable. Some approaches have been proposed to this task. A common idea behind them is to predict a score for each review to estimate its helpfulness, and select the top ones as informative reviews. However, most of them do not take the following two issues into consideration: 1) **redundancy**, the reviews with highest scores on helpfulness may contain redundant information; 2) **coverage**, the reviews with highest scores on helpfulness may not cover all aspects of the entity, and some important aspects may be missing.

In our prior work [46], we have proposed a new opinion summarization framework, named *sentence-based opinion summarization*, to address these issues. Given a set of reviews for a specific entity, the goal of sentence-based opinion summarization is to extract a small subset of informative sentences to represent the reviews, under the assumption that important sentences are origins of topics and opinions. Importance analysis is widely studied in various areas such as business-management, social network analysis and so on. In the early 1900s, economists have observed the Pareto principle [5]: where something is shared among a sufficiently large set of participants, there must be a number k between 50 and 100 such that “ $k\%$ is taken by $(100-k)\%$ of the participants”. In the same way, in this work, given a piece of opinion

¹ <http://www.tripadvisor.com/>

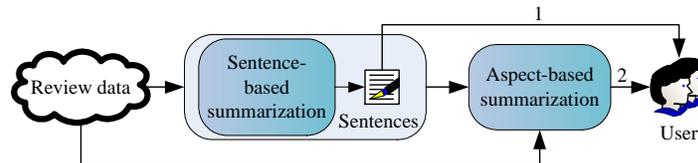


Fig. 1 Sentence-based summarization versus aspect-based summarization.

text, we focus on extracting a small number of sentences that cover the great mass of opinions and topics and generating a summary for it. The quality of summary is evaluated in terms of the coverage of the entity aspects and the polarity distribution preservation of the aspects (i.e., positive, negative, or neutral). In other words, we aim to generate summaries by extracting a small number of sentences from the reviews of a specific entity, such that the coverage of the entity aspects and the polarity distribution of the aspects can be preserved as much as possible. Note that the proposed framework is not to resume aspect-based opinion summarization approaches. In contrast, since the selected informative sentences preserve the coverage and sentiment polarity distribution of the entity aspects, aspect-based opinion summarization techniques can be post-applied to the selected sentences to generate summarization towards each aspect without information loss. Figure 1 depicts the relationship between our sentence-based opinion summarization and aspect-based opinion summarization.

Based on our opinion summarization framework, we propose a graph-based method to identify informative sentences. More specifically, we formulate the informative sentence selection problem in opinion summarization as a community and leader detection problem in social computing. A sentence graph is constructed by adding an edge between a pair of sentences if they are similar in both word distribution and sentiment polarity distribution. Subsequently, each node of the graph representing a sentence can be considered as a user in social computing. Thus, in the sentence graph, a community consists of a set of sentences towards the same aspect and the leaders of the community can be considered as the most informative sentences.

Finally, we propose two algorithms to detect leaders on the sentence-graph. We first propose a *Clique-based Community and Leader detection algorithm* (CCL), where we find overlapping communities by enumerating all maximal cliques and then model the community leader detection problem as a budgeted maximum coverage problem. The CCL algorithm is able to well preserve both aspect coverage and polarity distribution. However, there are some limitations of CCL in terms of efficiency (enumerating all maximal cliques is very time-consuming) and parameter-free issue (the size of summary is highly depended on the parameter setting). To this end, we further develop an alternative algorithm, which aims to *Simultaneously detect Communities and Leaders on the sentence-graph* (SCL), where communities are formed by

assigning other sentences to leaders (i.e., informative sentences), and leaders are selected according to their informativeness in both documents and communities. Though SCL obtains lower aspect coverage than CCL, it is a good trade-off between efficiency and effectiveness. In addition, our user study show that SCL is preferred by real users in terms of conciseness. Therefore, if aspect-based opinion summarization is required to be post-applied to the selected sentences (Case 2 in Figure 1), CCL would be a better choice with less information loss. When a summary which is generated from selected sentences is directly displayed to users (Case 1 in Figure 1), we suggest using the SCL algorithm with conciseness.

In all, we summarize our contributions of this research:

- We have introduced a new sentence-based summarization framework which generates summaries that preserve aspect coverage as much as possible and are representatives of aspect-level viewpoints.
- We have bridged across the area of sentiment analysis and the area of social computing by applying community and leader detection algorithm to solve the informative sentences selection problem.
- We have presented two effective leader-community detection algorithms, namely, clique-based community and leader detection algorithm (CCL) and simultaneous community and leader detection algorithm (SCL), to find informative sentences from a sentence graph.
- We have conducted experiments using real data collected from Amazon product review and two evaluation metrics “aspect coverage” and “polarity distribution preservation”. Our experimental results demonstrated the effectiveness of the proposed technique.

2 Related Work

The most related work to ours is sentiment summarization [3, 11, 21] where a summary is built by extracting representative bits of text from a set of documents. Lerman et al. [21] aim to generate summaries that are representative of the average opinion and cover important aspects when aspect set is given. The quality of summary is evaluated in terms of the mismatch between the average sentiment of a summary and the known sentiment of an entity and the coverage of aspects. The goal of our work is more fine-grained: to generate summaries that maximize aspect coverage and preserve the aspect-level viewpoints of an entity without knowing aspect set in advance. Another work which is known as Opinosis [11], aims to generate concise abstractive summaries of highly redundant opinion data for a specific aspect (i.e., battery life for kindle). The key idea of Opinosis is to use a word graph to represent the opinion data, and then repeatedly find paths through the graph to produce concise summaries. However, our work differs in both the problem setting and methodology from Opinosis. In our work, aspects are unknown and sen-

tences are not grouped via aspects in advance, while Opinosis takes groups of sentences towards different aspects as inputs. In addition, our method uses sentence graph and detects leaders of sentence community to generate concise summaries instead of using word graph and finding paths to generate summaries.

Besides sentences/words selection, aspect-based approaches are another important branch of sentiment summarization, which includes three distinct steps: aspect/feature identification [16], sentiment prediction/classification [12, 36, 44, 45], and summary generation [28]. According to a survey on opinion summarization [19], most of the existing works use three kinds of approaches to perform aspect identification: mining techniques [16, 15], Natural Language Processing (NLP)-based techniques [34], and integrated techniques [4, 14, 28, 29, 39, 47]. In this work, we propose a new sentence-based summarization framework, whose objective is totally different from those aspect-based summarization approaches.

Review quality prediction is another branch of related works [7, 20, 22, 23, 26], which aims to estimate scores for each review, and rank the reviews based on the scores. Recently, Tsaparas et al. [40] propose to select a comprehensive subset of reviews to represent the original reviews. In their work, the review-selection problem is modeled as a maximum coverage problem and several heuristic algorithms are proposed to greedily select a set of comprehensive reviews. Our work is different from theirs in two ways: 1) our opinion data selection is done in the sentence level rather than the review level, and 2) we model the summarization problem as a community leader detection problem in sentence graph.

Our work is also related (but not highly relevant) to existing works on multi-document summarization via sentence selection [8, 17, 25, 41, 30, 31], subjective summarization [33], and sentence compression for single-document summarization [9, 42]. In document summarization, the objective is to summarize the information content in the document with shorter texts, while the opinion summarization task focused on features or objects on which customers have opinions. In addition, our methodology differs from previous graph-based ranking methods such as textrank [30, 31] and clustering-based techniques such as [41] on multi-document summarization. Compared with textrank, our work generate summaries using both the sentence-sentence term similarity and the sentiment polarity information; while in textrank, either the sentiment polarity information of the sentences or the intersection between sentences is not taken into consideration. Though community detection is essentially a clustering problem, we highlight that our method differs from previous clustering-based techniques in the following aspects: 1) Our goal is to select informative sentences rather than group similar sentences together, thus our main focus is to detect leaders. 2) The extracted leaders are different from the centroids of clusters. A centroid represents a statistical high relevance to a cluster of sentences, but may suffer from the low informative and manipulated issues. For instance, a cluster of sentences may consist

1. D_1 Definitely worth the price!	<table border="1"> <thead> <tr> <th>sentence</th> <th>aspect-level sentiment</th> <th>review-level sentiment</th> </tr> </thead> <tbody> <tr> <td>s_1</td> <td>price +</td> <td>D_1 +</td> </tr> <tr> <td>s_2</td> <td>price +</td> <td>D_2 +</td> </tr> <tr> <td>s_3</td> <td>bubble -</td> <td>D_1 +</td> </tr> <tr> <td>s_4</td> <td>bubble +</td> <td>D_2 +</td> </tr> <tr> <td>s_5</td> <td>price -, bubble -</td> <td>D_3 -</td> </tr> <tr> <td>s_6</td> <td>bubble -</td> <td>D_4 -</td> </tr> </tbody> </table>	sentence	aspect-level sentiment	review-level sentiment	s_1	price +	D_1 +	s_2	price +	D_2 +	s_3	bubble -	D_1 +	s_4	bubble +	D_2 +	s_5	price -, bubble -	D_3 -	s_6	bubble -	D_4 -
sentence		aspect-level sentiment	review-level sentiment																			
s_1		price +	D_1 +																			
s_2		price +	D_2 +																			
s_3		bubble -	D_1 +																			
s_4		bubble +	D_2 +																			
s_5	price -, bubble -	D_3 -																				
s_6	bubble -	D_4 -																				
2. D_2 For the price, I would recommend it.																						
3. D_1 They get bubbles, and a pain to apply.																						
4. D_2 Completely bubble-free, good protector!																						
5. D_3 Lured by the low price tag for these iPad protectors, I tried to apply them twice and finally gave up																						
6. D_4 No matter what I did I could never get the air bubbles out and ended up wasting 2 out of 3 covers.																						

Fig. 2 An illustrative example.

of all spam reviews resulting in the centroid sentence being of low quality. Instead, our leader detection algorithm makes use of informativeness within a community and within a review to select high quality sentences.

3 Problem Formulation

Denote x a specific entity that consists of a set of aspects $\mathcal{A} = \{a_1, a_2, \dots, a_m\}$, and a set of reviews on the entity $\mathcal{R} = \{D_1, D_2, \dots, D_l\}$, where D_i ($i = 1$ to l) represents a review. Each review D_i consists of several sentences $D_i = \{s_1, s_2, \dots, s_{n_i}\}$, where s_j ($j = 1$ to n_i) represents a sentence. Define $|D_i| = n_i$ the size of the review D_i , and $|\mathcal{R}| = \sum_{i=1}^l |D_i|$ the size of the review set \mathcal{R} .

Based on the above terminologies, the informative sentence selection problem is defined as follows,

Problem 1. (*Sentence-based opinion summarization*) Given a set of reviews \mathcal{R} on a specific entity x , which consists of a set of aspects \mathcal{A} , our goal is to find a few number of sentences \mathcal{S} where $|\mathcal{S}| \ll |\mathcal{R}|$ such that \mathcal{S} covers the aspects in \mathcal{A} as many as possible and preserves the **aspect-level** sentiment polarity distribution of \mathcal{R} as much as possible. Note that both the aspect set \mathcal{A} and their sentiments are unknown in training.

The goal of Problem 1 is to generate a summary of documents that is representative of the average aspect-level sentiment. We provide more perspectives of Problem 1 using the following example.

Example 1. Figure 2 shows an example of six sentences from four reviews discussing about an entity iPad protector. Though both aspects and sentiments are unknown, here we also list them in the right Table to help illustrate what a good solution is to Problem 1. From the example, the overall sentiments of review D_1 , D_2 , D_3 and D_4 are positive (+), positive (+), negative (-), negative (-) respectively. The average sentiment for aspect “price” is positive

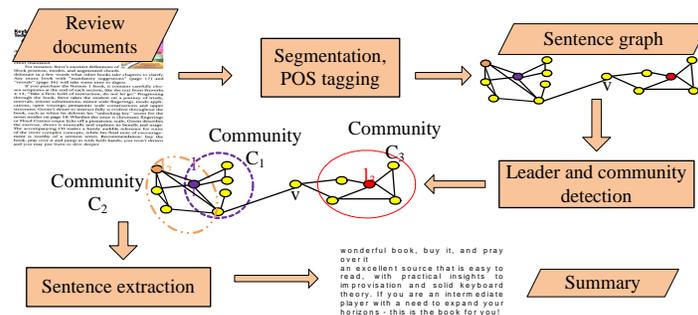


Fig. 3 An overview of the proposed opinion summarization framework

(+/-: 2/1) and is negative (+/-: 1/3) for aspect “bubble”; while the overall sentiment towards ipad protector is neutral (+/-: 2/2). A possible solution to the Problem 1 is the summary $\{s_4, s_5\}$, which looks good since it covers both aspects and preserves the overall neutral sentiment. However, this summary is misleading, especially to users who are concerned about aspect “price” (Most of the reviewers feel price is good, while this summary states a very negative opinion towards price). Instead, the summary $\{s_1, s_5\}$ that preserves the aspect-level sentiment is more meaningful and a better solution to Problem 1.

It’s non-trivial to deal with Problem 1. One may formulate it as an optimization problem $\arg \max_{\mathcal{S} \subseteq \mathcal{R}} f(\mathcal{S})$ where f denotes a scoring function over possible summaries. The definition of f can take the aspect coverage and aspect-level sentiment difference between summary \mathcal{S} and review set \mathcal{R} into consideration. However, since both aspect set and aspect-level sentiment are unknown, it is difficult to estimate either the aspect coverage or the sentiment difference, not to mention to embed them into f . Besides, even solving f is possible, usually tackling optimization problem is typically NP-hard.

Another method to solve Problem 1 is to group sentences towards similar aspects into a cluster, and select representative sentences from each group to generate summaries.

Generally, our solution is a combination of these two methods. An overview of our proposed framework is summarized in Figure 3, where we also group sentences into communities and extract informative sentences from communities by solving an optimization problem. Instead of using the content information (e.g., term vectors) to group sentences, we utilize the term similarity and sentiment polarity distributions to build graphs and then group sentences based on structure proximity. Since the sentence graph is built on texts, it identifies connections between various sentences in a corpus, and implements the concept of recommendation. The nodes that are highly recommended by other nodes in the sentence graph are likely to be more informative for the given corpus. Therefore, with the sentence graph, the informative sentence

selection problem can be formulated as a leader identification problem in the sentence graph. We then propose two algorithms to detect communities (a group of sentences S_i which are related to a specific aspect a_i and have similar sentiment polarity distributions towards a_i) and leaders (informative sentences). After that, a set of informative sentences are extracted from each community and a system summary is generated accordingly. We discuss the details in the next section.

4 Methodologies

In this section, we first provide the details of sentence graph computation in Section 4.1. Then in Section 4.2, we introduce a clique-based community and leader detection algorithm, where each maximal clique represents a community and leaders are detected by solving budgeted maximum coverage problem. Next in Section 4.3, we propose an algorithm which simultaneously identify both communities and leaders. Finally, we summarize our opinion summarization framework in Section 4.4.

4.1 Sentence-Graph Construction

Denote $G=(V, E)$ the sentence graph constructed from the set of sentences $S=\{s_1, s_2, \dots, s_n\}$, where each node $v \in V$ represents a sentence and each weighted edge $e \in E$ evaluates the similarity between the two corresponding sentences. A key research issue in sentence graph construction is to design a function to measure similarity between sentences. Before presenting the similarity function we used in this paper, we first introduce two definitions.

Definition 1. (Term Similarity) Given two sentences s_i and s_j , their term similarity is defined as

$$\tau(s_i, s_j) = \cos(\vec{v}_i, \vec{v}_j),$$

where \vec{v}_i and \vec{v}_j are the term vector representations of s_i and s_j respectively, and $\cos(\cdot)$ denotes the cosine similarity function.

Definition 2. (Adjective Orientation Similarity) The adjective orientation similarity of two sentences s_i and s_j is defined by the following equation,

$$\alpha(s_i, s_j) = 1 - \frac{|\sum_{t_i \in s_i} SO(t_i) - \sum_{t_j \in s_j} SO(t_j)|}{|\sum_{t_i \in s_i} SO(t_i) + \sum_{t_j \in s_j} SO(t_j)|},$$

where $t_i \in s_i$ (or $t_j \in s_j$) denotes an adjective term in sentence s_i (or s_j), and $SO(t_i)$ (or $SO(t_j)$) denotes the probability of t_i (or t_j) being positive, which is derived from the Semantic Orientation Dictionary [37].

As mentioned in the previous section, we aim to group the sentences that are towards the same aspect and have similar sentiment polarity orientation into a community. Therefore, the above two similarities are both important for constructing the sentence-graph. As a result, we define our similarity function between sentences as follows,

$$\mathbf{sim}(s_i, s_j) = \lambda\tau(s_i, s_j) + (1 - \lambda)\alpha(s_i, s_j), \quad (1)$$

where $\lambda \in [0, 1]$ is a trade-off parameter to control the contribution balance between the term and adjective orientation similarities.

Given the similarity function, we link sentences s_i and s_j with an edge associated with a nonnegative weight w_{ij} as follows,

$$w_{ij} = \begin{cases} \mathbf{sim}(s_i, s_j), & \text{if } s_i \in \mathcal{N}_k(s_j) \text{ or } s_j \in \mathcal{N}_k(s_i), \\ 0, & \text{otherwise,} \end{cases}$$

where $\mathcal{N}_k(s_j)$ is the k -nearest neighbors of the sentence s_j according to the similarity measure². From the preliminary test, we use a grid search to find the best combination for λ and k . The optimal values we found are $\lambda = \frac{2}{3}$ and $k = \lceil \frac{N}{5} \rceil$, where $N = |\mathcal{R}|$. Therefore, for all the experiments in this paper, we set $\lambda = \frac{2}{3}$ and $k = \lceil \frac{N}{5} \rceil$.

Example 2. Figure 2 shows an example of four review documents for iPad protector with 6 sentences in total. The associated sentence graph with $k=2$ is constructed in Figure 4(a). Each node is linked with its k -nearest neighbors as well as reversed k -nearest neighbors. For clearer representation, the figure only shows the edges but the weights are omitted. Note that not every node has the same degree k since a node can be reversed k -nearest neighbors of many nodes.

4.2 Clique-based Community and Leader Detection Algorithm (CCL)

Intuitively, since edges in sentence graph are created based on the similarity of sentences, we can make the assumption that a group of highly connected sentences are more likely to share the same topic. Therefore, we find the set of all maximal cliques in sentence graph and each maximal clique forms a community.

More Specifically, given a graph G , a clique in G is a subset of vertices, $C \subseteq V$, such that the induced subgraph by C is a complete graph in G . C

² Note that $|\mathcal{N}_k(s)|$ can be larger than k since there could be the event of ties (i.e., a set of neighbors have the same similarity to s)

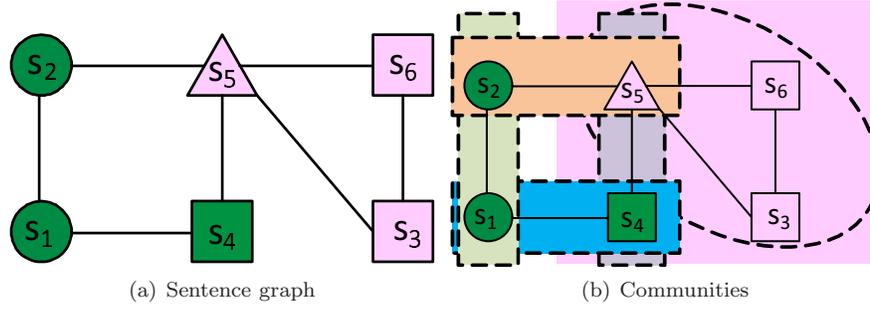


Fig. 4 Figure 4(a) shows the sentence graph of Figure 2. Circles represent sentences with aspect “price”, squares represent sentences with aspect “bubble”, and triangles represent sentences with both aspects. Green nodes denote sentences with positive sentiments and pink nodes denote sentences with negative sentiments. A set of overlapping communities, $C = \{\{s_3, s_5, s_6\}, \{s_1, s_2\}, \{s_1, s_4\}, \{s_2, s_5\}, \{s_4, s_5\}\}$, which is computed based on maximal cliques, is shown in Figure 4(b).

is called a maximal clique (maxclique in short) in G if there exists no clique C' in G such that $C' \supset C$. For example, consider the sentence graph shown in Figure 4(a), the set of all maximal cliques are $\{s_3, s_5, s_6\}$, $\{s_1, s_2\}$, $\{s_1, s_4, s_4\}$, $\{s_2, s_5\}$ and $\{s_4, s_5\}$. Therefore, we can compute a set of overlapping communities based on maximal cliques, as shown in Figure 4(b).

In the development of this system, we adopt an efficient algorithm proposed by Cheng et al. [6] to find the set of all maximal cliques.

Once we have a set of overlapping communities, the next focus is to identify a set of leaders (i.e., informative sentences) to generate a concise summary. Recall in Section 1, we have raised two critical issues for a concise summary: redundancy and converge. Thus, we investigate two principles that a set of leaders should have: good aspect coverage and informativeness. Aspect coverage accesses whether the set of selected leaders \mathcal{S} have well captured all the communities representing subtopics, while informativeness evaluates whether the selected leaders well represent the communities they belong to. In addition, since users demand a concise summary, the size of a summary (i.e., total number of words) can not exceed a given budget.

Intuitively, if we know the informativeness of each node (e.g., relative importance score) in the community, we then start picking up high informative nodes from each community until all the communities are covered or the size of summary reaches the budget. This discussion motivates us to formulate the leader detection problem as a budgeted maximum cover problem [1] as follows:

Problem 2. (*Leader Detection Problem*) Given a sentence graph $G=(V, E)$ where each sentence s is associated with a penalty cost $w(s)$ and a informativeness score $\varphi(s)$, its overlapping communities $P = \{C_1, C_2, \dots, C_m\}$ where each group of sentences C_i ($i=1$ to m) represents a subtopic, and a number

Algorithm 1 Clique-based Leader Detection $\text{CCL}(V, P, w, \varphi)$

Input: sentence set V , clique-based communities P , cost function w and φ , budget \mathcal{B}
Output: a subset sentences \mathcal{S}
1: $\mathcal{S}=\emptyset, X'=\emptyset$
2: **repeat**
3: $s^*=\arg \max_{v \in (V \setminus X')} \left\{ \frac{\varphi((\mathcal{S} \cup v) \cap P) - \varphi(\mathcal{S} \cap P)}{w(v)} \right\}$ subject to $w(\mathcal{S} \cup \{v\}) \leq \mathcal{B}$
4: $\mathcal{S}=\mathcal{S} \cup \{s^*\}$
5: **for** each $V_i \in P$
6: **if** $s^* \in V_i$
7: $X'=X' \cup V_i$
8: **until** $w(\mathcal{S}) \geq \mathcal{B}$ or $X' == V$
9: **return** \mathcal{S}

\mathcal{B} , the leader detection problem is to find a subset of sentences $\mathcal{S} \subseteq V$ such that the cost of \mathcal{S} is within budget ($w(\mathcal{S}) \leq \mathcal{B}$) and the reward of covering communities (which is denoted as $\varphi(P \cap \mathcal{S})$) is maximized.

Naturally, in this work, the penalty cost $w(s)$ of each sentence s is defined as the total number of words in the sentence s . Regarding the informativeness score $\varphi(s)$, as we know, the centrality of nodes in a community measures the relative importance of nodes within the group. Therefore, we consider a sentence to be informative if it has high centrality within its community in the sentence-graph. There are many measures of centrality that could be parameters to the algorithm, namely degree, betweenness [10], closeness [35] and eigenvector centrality measures[32]. We experimented with all of them and based on our results, we selected the degree centrality for the default measure which yields the most accurate results in most of the cases and also is easy to compute.

The degree centrality of the node v within the community C is simply the number of edges from the community incident upon the node v and represents to some extent the ‘‘popularity’’ of v in the community. That is,

$$\text{deg}(v, C) = \frac{\sum_{u \in C} w(u, v)}{|C| - 1}$$

where each edge (u, v) denotes an edge in C that is incident to node v , and w is the weight of the edge.

Since a sentence may be inside more than one community of P in the sentence graph G , we then further define $\varphi(s)$ as

$$\varphi(s) = \frac{1}{|\mathcal{C}_s|} \sum_{C \in \mathcal{C}_s} \text{deg}(s, C) \quad (2)$$

where \mathcal{C}_s denotes a set of communities which contain sentence s .

We have discussed the details of penalty cost $w(s)$ and informativeness of sentence $\varphi(s)$, now let us turn our attention to the solution of Problem 2.

Unfortunately, the budgeted maximum cover problem is known to be NP-hard for general graphs and approximation algorithms are needed [1, 18]. Hence, we develop a greedy algorithm, which iteratively adds an important but cheap node, to solve Problem 2. The details are shown in Algorithm 1.

Start with an empty sentence set \mathcal{S} (line 1), in each iteration, this greedy algorithm picks up a sentence s^* from those uncovered partitions which maximizes the marginal gain (lines 3–4). Furthermore, after s^* is added to \mathcal{S} , it is required to update the set of covered communities: all the communities that contain s^* can be marked as covered (lines 5–7). The algorithm stops and returns \mathcal{S} until the budget is exhausted or all the communities are covered (lines 8–9).

Bounds of the Greedy Algorithm. Khuller et al. [18] have proved that for nondecreasing reward φ and nonnegative penalty cost w , there exists a greedy algorithm with an approximation factor of $\frac{1}{2}(1-\frac{1}{e})$. Note that in Algorithm 1, φ is nondecreasing and w is nonnegative, hence, following the proof in [18], we show that Algorithm 1 achieves an approximation factor of $\frac{1}{2}(1-\frac{1}{e})$ as well. And the worst case running time of this algorithm is bounded by $O(\mathcal{B}|V|)$ where $|V| = |\mathcal{R}|$ denotes total number of sentences in review set.

4.3 Simultaneous Community and Leader Detection Algorithm (SCL)

In the previous section, we have proposed a sequential algorithm to first identify communities by enumerating all maximal cliques and then detect leaders by solving the budgeted maximum coverage problem. However, there are some limitations of the proposed CCL algorithm: first, the size of leader set (i.e., summary) highly depend on the parameter \mathcal{B} . Next, in the CCL Algorithm, the leader sentences are selected according to the only two criteria aspect coverage and representativeness. In real application such as Amazon, each review may have a helpful vote number which indicates the quality of review itself. We suggest that the quality of review is helpful for identifying informative sentences, with the assumption that a sentence from a more helpful review is more informative than another low quality review.

To address the aforementioned issues, we propose an alternative leader detection algorithm, namely, simultaneous community and leader detection algorithm (SCL). The general idea is similar to the k -mean clustering: we first initialize a set of leaders with high degrees and then assign other nodes to each leader to form communities. Given each community, we then update its leaders based on the informativeness of sentences within both communities and reviews. We iteratively repeat the above process until there is no change in leadership. An overview of SCL algorithm is outlined in Algorithm 2.

Algorithm 2 Simultaneous Community and Leader Detection $SCL(G)$

Input: a sentence-graph G and review score $\varphi(D)$ for each review D
Output: a subset of sentences \mathcal{S}

- 1: Initialize a set of leader \mathcal{S} in G : $\mathcal{S} = LL(G)$ (Alg. 3)
- 2: Initialize communities \mathcal{C} by assigning each leader to a single community
- 3: **repeat**
- 4: let followers $F = \{v \in V | v \notin \mathcal{S}\}$ and order $v \in F$ by its distance to \mathcal{S}
- 5: **for** each $v \in F$
- 6: update community set: $\mathcal{C} = \text{Community}(G, \mathcal{S}, v, \mathcal{C})$ (Alg. 4)
- 7: **for** each sentence $s \in \mathcal{S}$ with $s \in D$
- 8: update community leader for $C(s)$ and review score $\varphi(D)$ (Equ. 3)
- 9: update \mathcal{S} as new leaders of each community of \mathcal{C} (Equ. 4)
- 10: **until** no change in the leader list \mathcal{S}
- 11: **return** \mathcal{S}

There are several advantages of the proposed SCL algorithm in terms of parameter-free property and efficiency. First, the number of leaders and the size of summary is automatically determined by Algorithm 2. There is no requirement for additional parameters such as \mathcal{B} in the CCL algorithm. Next, instead of first using the very time-consuming maximal clique enumeration approach to find communities and then another approach to detect leaders in the CCL algorithm, the SCL algorithm efficiently determines communities and leaders simultaneously in a unified framework. Finally, in the SCL algorithm, the leader sentences are selected based on not only their informativeness in communities but also qualities of reviews they belong to.

We now present more details about each important step in the Algorithm 2: Leader initialization, Community assignment and Leader reassignment.

4.3.1 Leader Initialization

Once the sentence-graph is built, we can initialize some nodes of the graph as leaders and iteratively identify and update the communities and leaders. The naïve initialization is to randomly select k sentences from the sentence-graph as leaders. This is simple to implement, but is not deterministic and may produce unexpected results. Another approach is to select a set of global top sentences such as selecting k sentences that have highest degrees in the sentence-graph. However, choosing arbitrarily top- k high-degree sentences may suffer from the redundancy and low coverage issues. An extreme case is that all of the top- k sentences are discussing about the same aspect and hence the results are not satisfied.

As an alternative, we want to select a set of leader sentences that are well distributed in the sentence-graph (i.e., to avoid choosing leaders from the same community). More specifically, a node v in the sentence-graph is selected as an initial leader if

Algorithm 3 Leader Initialization $LL(G)$

Input: Graph $G=(V, E)$ **Output:** a set of leaders L

```

1:  $L=\emptyset$ 
2: Compute the set of  $h$ -nodes  $H \in V$  and sort  $H$  by node degree in descending order
3: while  $H$  is not empty
4:   pick  $v$  from the front of  $H$ 
5:    $H=H \setminus \{v\}$ ,  $\text{flag}=\text{true}$ ;
6:   for each  $s \in L$ 
7:     if  $v$  is a neighbor of  $s$ 
8:        $\text{flag}=\text{false}$ ;
9:   if  $\text{flag}==\text{true}$ 
10:     $L=L \cup \{v\}$ 
11: return  $L$ 

```

1. It is a h -node in sentence graph G , and
2. None of its neighbors is a leader.

The key component of our lead initialization is the largest set of h nodes in sentence graph G that have degree at least h , called the h -node[6]. The concept of h -node is originated from the the h -index [13] that attempts to measure both the productivity and impact of the published work of a scientist or scholar. Putting it into the concept of our sentence graph, a h -node in sentence graph corresponds to a sentence that is similar to at least another h sentences and to a certain extent represents the “ground truth”. Therefore, it is straightforward to adopt the h -node concept for initial leadership evaluation. Note that the h value and the set of h -nodes can be computed easily using a deterministic and parameter-free algorithm proposed by [6].

Another component of our leader initialization aims to reduce redundancy and achieve better community coverage. After finding the set of h -nodes, we start from the node with highest degree, and add the next higher degree h -node to the current set of leaders if it is not a neighbor of any of the already selected leaders. All the details of the leader initialization are outlined in Algorithm 3.

4.3.2 Community Assignment

Once some leaders are initialized, we can initialize communities by assigning each leader to a single community. After that the community membership of the remaining nodes can be determined by assigning them to nearby leaders. The intuitive idea is similar to label propagation algorithms for link-based classification [27, 38], where class labels (i.e., community membership in our scenario) of linked nodes are correlated. Therefore, a node is assigned to a community if most of its neighbors have already resided in the community.

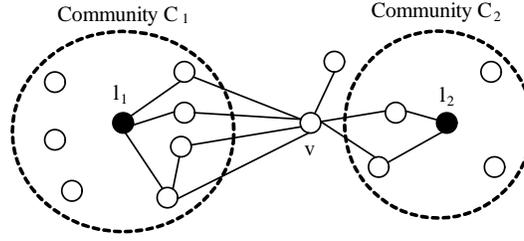


Fig. 5 An example of community membership determination

Algorithm 4 Update community $\text{Community}(G, L, v, \mathcal{C})$

Input: graph G , leaders L , node v , communities \mathcal{C}

Output: A refined community set \mathcal{C}

```

1: let  $\max = 0, l_t = -1$ 
2: for each  $l \in L$ 
3:   let  $N$  denote the set of common neighbors between  $v$  and community  $C(l)$ 
4:    $\delta = \sum_{u \in N} w(u, l) + w(u, v)$ 
5:   if  $\delta > \max$ 
6:      $\max = \delta$ 
7:      $l_t = l$ 
8: if  $l_t$  is negative
9:   mark  $v$  as outlier
10: else
11:   update  $C(l_t) \in \mathcal{C}$  to  $C(l_t) \cup \{v\}$ 
12: return  $\mathcal{C}$ 

```

Algorithm 4 presents the method to determine the community membership for a node v . Note that in Algorithm 2 (Lines 6–7), we start calling Algorithm 4 for non-leader nodes with ascending order of distances to leaders. By doing this, we iteratively propagate the community membership from leaders to royal members (i.e., neighbors of leaders), and then to the descendants of royal members (i.e., n -hop neighbors of leaders).

Example 3. Figure 5 shows an example of sentence graph with two communities formed by leader l_1 and l_2 . Assume that each edge is equally weighted, then node v should be assigned to leader l_1 since v shares more common neighbors with community C_1 than C_2 . Consider another extreme case where edges connecting v and nodes in C_1 are with weight 0.001 and edges connecting v and nodes in C_2 are with weight 0.9, then node v is assigned to leader l_2 since it is more similar to community C_2 in terms of content and polarity similarity.

4.3.3 Leader Reassignment

As we have discussed earlier, in CCL algorithm, the informativeness of a sentence is only evaluated by its degree centrality within the community. However, we argue that the informativeness of a sentence is related to not only its representative within the community, but also the quality of the review it belongs to. More specifically, we have the following two assumptions:

1. A review is important if it contains lots of informative sentences;
2. A sentence is informative if it appears in an important review.

Hence, given a sentence s from a review D , which is represented as a node v in the sentence graph and is in the community $C(s)$, the informativeness of the sentence $\varphi(s)$ is defined as follows,

$$\begin{cases} \varphi(s) = \varphi(D)\text{deg}(v, C(s)), \\ \varphi(D) = \frac{1}{|D|} \sum_{s \in D} \varphi(s), \end{cases} \quad (3)$$

where $\text{deg}(v, C(s))$ is the degree centrality of the node v within the community $C(s)$, and $\varphi(D)$ denotes the importance of a review D . Without any prior knowledge, for each review $D \in R$, we can just initialize the $\varphi(D)=1/l$ where l is number of reviews. However, when additional information such as “helpfulness” rating score of each review is known in advance, we can initialize the value of $\varphi(D)$ as the “helpfulness” score.

Based on Equ. 3, we can update the $\varphi(s)$ and $\varphi(D)$ mutually in each iteration. After that, for each community, the sentence with the highest informativeness score is selected as the new leader,

$$s^* = \arg \max_{s \in C(s)} \varphi(s) \quad (4)$$

4.4 Summary Generation

We now conclude the proposed sentence-based opinion summarization using the following example.

Example 4. Given a set of reviews shown in Figure 2, in our sentence-based opinion summarization, the first step is to construct a sentence graph shown in Figure 4(a). Next, we either use the CCL algorithm to find a set of clique-based communities, as shown in Figure 4(b) and then the sentences s_1, s_5 are extracted for summary. Or as an alternative, we use the SCL algorithm to find communities $\{\{s_1, s_2, s_4\}, \{s_5, s_3, s_6\}\}$ and leaders $\{s_1, s_5\}$ simultaneously. Both of the two algorithms result in a summary with two sentences and about 24 words. A manually generated Aspect-based summary, which can be considered as a reference summary, is “Price: 4 stars and Bubble: 1.5 stars”. We observe that our summary does not lose any aspect coverage.

Table 1 Summary of the dataset.

product	# of reviews	# of sentences $ \mathcal{R} $	% of positive reviews	# of aspects $ \mathcal{A} $
case	625	2865	83%	19
laptop	68	469	39.2%	23
iMac	34	567	74%	17
ipad	218	3572	61%	41
protector	141	953	64%	20
kindle	1858	21948	72.6%	43

In addition, there is no mismatch of sentiments for any aspect between our system summary and manual summary. Therefore, from the comparison, we can conclude that our Leader-based summary covers as many aspects as manual summary and selects most of the informative sentences. What’s more, it is more convenient to generate the manual Aspect-based summary from our system summary than from the original reviews in Figure 2.

5 Experiments

5.1 Datasets

The dataset³ used in our experiments is a collection of product reviews crawled from Amazon.com. The reviews are about six product domains: *Belkin case* (case), *Dell laptop* (laptop), *Apple iMac* (iMac), *Apple ipad* (ipad), *ipad protector* (protector), and *Kindle* (kindle). Each review contains a title, review content, reviewer information, and an overall rating. The labeling of polarity of each review mainly depends on the given overall rating. In addition, for each product domain, we manually label its aspects and the sentiment polarity towards them on each sentence. The detailed information of the dataset is summarized in Table 1.

5.2 Evaluation Metric

We evaluate the proposed method together with the three baselines using two metrics: the aspect coverage and the polarity distribution preservation.

Aspect coverage: Given the review set \mathcal{R} with a set of aspects \mathcal{A} , the aspect coverage of a summary \mathcal{S} is defined as

$$\zeta = \frac{|\{a_i | a_i \in \mathcal{A}, a_i \in \mathcal{S}\}|}{|\mathcal{A}|} \times 100\%$$

³ available at sites.google.com/site/linhongi2r/data-and-code

Note that higher value of ζ implies better aspect coverage.

Polarity distribution preservation: Given the review set \mathcal{R} and the aspect set \mathcal{A} , the aspect-level polarity distribution of \mathcal{R} can be represented as a vector $\vec{t} = (t_1, \dots, t_n)$ with length $3 \times |\mathcal{A}|$ where t_{3i-2} , t_{3i-1} and t_{3i} denote the percentage of positive, negative, and neutral sentences that are related to aspect a_i ($i=1$ to $|\mathcal{A}|$) respectively. Assume that vector \vec{t}' denotes the aspect-level polarity distribution of a summary \mathcal{S} , then its polarity distribution preservation ratio to \mathcal{R} is defined as

$$\eta = \text{corr}(\vec{t}', \vec{t})$$

where $\text{corr}(\cdot)$ denotes the Pearson correlation coefficient function. A value of $\eta \in [-1, 1]$ that is close to one means that the summary has well preserved the aspect-level polarity distribution of \mathcal{R} .

5.3 Baselines

We compare our methods, denoted by \mathcal{S}_{CCL} and \mathcal{S}_{SCL} , with other three baselines. In order to avoid length-based bias⁴, we add constraints on the number of sentences selected so that the sizes of summary returned by each baseline are roughly equal to that of \mathcal{S}_{SCL} . For \mathcal{S}_{CCL} , we report the result when $\mathcal{B} = |\mathcal{S}_{\text{SCL}}|$ (denoted as $\mathcal{S}_{\text{CCL}}^b$) and the optimal result ($\mathcal{S}_{\text{CCL}}^*$) in terms of both aspect coverage and polarity preservation achieved by varying \mathcal{B} .

- Aspect-based sentence selection (\mathcal{S}_a): In aspect-based sentence selection, we assume that a set of aspects are given as inputs. Therefore, we read the manually labeled aspect lists as an input, group sentences towards the same aspect into a same cluster, and select a number of representative sentences (i.e., a set of sentences that are most similar to other sentences in the same cluster) from each cluster C with probability $p_1 = \frac{|C|}{|\mathcal{R}|}$, which implies that for hot aspects, more sentences would be selected. The extraction is terminated when the size of selected sentences reaches the size of sentences selected by $|\mathcal{S}_{\text{SCL}}|$.
- Position-based sentence selection (\mathcal{S}_p): In position-based sentence selection, sentences are selected from the beginning and ending positions of each review document/paragraph, assuming that the locations are related to the likelihood of the sentences of being chosen for summarization [2].
- Ranking-based sentence selection (\mathcal{S}_r): After computing the sentence graph, ranking-based sentence selection uses graph-based ranking techniques [30] to sort sentences in a reversed order based on their scores, and the top ranked sentences are selected. The number of selected sentences is equal to that in \mathcal{S}_{SCL} .

⁴ A longer summary is more likely to provide better information but is less concise

5.4 Quantitative Evaluation

Table 2 The size of summary.

	case	laptop	iMac	ipad	protector	kindle
$ \mathcal{S}_{\text{SCL}} $	96	27	44	94	69	234
$\frac{ \mathcal{S}_{\text{SCL}} }{ \mathcal{R} }$	3.35%	5.76%	7.76%	2.63%	7.24%	1.07%
$ \mathcal{S}_{\text{CCL}^*} $	183	104	40	303	108	544
$\frac{ \mathcal{S}_{\text{CCL}^*} }{ \mathcal{R} }$	6.39%	22.17%	7.05%	8.48%	11.33%	2.48%

Firstly, we report the number of sentences in summary returned by \mathcal{S}_{SCL} and $\mathcal{S}_{\text{CCL}^*}$ in Table 2. Note that we do not report the size of other summaries since they are either equal to or very similar to the size of \mathcal{S}_{SCL} . In terms of concise, \mathcal{S}_{SCL} summary which is able to achieve 92% compression ratio in the worst case, is significantly better than the $\mathcal{S}_{\text{CCL}^*}$ summary.

Table 3 Aspect coverage ζ comparison.

product	\mathcal{S}_{SCL}	$\mathcal{S}_{\text{CCL}^*}^*$	$\mathcal{S}_{\text{CCL}^b}^b$	\mathcal{S}_a	\mathcal{S}_p	\mathcal{S}_r
case	94%	95%	88%	100%	88%	76%
laptop	90%	93%	86%	85%	65%	86%
iMac	88%	83%	83%	94%	47%	47%
ipad	97%	95%	88%	94%	85%	69%
protector	84%	89%	82%	84%	37%	82%
kindle	94%	98%	91%	97%	88%	91%
average	91%	92%	86%	92%	68%	75%

Table 4 Polarity preservation η comparison.

product	\mathcal{S}_{SCL}	$\mathcal{S}_{\text{CCL}^*}^*$	$\mathcal{S}_{\text{CCL}^b}^b$	\mathcal{S}_a	\mathcal{S}_p	\mathcal{S}_r
case	0.93	0.99	0.91	0.78	0.92	0.84
laptop	0.98	0.93	0.73	0.61	0.64	0.63
iMac	0.79	0.97	0.46	0.14	0.24	0.51
ipad	0.97	0.90	0.84	0.57	0.9	0.66
protector	0.87	0.84	0.33	0.48	0.45	0.73
kindle	0.85	0.97	0.76	0.68	0.76	0.80
average	0.9	0.93	0.67	0.54	0.65	0.70

Next, we study how the proposed method performs with respect to the aspect coverage ζ . The results are reported in Table 3. The baseline \mathcal{S}_a is supposed to maximize the aspect coverage and achieve 100% coverage. However, with the usage of the probing probability p_1 , some unpopular aspect is missing in \mathcal{S}_a . Therefore, \mathcal{S}_a achieves only 92% coverage on average. Regarding leader-based summaries, \mathcal{S}_{SCL} performs better than $\mathcal{S}_{\text{CCL}^b}^b$, but slightly worse than $\mathcal{S}_{\text{CCL}^*}^*$. This is understandable since the size of summary outputted by $\mathcal{S}_{\text{CCL}^*}^*$ is much longer than that of \mathcal{S}_{SCL} . Furthermore, from the results, we can find that aspect coverage of leader-based summaries (\mathcal{S}_{SCL} , $\mathcal{S}_{\text{CCL}^*}^*$, $\mathcal{S}_{\text{CCL}^b}^b$) is comparable to that of \mathcal{S}_a on average. On some product domains such as Dell laptop and ipad, leader-based summaries are even better. Ranking-based method \mathcal{S}_r , performs worse than both \mathcal{S}_a and leader-base summaries, but has much better aspect coverage than \mathcal{S}_p . These results indicate that the proposed methods \mathcal{S}_{SCL} , $\mathcal{S}_{\text{CCL}^b}^b$, and $\mathcal{S}_{\text{CCL}^*}^*$ perform well in terms of aspect coverage ζ .

Finally, we compare the performance of different methods for opinion summarization in terms of polarity distribution preservation ratio η . The goal of this experiment is to evaluate whether the summary generated by different

methods can preserve the polarity distribution of each aspect of the original reviews \mathcal{R} . The results are shown in Table 4. As can be seen from the table, our proposed method \mathcal{S}_{SCL} , $\mathcal{S}_{\text{CCL}}^b$, and $\mathcal{S}_{\text{CCL}}^*$ obtain much better results than other baselines and can preserve polarity distribution of the original reviews in the aspect level. The Aspect-based sentence selection method \mathcal{S}_a may select a number of very popular sentences but express redundant viewpoints towards a specific aspect, which results in that the polarity distribution of the selected sentences within an aspect may easily get skewed. Surprisingly, from the table we find that the Position-based method \mathcal{S}_p does not perform worst in terms of polarity distribution preservation. A possible reason is that usually the first or last sentences in a paragraph/review are likely to express a viewpoint towards an entity, such as “Overall , 5 stars for the price!”. As a result, the sentences selected by \mathcal{S}_p can obtain reasonable performance in terms of polarity distribution preservation.

In the above study, both Table 3 and Table 4 show that $\mathcal{S}_{\text{CCL}}^*$ outperforms \mathcal{S}_{SCL} in terms of aspect coverage and polarity preservation. Since the CCL algorithm well preserves both aspect coverage and polarity distribution, we recommend the proposed opinion summarization system to use the CCL algorithm when aspect-based summarization is required to be post-applied to sentence-based summaries before displaying to users.

5.5 User Study

In the previous section we investigated how different methods perform in terms of aspect coverage and polarity distribution preservation. In this section, we perform a user study to understand how useful the sentence selected by different methods are to actual users. An ideal way to conduct the user study is to let users select sentences for summarization manually as references and then evaluate the similarity between the references and different system summaries using ROUGE-N [24]⁵. However, for our dataset, it is difficult to generate summaries manually especially for some product domain where the size of reviews is up to 21948 sentences. Instead, we asked a number of humans to express their preference for one summary over another one. Each person is required to conduct 20 groups of rating and in each group two summaries of the same product are placed side-by-side in a random order. We did not ask users to rate $\mathcal{S}_{\text{CCL}}^b$ since $\mathcal{S}_{\text{CCL}}^*$ is consistently better than $\mathcal{S}_{\text{CCL}}^b$, as defined.

The results of judgment agreement and preference evaluation are reported in Table 5, where “agreement” is the percentage of items for which all raters agreed on a positive/negative/no-preference rating while “prefer A/B” is the percentage of agreement items in which the raters prefer either A or B re-

⁵ ROUGE-N is a popular toolkit which measures the quality of a summary by comparing it to other reference summaries using n -gram con-occurrence.

Table 5 Results of user evaluation experiments

Comparisons (A V.S. B)	Agreement	Prefer A	Prefer B	Equal
\mathcal{S}_{SCL} V.S. \mathcal{S}_a	86.0%	72.4%	16.3%	11.3%
\mathcal{S}_{SCL} V.S. \mathcal{S}_{CCL}^*	75.0%	70.1%	12.0%	17.9%
\mathcal{S}_{SCL} V.S. \mathcal{S}_p	84.0%	73.2%	13.4%	13.4%
\mathcal{S}_{SCL} V.S. \mathcal{S}_r	85.3%	89.7%	2.0%	8.3%
\mathcal{S}_{CCL}^* V.S. \mathcal{S}_a	80.0%	67.3%	15.2%	17.5%
\mathcal{S}_{CCL}^* V.S. \mathcal{S}_p	86.4%	66.6%	12.2%	21.2%
\mathcal{S}_{CCL}^* V.S. \mathcal{S}_r	84.4%	88.7%	2.0%	9.3%
\mathcal{S}_a V.S. \mathcal{S}_p	65.0%	56.5%	34.5%	9.0%
\mathcal{S}_a V.S. \mathcal{S}_r	62.0%	64.7%	28.7%	6.6%
\mathcal{S}_p V.S. \mathcal{S}_r	68.1%	45.3%	48.4%	6.3%

spectively. As can be observed that the proposed methods are much better than the other baselines. More than 66.6% comparison judges show that the Leader-based summaries (\mathcal{S}_{CCL}^* and \mathcal{S}_{SCL}) are better than all the other baselines while the agreement is also up to 80%. In addition, users prefer the summary outputted by \mathcal{S}_{SCL} than \mathcal{S}_{CCL}^* . One possible reason is that usually summaries generated by \mathcal{S}_{CCL}^* are much longer and hence result in lower scores in readability and conciseness. Therefore, we recommend the proposed system to use the SCL algorithm due to its good trade-off between conciseness and aspect coverage when the sentence-based summary is directly displayed to users.

For the remaining baselines, there is no obvious winner among them except that the Aspect-based approach \mathcal{S}_a is more preferred than the Ranking-based approach \mathcal{S}_r . The reason may be that each baseline is designed to optimize a specific measure (e.g., ranking-based method is proposed to optimize the informativeness) while the user quality study is evaluated over a combination of criteria. In contrast, our proposed methods aim to select informative sentences by optimizing aspect coverage and preserving polarity distribution simulatively, which may be more desirable for users' demand.

6 Conclusions and future works

In this paper, we have developed an effective framework for informative sentence selection for opinion summarization. The informativeness of sentences is evaluated in terms of aspect coverage and viewpoints coverage. To this end, we have formulated the informative sentence selection problem as a community leader detection problem in sentence-graph, where edges encode the term similarity and viewpoint similarity of sentences. Next, we have presented two effective algorithms to find the leaders (informative sentences) and communities (sentences with similar aspects and viewpoints). A set of systematic

evaluation as well as quality evaluation verified that the proposed methods are able to achieve good performance.

Though the primary focus of this paper is opinion summarization, our approach is also applicable to other opinion mining problems. Therefore, one avenue for the future work is to exploit our sentence extraction method for other tasks such as spam review detection. In addition, in this paper, we conduct a set of empirical studies on product review data. In the future, we also plan to extend our methods to different domains such as twitter data, conversation, and political forum data.

Acknowledgment

This work is partially supported by DARPA under grant Number W911NF-12-1-0034.

References

1. Ageev, A.A., Sviridenko, M.: Approximation algorithms for maximum coverage and max cut with given sizes of parts. In: Proceedings of the 7th International Conference on Integer Programming and Combinatorial Optimization, pp. 17–30. Springer-Verlag, London, UK (1999)
2. Beineke, P., Hastie, T., Manning, C., Vaithyanathan, S.: Exploring sentiment summarization. In: AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications (2004)
3. Blair-goldensohn, S., Neylon, T., Hannan, K., Reis, G.A., Mcdonald, R., Reynar, J.: Building a sentiment summarizer for local service reviews. In: NLP in the Information Explosion Era (2008)
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. *The Journal of Machine Learning Research* **3**, 993–1022 (2003)
5. Bookstein, A.: Informetric distributions, part i: Unified overview. *Journal of the American Society for Information Science* **41**(5), 368–375 (1990)
6. Cheng, J., Ke, Y., Fu, A.W.C., Yu, J.X., Zhu, L.: Finding maximal cliques in massive networks by h^* -graph. In: Proceedings of the SIGMOD, pp. 447–458. ACM, New York, NY, USA (2010)
7. Danescu-Niculescu-Mizil, C., Kossinets, G., Kleinberg, J.M., Lee, L.: How opinions are received by online communities: A case study on amazon.com helpfulness votes. In: Proceedings of the 18th WWW, pp. 141–150. ACM, New York, NY, USA (2009)
8. Erkan, G., Radev, D.R.: Lexpagerank: Prestige in multi-document text summarization. In: Proceedings of EMNLP. Barcelona, Spain (2004)
9. Filippova, K.: Multi-sentence compression: Finding shortest paths in word graphs. In: COLING, pp. 322–330 (2010)
10. Freeman, L.C.: Centrality in social networks: Conceptual clarification. *Social Networks* **1**(3), 215–239 (1979)
11. Ganesan, K., Zhai, C., Han, J.: Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. In: Proceedings of the 23rd COLING (2010)

12. Heerschop, B., Goossen, F., Hogenboom, A., Frasinca, F., Kaymak, U., de Jong, F.: Polarity analysis of texts using discourse structure. In: Proceedings of the 20th CIKM, pp. 1061–1070. ACM, New York, NY, USA (2011)
13. Hirsch, J.E.: An index to quantify an individual’s scientific research output. Proceedings of the National Academy of Sciences of the United States of America **102**(46), 16,569–16,572 (2005)
14. Hofmann, T.: Probabilistic latent semantic analysis. In: Proceedings of Uncertainty in Artificial Intelligence, pp. 289–296 (1999)
15. Hu, B., Song, Z., Ester, M.: User features and social networks for topic modeling in online social media. In: ASONAM, pp. 202–209 (2012)
16. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the 10th ACM SIGKDD, pp. 168–177. ACM, New York, NY, USA (2004)
17. Jin, F., Huang, M., Zhu, X.: A comparative study on ranking and selection strategies for multi-document summarization. In: COLING (Posters), pp. 525–533 (2010)
18. Khuller, S., Moss, A., Naor, J.S.: The budgeted maximum coverage problem. Information Processing Letter **70**, 39–45 (1999)
19. Kim, H.D., Ganesan, K., Sondhi, P., Zhai, C.: Comprehensive review of opinion summarization (2011)
20. Kim, S.M., Pantel, P., Chklovski, T., Pennacchiotti, M.: Automatically assessing review helpfulness. In: Proceedings of EMNLP, pp. 423–430. Association for Computational Linguistics, Stroudsburg, PA, USA (2006)
21. Lerman, K., Blair-Goldensohn, S., McDonald, R.: Sentiment summarization: evaluating and learning user preferences. In: Proceedings of the 12th EACL, pp. 514–522. ACL, Stroudsburg, PA, USA (2009)
22. Li, F., Huang, M., Yang, Y., Zhu, X.: Learning to identify review spam. In: IJCAI, pp. 2488–2493 (2011)
23. Lim, E.P., Nguyen, V.A., Jindal, N., Liu, B., Lauw, H.W.: Detecting product review spammers using rating behaviors. In: Proceedings of the 19th CIKM, pp. 939–948. ACM, New York, NY, USA (2010)
24. Lin, C.Y., Hovy, E.: Automatic evaluation of summaries using n-gram co-occurrence statistics. In: Proceedings of the NAACL, pp. 71–78. ACL, Stroudsburg, PA, USA (2003)
25. Lin, H., Bilmes, J.: A class of submodular functions for document summarization. In: Proceedings of the 49th HLT/ACL, pp. 510–520. ACL, Stroudsburg, PA, USA (2011)
26. Liu, J., Cao, Y., Lin, C.Y., Huang, Y., Zhou, M.: Low-Quality product review detection in opinion summarization. In: Proceedings of the Joint Conference on EMNLP-CoNLL, pp. 334–342 (2007)
27. Lu, Q., Getoor, L.: Link-based classification. In: Proceedings of the 20th ICML, pp. 496–503. AAAI Press, Chicago, IL, USA (2003)
28. Lu, Y., Zhai, C., Sundaresan, N.: Rated aspect summarization of short comments. In: Proceedings of the 18th WWW, pp. 131–140. ACM, New York, NY, USA (2009)
29. Mei, Q., Ling, X., Wondra, M., Su, H., Zhai, C.: Topic sentiment mixture: modeling facets and opinions in weblogs. In: Proceedings of the 16th WWW, pp. 171–180. ACM, New York, NY, USA (2007)
30. Mihalcea, R., Tarau, P.: Textrank: Bringing order into text. In: EMNLP, pp. 404–411 (2004)
31. Muthukrishnan, P., Gerrish, J., Radev, D.R.: Detecting multiple facets of an event using graph-based unsupervised methods. In: COLING, pp. 609–616 (2008)
32. Newman, M.E.J.: The mathematics of networks. The New Palgrave Encyclopedia of Economics pp. 1–12 (2007)
33. Pang, B., Lee, L.: A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42nd ACL. Association for Computational Linguistics, Stroudsburg, PA, USA (2004)

34. Popescu, A.M., Etzioni, O.: Extracting product features and opinions from reviews. In: Proceedings of the HLT and EMNLP, pp. 339–346. Association for Computational Linguistics, Stroudsburg, PA, USA (2005)
35. Sabidussi, G.: The centrality index of a graph. *Psychometrika* **31**(4), 581–603 (1966)
36. Smith, L.M., Zhu, L., Lerman, K., Kozareva, Z.: The role of social media in the discussion of controversial topics. In: SocialCom, pp. 236–243 (2013)
37. Taboada, M., Anthony, C., Voll, K.: Methods for creating semantic orientation dictionaries. In: Proceedings of 5th ICLRE, pp. 427–432. Genoa, Italy (2006)
38. Taskar, B., Wong, M., Abbeel, P., Koller, D.: Link prediction in relational data. In: NIPS. Cambridge, MA: MIT Press (2004)
39. Titov, I., McDonald, R.T.: A joint model of text and aspect ratings for sentiment summarization. In: ACL, pp. 308–316 (2008)
40. Tsaparas, P., Ntoulas, A., Terzi, E.: Selecting a comprehensive set of reviews. In: Proceedings of the 17th ACM SIGKDD, pp. 168–176. ACM, New York, NY, USA (2011)
41. Wan, X., Yang, J.: Multi-document summarization using cluster-based link analysis. In: Proceedings of the 31st ACM SIGIR, pp. 299–306. ACM, New York, NY, USA (2008)
42. Wang, D., Li, T.: Document update summarization using incremental hierarchical clustering. In: Proceedings of the 19th CIKM, pp. 279–288. ACM, New York, NY, USA (2010)
43. Yu, J., Zha, Z.J., Wang, M., Chua, T.S.: Aspect ranking: Identifying important product aspects from online consumer reviews. In: ACL, pp. 1496–1505. The Association for Computer Linguistics (2011)
44. Zhu, L., Galstyan, A., Cheng, J., Lerman, K.: Tripartite graph clustering for dynamic sentiment analysis on social media. In: SIGMOD Conference, pp. 1531–1542 (2014)
45. Zhu, L., Galstyan, A., Cheng, J., Lerman, K.: Tripartite graph clustering for dynamic sentiment analysis on social media. *CoRR* **abs/1402.6010** (2014)
46. Zhu, L., Gao, S., Pan, S.J., Li, H., Deng, D., Shahabi, C.: Graph-based informative-sentence selection for opinion summarization. In: ASONAM, pp. 408–412 (2013)
47. Zhuang, L., Jing, F., Zhu, X.Y.: Movie review mining and summarization. In: Proceedings of the 15th CIKM, pp. 43–50. ACM, New York, NY, USA (2006)