Graph-based Informative-Sentence Selection for Opinion Summarization

Linhong Zhu
Information Sciences Institute
University of Southern California
linhong@isi.edu

Sheng Gao, Sinno Jialin Pan, Haizhou Li
Institute for Infocomm Research
A-STAR, Singapore
{gaosheng,jspan,hli}@i2r.a-star.edu.sg

Dingxiong Deng and Cyrus Shahabi
Computer Science Department
University of Southern California
{dingxiong.deng, shahabi}@usc.edu

***Abstract*—In this paper, we propose a new framework for opinion summarization based on sentence selection. Our goal is to assist users to get helpful opinion suggestions from reviews by only reading a short summary with 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. The detected leaders of the communities can be considered as the most informative sentences of the corresponding aspect, while informativeness of a sentence is defined by its informativeness within both its community and the document it belongs to. Review data from six product domains from Amazon.com are used to verify the effectiveness of our method for opinion summarization.**

I. Introduction

Nowadays, the flourish of online opinions 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 hard for potential customers to go through all the reviews to make an informed decision on purchase. Furthermore, some reviews are noninformative and may 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 in aspect-based opinion summarization [1], [2], [3], [4], [5], [6] and review quality evaluation [7], [8], [9], [10].

Aspect-based opinion summarization aims to identify aspects of a given entity, and summarize the overall sentiment orientation towards each aspect. 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 such as [11], [12], [13], [14] 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 this paper, we propose a new opinion summarization framework, named *sentence-based opinion summarization*, to address these issues. Given a set of reviews for a specific entity, our goal is 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 our 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.

Based on the new 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 leader detection problem in social computing. We first construct a sentence-graph by adding an edge between a pair of sentences if they are similar in word distribution and sentiment polarity distribution. Here each node of the graph representing a sentence can be considered as a user in social computing. Finally, we propose an algorithm to detect *leaders* and *communities* simultaneously on the sentence-graph, where 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.

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 representative 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 an effective leader-community detection algorithm based on both the structure of graph and the context information of review documents.

Algorithm 1 Sentence-based opinion summarization

Input: a set of review $\mathcal{R}$
Output: a subset of sentences $\mathcal{S}$
1: Construct sentence-graph G from $\mathcal{R}$ (Sec. III-A)
2: Initialize a set of leader $\mathcal{S}$ in G (Sec. III-B)
3: Initialize review importance scores (Sec. III-D)
4: **repeat**
5: let F=$\{v \in V | v \notin \mathcal{S}\}$
6: order $v \in F$ by its distance to $\mathcal{S}$
7: **for** each $v \in F$
8: assign v to a leader (Sec. III-C)
9: **for** each $s_l \in \mathcal{S}$
10: update leaders for $C(s_l)$ (Sec. III-D)
11: update review importance score (Sec. III-D)
12: **until** no change in the leader list $\mathcal{S}$
13: **return** $\mathcal{S}$

II. PROBLEM FORMULATION

Denote x a specific entity that consists of a set of aspects $\mathcal{A}$ = $\{ a_1, a_2, \ldots, a_m \}$, and a set of reviews on the entity $\mathcal{R}$=$\{D_1, D_2, \ldots, D_l\}$, where D_i (i = 1 to l) represents a review. Each review D_i consists of several sentences D_i=$\{s_1, s_2, \ldots, s_{n_i}\}$, 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 aspect set $\mathcal{A}$ and their sentiments are unknown in training.

A. Our Proposed Framework

It's non-trivial to deal with the 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 differences between summary $\mathcal{S}$ and review set $\mathcal{R}$ into consideration. However, since both aspect set and aspect-level sentiment is unknown, it is difficult to estimate either the aspect coverage or the sentiment differences, 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 the 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 belongs to this category but is very unique. The first step is to construct a sentence-graph G from the review set $\mathcal{R}$ (Line 1 in Algorithm 1). In the sentence-graph G, each node is a sentence and each edge represents the similarity in word and sentiment polarity distributions between the two corresponding sentences. Then, in Lines 2–12, instead of using existing structure-based community detection algorithm such as [15] and [16], we propose a new algorithm that utilize both structure and context linkage (i.e., informativeness of a review to a sentence) 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). More specifically, we select a number of initial leaders on the graph G, as presented in Section III-B. Then, in Line 4–12, we iteratively discover and update the communities and leaders (Section III-C and Section III-D). After that, a set of informative sentences are extracted from each community and a system summary is generated accordingly.

III. METHODOLOGIES

A. Sentence-Graph Construction

Denote G=(V, E) the sentence-graph constructed from the set of sentences S=$\{s_1, s_2, \ldots, s_n\}$, 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 [17].

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,

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

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 an nonnegative weight w_{ij} as follows,

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

where $\mathcal{N}_k(s_j)$ is the k-nearest neighbors of the sentence s_j according to the similarity measure[1]. From the preliminary

[1]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)

test, we use a grid search to find the best combination for λ and k. The optimal values we found are λ=$\frac{2}{3}$ and k=$\lceil\frac{N}{5}\rceil$, where $N = |\mathcal{R}|$. Therefore, for all the experiments in this paper, we set λ=$\frac{2}{3}$ and k=$\lceil\frac{N}{5}\rceil$.

B. 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

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[18]. The concept of h-node is originated from the the h-index [19] 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 [18].

Another component of our leader initialization aims to reduce redundance 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 2.

C. Community Assignment

Once some leaders are initialized, we can initialize communities by assigning a single community to each leader. 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 [20], 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.

Algorithm 3 presents the method to determine the community membership for a node v. Note that in Algorithm 1 (Lines 6–7), we start calling Algorithm 3 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).

Algorithm 2 Leader Initialization

Input:Graph G=(V, E)
Output: a set of leaders L
L=$\emptyset$
Compute the set of h-nodes $H \in V$
Order H by node degree in descending order
while H is not empty
    pick v from the front of H
    H=$H \setminus \{v\}$, flag=true;
    for each $s \in L$
        if v is a neighbor of s
            flag=false;
    if falg==true
        L=$L \cup \{v\}$
return L

Algorithm 3 Find community

Input: graph G, leaders L, node v, communities $\mathcal{C}$
Output: A community $C \in \mathcal{C}$
let `maxcommon` = 0, C=$\emptyset$
for each $l \in L$
    let `num` = the number of edges between v and community $C(l)$
    if `num>maxcommon`
        `maxcommon=num`
        C=$C(l)$
if C is empty
    mark v as outlier
else
    C=$C \cup \{v\}$
return C

D. Leader Reassignment

We now update the leaders of each community based on the following two intuitions:

Informativeness within a community. As we know, the centrality of nodes in a community measures the relative importance of a node 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 [21] that could be parameter to the algorithm, namely degree, betweenness, closeness and eigenvector centrality measures. We experimented with them all 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 neighbors of the node v from the community and represents to some extent the "popularity" of v in the community. That is,

$$\mathtt{deg}(v, C) = \frac{|\{u|u \in C, (u, v) \in E\}|}{|C| - 1}$$

where $|\{u|u \in C, (u,v) \in E\}|$ denotes the number of nodes in C that are neighbors of node v.

Informativeness within a review. The sentence informativeness within a review is estimated using a mutual reinforcement algorithm. 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.

Combining the above two informativeness measurements, 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,

$$\begin{cases} \varphi(s) = & \varphi(D)\mathtt{deg}(v, C(s)), \\ \varphi(D) = & \frac{1}{|D|}\sum_{s \in D} \varphi(s), \end{cases} \tag{2}$$

where $\mathtt{deg}(v,\ C(s))$ 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. 2, 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,

$$s_l = \arg\max_{s \in C(s)} \varphi(s).$$

IV. Experiments

A. Datasets, Baselines and Evaluation Metrics

The dataset[2] 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). We manually label its aspects and the sentiment polarity towards them on each sentence.

We compare our method, denoted by Leader-based sentence selection $\mathcal{S}_{\mathtt{leader}}$, with three baselines. In order to avoid length-based bias[3], 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 $\mathcal{S}_{\mathtt{leader}}$.

• Aspect-based sentence selection ($\mathcal{S}_{\mathtt{aspect}}$): 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 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 summary reaches $|\mathcal{S}_{\mathtt{leader}}|$.

[2]available at sites.google.com/site/linhongi2r/tools/dataset

[3]A longer summary is more likely to provide better information but is less concise

TABLE I. The size of summary.

	case	laptop	iMac	ipad	protector	kindle
$\lvert\mathcal{S}_{\mathtt{leader}}\rvert$	96	27	44	94	69	234
$\frac{\lvert\mathcal{S}_{\mathtt{leader}}\rvert}{\lvert\mathcal{R}\rvert}$	3.3%	5.7%	7.7%	2.6%	7.2%	1.1%

TABLE II. Aspect coverage ζ comparison.

product	$\mathcal{S}_{\mathtt{leader}}$	$\mathcal{S}_{\mathtt{aspect}}$	$\mathcal{S}_{\mathtt{position}}$	$\mathcal{S}_{\mathtt{rank}}$
case	94%	100%	88%	76%
laptop	90%	85%	65%	86%
iMac	88%	94%	47%	47%
ipad	97%	94%	85%	69%
protector	84%	84%	37%	82%
kindle	94%	97%	88%	91%
average	91%	92%	68%	75%

• Position-based sentence selection ($\mathcal{S}_{\mathtt{position}}$): 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 [22].

• Ranking-based sentence selection ($\mathcal{S}_{\mathtt{rank}}$): After computing the sentence graph, ranking-based sentence selection uses graph-based ranking techniques [23] 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 $\mathcal{S}_{\mathtt{leader}}$.

We evaluate all the approaches in this paper 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

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

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 $\overrightarrow{t} = (t_1, \ldots, t_n)$ 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 $\overrightarrow{t'}$ denotes the aspect-level polarity distribution of a summary $\mathcal{S}$, then its polarity distribution preservation ratio to $\mathcal{R}$ is defined as

$$\eta = \mathtt{corr}(\overrightarrow{t'}, \overrightarrow{t})$$

where $\mathtt{corr}(\cdot)$ 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 $\mathcal{R}$.

B. Quantitative Evaluation

Firstly, we report the number of sentences in summary returned by Leader-based sentence selection $\mathcal{S}_{\mathtt{leader}}$ in Table I. The results illustrated that all of the summaries achieve good compression ratios between 1.1% to 7.7%. Note that we do not report the size of other summaries since they are either equal to or very similar to the size of $\mathcal{S}_{\mathtt{leader}}$.

TABLE III. POLARITY DISTRIBUTION PRESERVATION η COMPARISON.

product	$\mathcal{S}_{\texttt{leader}}$	$\mathcal{S}_{\texttt{aspect}}$	$\mathcal{S}_{\texttt{position}}$	$\mathcal{S}_{\texttt{rank}}$
case	0.926	0.783	0.924	0.844
laptop	0.98	0.61	0.641	0.629
iMac	0.789	0.139	0.238	0.507
ipad	0.97	0.568	0.9	0.663
protector	0.867	0.482	0.4544	0.732
kindle	0.85	0.68	0.76	0.8
average	0.897	0.544	0.653	0.696

Next, we study how the proposed method performs with respect to the aspect coverage ζ. The results are reported in Table II. The baseline $\mathcal{S}_{\texttt{aspect}}$ is supposed to maximize the aspect coverage and on average it could attain 92% coverage. From the results, we can find that the aspect coverage of the proposed method $\mathcal{S}_{\texttt{leader}}$ is close to that of $\mathcal{S}_{\texttt{aspect}}$. On some product domains, the aspect coverage of $\mathcal{S}_{\texttt{leader}}$ is even better than that of $\mathcal{S}_{\texttt{aspect}}$, such as Dell laptop and ipad. Ranking-based method $\mathcal{S}_{\texttt{rank}}$, performs worse than both $\mathcal{S}_{\texttt{aspect}}$ and $\mathcal{S}_{\texttt{leader}}$, but has much better aspect coverage than $\mathcal{S}_{\texttt{position}}$. These results indicate that the proposed method $\mathcal{S}_{\texttt{leader}}$ performs 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 III. As can be seen from the table, our proposed method $\mathcal{S}_{\texttt{leader}}$ can preserve polarity distribution of the original reviews in the aspect level. The Aspect-based sentence selection method $\mathcal{S}_{\texttt{aspect}}$ 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 got skewed. Surprisingly, from the table we find that the Position-based method $\mathcal{S}_{\texttt{position}}$ 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 $\mathcal{S}_{\texttt{position}}$ can obtain reasonable performance in terms of polarity distribution preservation.

V. CONCLUSIONS

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 a deterministic algorithm to find the leaders (informative sentences) and communities (sentences with similar aspects and viewpoints) simultaneously. A set of systematic evaluation as well as quality evaluation verified that the proposed method is able to achieve good performance.

ACKNOWLEDGMENT

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

REFERENCES

[1] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the 10th ACM SIGKDD*. New York, NY, USA: ACM, 2004, pp. 168–177.

[2] Y. Lu, C. Zhai, and N. Sundaresan, "Rated aspect summarization of short comments," in *Proceedings of the 18th WWW*. New York, NY, USA: ACM, 2009, pp. 131–140.

[3] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai, "Topic sentiment mixture: modeling facets and opinions in weblogs," in *Proceedings of the 16th WWW*. New York, NY, USA: ACM, 2007, pp. 171–180.

[4] A.-M. Popescu and O. Etzioni, "Extracting product features and opinions from reviews," in *Proceedings of the HLT and EMNLP*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2005, pp. 339–346.

[5] B. Hu, Z. Song, and M. Ester, "User features and social networks for topic modeling in online social media," in *ASONAM*, 2012, pp. 202–209.

[6] I. Titov and R. T. McDonald, "A joint model of text and aspect ratings for sentiment summarization," in *ACL*, 2008, pp. 308–316.

[7] C. Danescu-Niculescu-Mizil, G. Kossinets, J. M. Kleinberg, and L. Lee, "How opinions are received by online communities: A case study on amazon.com helpfulness votes," in *Proceedings of the 18th WWW*. New York, NY, USA: ACM, 2009, pp. 141–150.

[8] S.-M. Kim, P. Pantel, T. Chklovski, and M. Pennacchiotti, "Automatically assessing review helpfulness," in *Proceedings of EMNLP*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, pp. 423–430.

[9] E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw, "Detecting product review spammers using rating behaviors," in *Proceedings of the 19th CIKM*. New York, NY, USA: ACM, 2010, pp. 939–948.

[10] J. Liu, Y. Cao, C. Y. Lin, Y. Huang, and M. Zhou, "Low-Quality product review detection in opinion summarization," in *Proceedings of the Joint Conference on EMNLP-CoNLL*, 2007, pp. 334–342.

[11] S. Blair-goldensohn, T. Neylon, K. Hannan, G. A. Reis, R. Mcdonald, and J. Reynar, "Building a sentiment summarizer for local service reviews," in *NLP in the Information Explosion Era*, 2008.

[12] K. Ganesan, C. Zhai, and J. Han, "Opinosis: A graph based approach to abstractive summarization of highly redundant opinions," in *Proceedings of the 23rd COLING*, 2010.

[13] K. Lerman, S. Blair-Goldensohn, and R. McDonald, "Sentiment summarization: evaluating and learning user preferences," in *Proceedings of the 12th EACL*. Stroudsburg, PA, USA: ACL, 2009, pp. 514–522.

[14] P. Tsaparas, A. Ntoulas, and E. Terzi, "Selecting a comprehensive set of reviews," in *Proceedings of the 17th ACM SIGKDD*. New York, NY, USA: ACM, 2011, pp. 168–176.

[15] V. D. Blondel, J. loup Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," 2008.

[16] P. J. McSweeney, K. Mehrotra, and J. C. Oh, "A game theoretic framework for community detection," in *ASONAM*, 2012, pp. 227–234.

[17] M. Taboada, C. Anthony, and K. Voll, "Methods for creating semantic orientation dictionaries," in *Proceedings of 5th ICLRE*, Genoa, Italy, May 2006, pp. 427–432.

[18] J. Cheng, Y. Ke, A. W.-C. Fu, J. X. Yu, and L. Zhu, "Finding maximal cliques in massive networks by h*-graph," in *Proceedings of the SIGMOD*. New York, NY, USA: ACM, 2010, pp. 447–458.

[19] J. E. Hirsch, "An index to quantify an individual's scientific research output," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 46, pp. 16 569–16 572, 2005.

[20] Q. Lu and L. Getoor, "Link-based classification," in *Proceedings of the 20th ICML*. AAAI Press, Chicago, IL, USA, 2003, pp. 496–503.

[21] L. C. Freeman, "Centrality in social networks: Conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, 1979.

[22] P. Beineke, T. Hastie, C. Manning, and S. Vaithyanathan, "Exploring sentiment summarization," in *AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, 2004.

[23] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in *EMNLP*, 2004, pp. 404–411.