Graph-Based Opinion Entity Ranking in Customer Reviews

Kunuch Chutmongkolporn, Bundit Manaskasemsak and Arnon Rungsawang
Massive Information & Knowledge Engineering Laboratory
Department of Computer Engineering, Faculty of Engineering
Kasetsart University, Bangkok 10900, Thailand.
kunuch@gmail.com, {un, arnon}@mikelab.net

Abstract—**Online product reviews currently pose large impact on purchasing decision of potential customers. However, the overwhelming number of those reviews hinder people to find useful information and make good decisions on the purchases. In this paper, we propose a graph-based opinion entity ranking framework to mine opinion data from former customers, and rank either entities (products) or aspects (features) in accordant with those opinions. From the customer reviews, we first extract aspects and their sentiment words for each entity. We then represent relationships between reviewers and pairs of entity-aspect by a weighted bipartite graph, and propose an algorithm to compute the ranking scores. Experimental results on a hotel review dataset show higher ranking agreements with those of the human users than ones from tradition frequency-based baseline.**

Keywords—opinion mining, entity ranking, aspect ranking, customer review, bipartite graph

I. Introduction

The advent of e-commerce has facilitated consumers to purchase products or services online. Many online shopping sites provide opportunity and encourage their customers to add reviews or comments of products purchased. Opinions expressed on various aspects of those products, services, as well as shopping experiences, give rise to enormous collections of consumer reviews on the Web. Prior to make a purchase decision, new coming customers commonly seek quality word-of-mouth information from online reviews. On the other hand, product entrepreneurs or service providers utilize the same reviews as an important resource in their product development, service improvement, marketing plan, as well as customer relationship management.

However, the sheer volume of available online reviews gives burden to consumers to distinguish the most helpful information, and to evaluate the true underlying quality of the interested product based on those reviews [6]. In the same way, product manufacturers need to recognize reviews that affect sales most. To overcome this problem, researchers commonly first identify the important product aspects from reviews and determine consumers' opinions on those aspects via sentiment classifier. They then develop the ranking algorithms to rank the aspects of an entity [9], the entities according to the influent opinion aspects given by the former customers [10], [2], or the reviews according to their expected helpfulness or expected affect on sales [3], [8].

To help future customers make better purchase decision, in this paper, we propose a new graph-based opinion entity ranking framework to mine opinion data from customer reviews, and to rank either the entities or the aspects of entities according to those opinions based on graph topology built from entities, aspects, and reviewers. From a hotel review dataset, we first extract aspects (i.e., features) and their sentiment words from each hotel review. We then identify the opinion orientation of those aspects and then create a weighted bipartite graph in which edges connect from a reviewer node to several pairs of entity-aspect nodes. A ranking algorithm running on this graph is then proposed. From experiments, we found that our proposed framework provides higher agreement in rankings with a user study than those of the traditional frequency-based baseline adapted from [5].

To what follows, we briefly mention related work, provide detail of our ranking framework, and give evaluation results. We finally conclude the paper in the last section.

II. Related Work

As the e-commerce activities produce an overwhelming amount of comments and reviews regarding products and shopping experiences, it is impractical for people to find useful reviews and make decisions about purchases or recommendations from existing customers. Traditional recommendation systems, such as Amazon.com and Tripadvisor.com, rely on their own built-in reviews and rating networks. However, different customers put different emphasis on different aspects (e.g., prefer good breakfast service to spacious room in a hotel), those numeric ratings do not help reduce prospect customer's effort on reading though such numerous reviews. Thus, task of developing a computational model to help newcomers to digest and exploit existing opinions is an interesting research challenge, and very helpful.

To tackle this challenge, early work by Hu and Liu [5] has been concentrated on identifying important sentiment aspects that best describe the product from reviews, and then classifying those reviews into positive and negative opinions. Ghose and Ipeirotis [3] apply econometric analysis and text mining technique to devise two ranking algorithms: a consumer-oriented ranking base that ranks reviews according to their expected helpfulness, and a manufacturer-oriented base that ranks reviews according to their expected effect on sales. Krestel and Dokoohaki [8] propose to summarize a product with the top-k ranked reviews. They exploit both latent topics within reviews and the assigned start rating numbers as an indicator of the polarity expressed towards the product.

978-1-4673-6820-9/15/$31.00 ©2015 IEEE

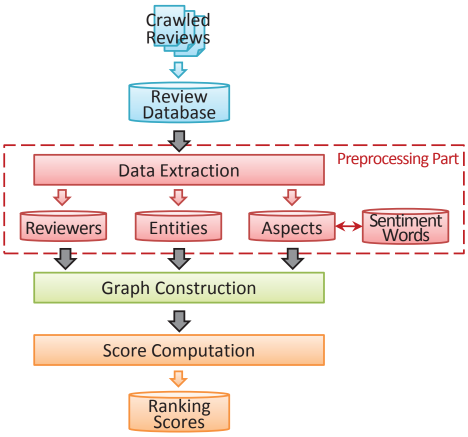

Fig. 1. An opinion entity ranking framework.

To rank the aspects of a product, Yu *et al.* [9] first identify important product aspects from reviews using a shallow dependency parser, and then determine opinions on those aspects via a sentiment classifier. They then propose an aspect ranking algorithm to rank the important aspects by considering both aspect frequency and influence of opinions given to each aspect on their overall opinions. Zhang *et al.* [10] incorporate various product review factors, such as content related to product quality and product durability, time of the review, historically older positive customer reviews, to their product ranking model. They report the closely related product ranking results to those of the sale ranks from Amazon.com. Recently, Ganesan and Zhai [2] propose an entity ranking model in which all candidate entities will be ranked based on how well opinions on those entities match the users' preferences.

III. Graph-Based Opinion Entity Ranking

In this section, we present a novel framework, called a *reviewer-entity* model, for the problem of an opinion entity ranking. Consider a collection of customer reviews, they are needed to be first characterized into a set of *reviewers* who express opinions on *entities* (i.e., products or services) associated with some *aspects* (i.e., features). We here adopt an extraction process introduced in [5] to detect aspects and *sentiment words* for an entity. The orientation (i.e., either positive, negative, or neutral) of an aspect is assigned by examining its corresponding sentiment word through the SentiWordNet[1] library. Afterwards, a graph representation is constructed based on the relationships between reviewers and entity-aspect pairs. Finally, a ranking score is calculated for an individual aspect, entity, or reviewer. Fig. 1 depicts an overview of our framework.

A. Reviewer-Entity Model

In the reviewer-entity model, we interpret the problem as a weighted bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, where $\mathcal{V}$, $\mathcal{E}$, and $\mathcal{W}$ represent vertices, multiple directed edges, and weights, respectively. Given a set of l reviewers denoted as $R = \{r_1, r_2, \ldots, r_l\}$, a set of m entities denoted as $E = \{e_1, e_2, \ldots, e_m\}$, and a set of n possible aspects denoted as $A = \{a_1, a_2, \ldots, a_n\}$, we aim to first construct the graph whose vertices are decomposed by two types: reviewers and pairs of entities associated with their aspects, i.e., $\mathcal{V} = R \cup (E \times A)$. A directed edge $\varepsilon(r_i, \langle e_j, a_k\rangle)$ going from a reviewer r_i to an entity's aspect $\langle e_j, a_k\rangle$ will be defined if there exists a review that the former expresses his/her opinion on the latter. Fig. 2 illustrates an example reviewer-entity graph.

[1]http://sentiwordnet.isti.cnr.it

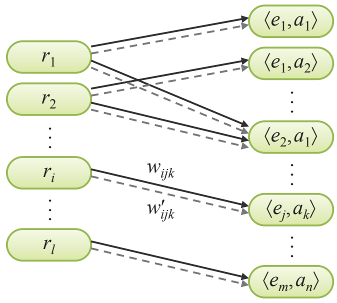

Fig. 2. An example graph of the reviewer-entity model.

As it can be seen from the example graph, vertices on the left side (i.e., reviewers) connect to ones on the right side (i.e., entity-aspect pairs) via multiple directed edges. A weight w_{ijk} defined on a solid edge refers to a normalized sentiment score that the reviewer r_i expresses his/her opinion on the entity e_j based on its aspect a_k. In this work, the sentiment score is simply obtained from the SentiWordNet library by examining the sentiment word corresponding to a_k in the review. Let s_{ijk} be the sentiment score from r_i to $\langle e_j, a_k\rangle$. Then, the sentiment weight w_{ijk} is formulated as:

$$w_{ijk} = \frac{s_{ijk}}{\sum_{\forall y,z:\varepsilon(r_i,\langle e_y,a_z\rangle)\in\mathcal{E}} |s_{iyz}|}. \tag{1}$$

Since the SentiWordNet returns a floating-point value in range $[-1, 1]$, meaning strongly negative through strongly positive expression, this sentiment weight then has the same range.

For a weight w'_{ijk} assigned to a dashed edge, we define it as how the orientation expressed by the reviewer r_i on the entity e_j and its aspect a_k is agreed with those of the others. To measure the degree of agreement, we need to first divide the continuous range of sentiment score into a number—five in our work—of equal intervals. Then, s_{ijk} is discretized into one of them. We here use the Fleiss' kappa [1] to assess the agreement of multiple reviewers. Let $\Delta\kappa_{ijk} = \kappa_{ijk}^{(after)} - \kappa_{ijk}^{(before)}$ be the difference Fleiss' kappa between after and before r_i contributes an expression on $\langle e_j, a_k\rangle$. Then, the normalized agreement weight w'_{ijk} is formulated as:

$$w'_{ijk} = \frac{1/(1+e^{-\Delta\kappa_{ijk}})}{\sum_{\forall x:\varepsilon(r_x,\langle e_j,a_k\rangle)\in\mathcal{E}} 1/(1+e^{-\Delta\kappa_{xjk}})}. \tag{2}$$

The agreement weight is a floating-point value in range $[0, 1]$, meaning strong disagreement through strong agreement with the others.

To compute ranking scores given the example graph $\mathcal{G}$ in Fig. 2, our algorithm contributes a key concept inspired from HITS [7]. The algorithm calculates two types of scores: a reviewer score and an entity-aspect pair score. Both are defined recursively in terms of each other. That is, the entity-aspect pair score $\phi(e_j, a_k)$ given for a vertex $\langle e_j, a_k\rangle$ in the graph is the

function $Reviewer\text{-}Entity(\mathcal{G})$
1: $t = 0$
2: $\phi^{(t)} = \mathbf{1}$
3: $\varphi^{(t)} = \mathbf{1}$
4: **repeat**
5: $\quad t = t + 1$
6: $\quad$ **foreach** $\langle e_j, a_k \rangle \in \mathcal{V}$ **do**
7: $\quad\quad \phi^{(t)}[\langle e_j, a_k \rangle] = \sum_{\forall i:\varepsilon(r_i,\langle e_j,a_k\rangle)\in\mathcal{E}} w_{ijk} \cdot \varphi^{(t-1)}[r_i]$
8: $\quad$ **foreach** $r_i \in \mathcal{V}$ **do**
9: $\quad\quad \varphi^{(t)}[r_i] = \sum_{\forall j,k:\varepsilon(r_i,\langle e_j,a_k\rangle)\in\mathcal{E}} w'_{ijk} \cdot |\phi^{(t-1)}[\langle e_j, a_k \rangle]|$
10: $\quad \phi^{(t)} = \phi^{(t)} / \|\phi^{(t)}\|_1$
11: $\quad \varphi^{(t)} = \varphi^{(t)} / \|\varphi^{(t)}\|_1$
12: **until** $\|\phi^{(t)} - \phi^{(t-1)}\|_1 + \|\varphi^{(t)} - \varphi^{(t-1)}\|_1 < \epsilon$
13: **return** $\phi^{(t)}, \varphi^{(t)}$

Fig. 3. The reviewer-entity algorithm.

sum of all reviewers' scores who express opinions on it:

$$\phi(e_j, a_k) = \sum_{\forall i:\varepsilon(r_i,\langle e_j,a_k\rangle)\in\mathcal{E}} w_{ijk} \cdot \varphi(r_i), \tag{3}$$

while the reviewer score $\varphi(r_i)$ of a reviewer r_i is the sum of scores of all entity-aspect pairs that have been mentioned to:

$$\varphi(r_i) = \sum_{\forall j,k:\varepsilon(r_i,\langle e_j,a_k\rangle)\in\mathcal{E}} w'_{ijk} \cdot |\phi(e_j, a_k)|. \tag{4}$$

More precisely, in (3), a high positive (or low negative) entity-aspect score comes from many high quality reviewers who all agree on that being indeed good (or bad). While, in (4), a high reviewer score comes many good and bad entity-aspect pairs which are agreed with the others.

In practice, those reviewer and entity-aspect pair scores can be computed using an iterative procedure. Fig. 3 illustrates the algorithm. First, at lines 2–3, the entire scores are all initialized uniformly. Next, the iterative computation is started. Based on scores given from the previous iteration, at lines 6–9, the new entity-aspect pair and the new reviewer scores are calculated by invoking (3) and (4), respectively. Afterwards, at lines 10–11, these scores are needed to be normalized, in order to guarantee that all entity-aspect pair scores and reviewer ones will have values in range $[-1, 1]$ and $[0, 1]$, respectively. These iterative steps are repeated until convergence occurs, which is defined as the total change compared to the previous iteration being less than a pre-defined threshold ϵ (e.g., 0.00001 in our experiments).

B. Ranking System

Considering scores returned from the reviewer-entity algorithm, we now present four paradigms to employ them in our ranking system.

1) Entity-specific aspect ranking: Sometimes, users would expect to know which aspect is either satisfied (i.e., positive) or unsatisfied (i.e., negative) by most former customers for an exact entity, "what is an advantage of this phone?" or "how about is this hotel?", for instance. In this paradigm, the system here aims to produce an entity-specific aspect ranking to those users. Therefore, the ranking Γ_{A_j} is defined as a list of all aspects a_k belonging to a given entity e_j in which their $\phi(e_j, a_k)$ scores are ordered decreasingly.

2) Aspect-specific entity ranking: When users would like to compare which entity is better or worse for a specific aspect, "which phone is better in case of battery life?" or "which hotel has a good swimming pool?", for instance, an aspect-specific entity ranking is required to those users. Therefore, the ranking Γ_{E_k} is defined as a list of all entities e_j having a given aspect a_k in which their $\phi(e_j, a_k)$ scores are ordered decreasingly.

3) Entity ranking: In general, when users would expect to know which entity is the best (on average aspects) over the whole ones, the system then needs to produce a ranking at the entity level. Thus, for an individual entity, the score is first calculated by averaging all scores of its corresponding aspects. We let $\phi(e_j)$ be the score of an entity e_j, defined as:

$$\phi(e_j) = \sum_{\forall k:\langle e_j,a_k\rangle\in\mathcal{V}} \phi(e_j, a_k).$$

Then, the ranking Γ_E is given by ordering all these $\phi(e_j)$ scores decreasingly.

4) Reviewer ranking: Our system can also specify which reviewer has higher or lower review quality (i.e., providing helpful or useless reviews), based on a reviewer ranking Γ_R. The ranking is simply given by ordering all reviewer scores, $\varphi(r_i)$, decreasingly. Moreover, we believe that these reviewer scores are the important information that helps further detect untruthful opinions and spammers, but beyond our work here.

IV. Experiments

A. Dataset and Setup

We archive a hotel review dataset from Tripadvisor.com. After selecting hotels reviewed by more than 10 former customers (i.e., reviewers), we finally obtain around 37K reviews from 296 hotels. Then, we follow the processes suggested in [5] to extract hotels (i.e., entities), hotel features (i.e., aspects), associated opinion words, and reviewers as the input of our reviewer-entity model.

We here only focus on the first three ranking paradigms mentioned in Sect. III-B. To evaluate the quality of those rankings with respect to human users' notion of preference, we have conducted a user study involving 11 people in Computer Engineering department of Kasetsart University. In the study, we samples 20 hotels located in Samui island[2] containing 745 reviews. The users are asked to manually extract hotel features and associated opinion words from each review, and freely assign a discrete sentiment score to each hotel feature from -2 (strongly negative opinion) to 2 (strongly positive one). Afterwards, we aggregate all users' assessments (in accordance with each ranking paradigm) and sort them from highest to lowest to create a user ranking for comparisons.

Experimental results are measured in term of the similarity between the ranking of our proposed algorithm and the human users' one within any top-k items. Moreover, we compare the ranking performance of our algorithm (*RE*) with that of the traditional frequency-based baseline (*Freq*) adapted from [5]. This baseline ranks items in decreasing order of different amount of positive opinions minus negative ones.

[2] http://tripadvisor.com/Hotels-g293918-Ko_Samui_Surat_thani_Province-Hotels.html

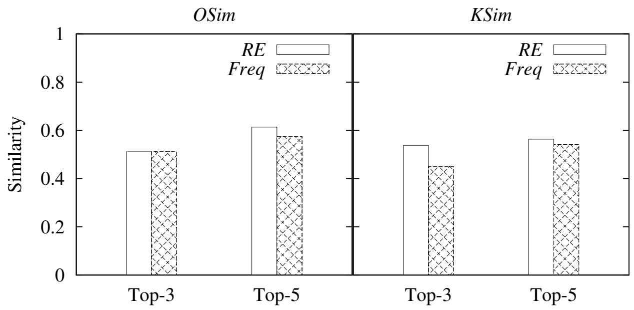

Fig. 4. Average similarities at top-3 and 5 of hotel-specific feature rankings.

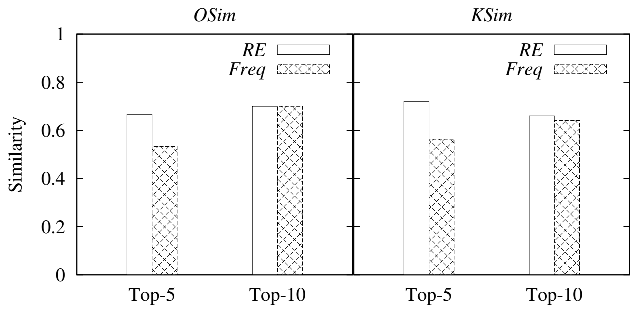

Fig. 5. Average similarities at top-5 and 10 of feature-specific hotel rankings

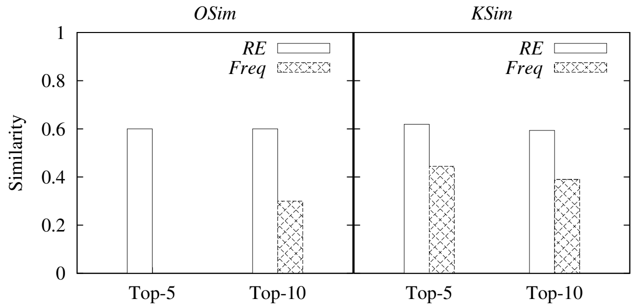

Fig. 6. Average similarities at top-5 and 10 of hotel rankings

B. Evaluation Metrics

We use $OSim$ and $KSim$ metrics, proposed in [4], to measure the similarity between any two ranked lists. $OSim_k(\Gamma_1, \Gamma_2)$ determines the degree of overlap between the top-k items of the two rankings Γ_1 and Γ_2:

$$OSim_k(\Gamma_1, \Gamma_2) = \frac{|\tau_1 \cap \tau_2|}{k}, \tag{5}$$

where τ_1 and τ_2 are the lists of top-k items contained in Γ_1 and Γ_2, respectively.

$KSim_k(\Gamma_1, \Gamma_2)$ determines the degree of agreement in which pairwise distinct items u and v within top-k have the same relative order in both rankings Γ_1 and Γ_2. Consider two lists τ_1 and τ_2 of top-k items. Let $\mathcal{U}$ be a union of items contained in both lists, and define τ_1' as the extension of τ_1 to add the elements $\mathcal{U} - \tau_1$ after all the items in τ_1. Similarly, τ_2' is also defined as the extension of τ_2. $KSim$ is defined as:

$$KSim_k(\Gamma_1, \Gamma_2) = \frac{|\{(u,v)\colon \begin{array}{c}\tau_1', \tau_2' \text{ agree on order} \\ \text{of } (u,v) \text{ and } u \neq v\end{array} \}|}{|\mathcal{U}| \times (|\mathcal{U}| - 1)}. \tag{6}$$

Note that the higher $OSim$ and $KSim$ values indicate more similar ranking between both lists.

C. Results

Figs. 4–6 illustrate average similarities of feature ranking for an individual hotel, hotel ranking for an individual feature, and overall hotel ranking, respectively. Note that since there exist a few features mentioned to for each hotel, in Fig. 4 we then report only similarities at top-3 and 5. As it can be seen, the *RE* algorithm gives higher $OSim$ and $KSim$ values than those of *Freq* in almost the cases, indicating that *RE* can identify hotel's features (or hotels) most relevant to human users' preference within the top few number of rankings.

V. Conclusion

In this paper, we handle the opinion entity ranking task in customer reviews using a weighted bipartite graph built from the relationship between opinions given by reviewers on aspects of an entity. An algorithm running on that graph which can produce the scores reflecting the aspect, entity, as well as reviewer qualities, has been proposed. We also conduct a user study for performance evaluation, and found that rankings produced from our framework have higher degree of agreement with those of the user study than those of the traditional baseline reported in the literature. In our future work, we will investigate the performance on quality of the reviewers and do an extensive experiment on other review dataset.

References

[1] J. L. Fleiss, "Measuring nominal scale agreement among many raters," *Psychological Bulletin*, vol. 76, no. 5, pp. 378–382, 1971.

[2] K. Ganesan and C. Zhai, "Opinion-based entity ranking," *Information Retrieval*, vol. 15, no. 2, pp. 116–150, 2012.

[3] A. Ghose and P. G. Ipeirotis, "Designing novel review ranking systems: Predicting the usefulness and impact of reviews," in *Proceedings of the 9th International Conference on Electronic Commerce*, 2007, pp. 303–310.

[4] T. H. Haveliwala, "Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, pp. 784–796, 2003.

[5] M. Hu and B. Liu, "Mining opinion features in customer reviews," in *Proceedings of the 19th National Conference on Artificial Intelligence*, 2004, pp. 755–760.

[6] N. Hu, P. A. Pavlou, and J. Zhang, "Can online reviews reveal a product's true quality?: Empirical findings and analytical modeling of online word-of-mouth communication," in *Proceedings of the 7th International Conference on Electronic Commerce*, 2006, pp. 324–330.

[7] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM*, vol. 46, no. 5, pp. 604–632, 1999.

[8] R. Krestel and N. Dokoohaki, "Diversifying product review rankings: Getting the full picture," in *Proceedings of the IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, 2011, pp. 138–145.

[9] J. Yu, Z.-J. Zha, M. Wang, and T.-S. Chua, "Aspect ranking: Identifying important product aspects from online consumer reviews," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, pp. 1496–1505.

[10] K. Zhang, Y. Cheng, W.-K. Liao, and A. Choudhary, "Mining millions of reviews: A technique to rank products based on importance of reviews," in *Proceedings of the 13th International Conference on Electronic Commerce*, 2011.