Reviewing the Reviewers: Characterizing Biases and Competencies using Socially Meaningful Attributes

Sihem Amer Yahia and **Andrei Z. Broder** and **Alban Galland**
Yahoo! Research
{sihem, broder, alban}@yahoo-inc.com

Abstract

We propose augmenting collaborative reviewing systems with an automatic annotation capability that helps users interpret reviews. Given an item and its review by a certain author, our approach is to find a reference set of similar items that is both easy to describe and meaningful to users. Depending on the number of available same-author reviews of items in the reference set, an annotation produced by our system may consist of similar items that the author has reviewed, the rank of the reviewed item among items in this set, a comparison of the author's scores to averages, and other similar information that indicate the biases and competencies of the reviewer.

We validate our approach in the context of movie reviews and describe an algorithm that, for example, presented with a review of a Woody Allen comedy, is able to derive annotations of the form: "This reviewer rates this movie better than 4 out of 6 other Woody Allen comedies that he rated" or "This is the only Woody Allen comedy among the 29 movies rated by this reviewer" or "This reviewer rated 85 comedies. He likes this movie more than 60% of them. He likes comedies less than the average reviewer."

Introduction

The advent of "Web 2.0", that is, the evolution of the Web from a technology platform to a social milieu, has been accompanied by an explosion in the number of *collaborative reviewing systems*. These systems grew out of *collaborative ranking systems* with the additional ability to enter textual reviews complementing the numerical evaluation of an item.

Although a few reviewing systems existed before the Web (e.g. *Zagat* surveys, *Consumer Reports*), the input to the system was cumbersome (mail forms, questionnaires, phone surveys) and the results were typically edited by professional editors. In contrast, there are numerous online reviewing systems that help organize and share socially produced information in support of various web-mediated activities: auctions (`www.ebay.com`), choosing a movie (`movies.yahoo.com`), choosing local "brick-and-mortar" services and shops (`www.yelp.com`), renting a DVD (`www.netflix.com`), buying a book (`www.amazon.com`), or booking a hotel (`www.tripadvisor.com`). There is at least one vendor of a platform for collaborative reviewing for e-commerce, `powerreviews.com` that boasts of over one thousands customers, including well-known U.S. retailers such as Staples and Walgreens.

All these systems include a collection of items of interest (books, movies, etc) and a collection of ratings (numerical values), some of them accompanied by written reviews provided by the users of the system. The number of ratings and reviews could be quite large. For example, on Yahoo! Movies, a recent relatively obscure movie, "La Vie en Rose" released on June 8, 2007, had by August 7, 2007 a total of 573 ratings including 89 written reviews. A popular movie, "Ratatouille", has accumulated 1743 user reviews and 21004 user ratings in just 6 weeks. Hence the aggregate numerical values convey only coarse information, and do not capture the rich information available in reviews. It is however possible to use text analysis to aggregate reviews *themselves*, e.g. (Popescu & Etzioni 2005). While informative, such aggregation is not widely deployed.

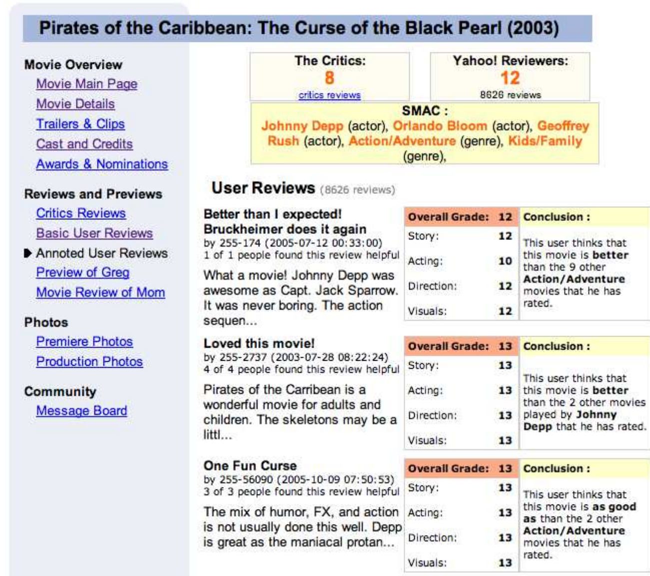
Some systems appeal again to "human computing" by allowing users to *vote* on the usefulness of a review (e.g., "6 of 11 people found this review helpful" in Amazon) or rate a review (e.g., as "useful", "funny", or "cool" in Yelp). A different approach is to enable readers to scrutinize the reviewer (e.g., "top reviewers" in Amazon). However, it has been shown that reviews by top reviewers do not have any extra impact (Chen, Dhanasobhon, & Smith 2007).

We are interested in the building of collaborative reviewing systems, in particular, providing a *context* to help users *interpret* a given review by a given author. Of course, the common practice of allowing readers to access all the reviews of a given author provides an exhaustive context, but it also requires an inordinate amount of effort by users. In contrast, we provide a customized background. Depending on the number of relevant reviews by the same reviewer, this background may consist of the set of similar items that the reviewer has reviewed, a comparison of the reviewer's scores to average scores, and similar information that indicates the biases and competencies of the reviewer.

A prototype movie reviews interpretation system

To illustrate our ideas we present a system based on Yahoo! movies (`movies.yahoo.com`) where each movie is described by a set of attributes: title, genre, directors, actors, etc and a set of reviews and scores. A review is written by

(a) Current Movie Display



(b) Annotated Movie Reviews

Figure 1: Movie Reviews and their Annotations

one reviewer and each reviewer is allowed to enter at most one review per movie. Figure 1(a) shows the current presentation of the movie "Pirates of the Caribbean" in this system.

To help the interpretation of reviews, an extension to this system, presented in Figure 1(b), provides, for each review, a description of the relevant experience and preferences of the reviewer. This requires selecting a collection of similar movies that serves as a background *reference set*, which, however, might be different for different reviews. In our example, the first review annotation uses Action/Adventure movies as a reference set while the second one uses movies starring Johnny Depp.

Choosing the reference set as a review interpretation context, is not a simple matter: an optimal set must be *socially meaningful* and *easy to describe in words*. It must be *large enough* that it provides a context, but *not so generic* that the information is diluted. For instance, the annotation "This reviewer rates this movie better than 51 out of 85 other comedy movies that he has rated" for a Woody Allen comedy, carries less information than "This reviewer rates this movie better than 6 out of 7 Woody Allen Comedies that he has rated". Finally, while more than one reference set could be acceptable, space considerations require to choose one *optimal set* to display.

Finding SMACs

Formalism

Let $\mathcal{I}$ be the set of items subject to review in our system, and assume that every item $i \in \mathcal{I}$ has an associated set of (attribute, value) pairs. For example, a movie has associated attributes such as title, director, actors, etc., and a restaurant has associated attributes such as name, location, chefs, types of cuisine, etc. We also define the set of users $\mathcal{U}$ who are the readers and potential authors of reviews in the system.

Definition 1 [Item Collection (IC)] *An Item Collection is a subset of $\mathcal{I}$.* ∎

Definition 2 [Attribute Collection (AC)] *An Attribute Collection is a set of pairs (att_i, $value_i$) that defines an IC consisting of all the items in $\mathcal{I}$ that have the value $value_i$ for each attribute att_i in AC.* ∎

For example, the AC {(director, Spielberg), (actor, Jeff Goldblum)} defines the IC consisting of all movies directed by Spielberg and starring Jeff Goldblum. The AC {(cuisine, French), (location, New York City), (price, moderate)} defines the (possibly empty) collection of all moderately priced French restaurants in New York City.

We say that AC_1 is *included* in AC_2 iff all the (att_i,$value_i$) pairs of AC_1 are in AC_2. We also say that AC_2 is more *specific* than AC_1 since $\text{IC}_2 \subseteq \text{IC}_1$.

The *coverage* of an AC is simply the size of the corresponding IC, and is similar to the notion of support in association rules (Agrawal & Srikant 1994). The *social meaningfulness* (SM) of an AC is a score that reflects whether a significant number of users would view the corresponding IC as a meaningful classification for the purpose of rating items in $\mathcal{I}$. As we discuss later, this score can be based on the amount of *reviewing attention* received by the corresponding IC. The formula for these scores may depend on the AC *type*, which is the set of attributes in the AC, but not their values. Note that coverage does not guarantee social meaningfulness. For instance, the attribute collection AC $=$ {(origin,France), (length, [90-100])} corresponding to "French movies of length between 90 to 100 minutes" although perfectly defined, is unlikely to be meaningful for the purpose of rating movies.

Definition 3 [Socially Meaningful Attribute Collection (SMAC)] *A Socially Meaningful Attribute Collection (SMAC) is an AC whose Coverage and SM scores exceed domain-dependent thresholds.* ∎

Generation algorithm

Algorithm 1 generates the sets of SMACs. It admits a list of pairs ($movie$,$attval$) where $attval$ contains the value of one of the objective attributes of $movie$. This algorithm builds SMAC$_k$, which are the sets of SMACs with exactly k attribute-value pairs. In line 1, SMAC$_1$ is initialized by the `initializeSMACs` function, who keeps all the attval pairs which are a SMAC. To do so, it uses a boolean function `isSMAC` which given the corresponding SMIC verifies the coverage and SM conditions. Their `left-parent` are fixed to a dummy root. In lines 3-6, SMAC$_{k+1}$ is recursively built using SMAC$_k$ with the function `generateSMACs` described in Algorithm 2. The algorithm returns all the SMAC$_k$ which have been constructed.

Algorithm 1 Generation of the SMACs

Require: L : list of pairs ($movie$,$attval$).
1: SMAC$_1$ = initializeSMACs(L);
2: $k = 1$;
3: **while** SMAC$_k \neq \emptyset$ **do**
4: SMAC$_{k+1}$ = generateSMACs(SMAC$_k$);
5: k++;
6: **end while**
7: return SMAC$_1$,...,SMAC$_{k-1}$

The `generateSMACs` function is described in Algorithm 2. It scans all the pairs of SMACs with the same `left-parent` and tries to build a new SMAC as a union of their attributes. We suppose that we have a total order on the attval pairs, for example the order of their id in the database, and we use this order to define the function `isListSmaller` which compares the lists of attribute values of two SMACs in a lexicographic way. Line 3 tests if the union is a SMAC using the function `isSMAC`. The important point is that the elements of the new potential SMAC are exactly the intersection of the elements of the two parents, which is similar to the stability through intersection of association rules. Line 5 computes `attvals` of the new SMAC. Line 6 computes its `movies`. Line 7 sets its `left-parent`, which is the smallest SMAC used to build it. The algorithm returns the set of SMACs built.

Coverage and Social Meaningfulness in our test set

As a test set for our experiments, we used a subset of the Yahoo! movies database containing 762964 reviews, 18717 movies (41 reviews/movies) and 436 495 users (1.7 reviews/users).

Coverage threshold Figure 2 shows the distribution of the number of ACs built using actors (e.g., one of the ACs is {(actor, Johnny Depp)}). For example, there are 1308 actor ACs which identify at least 5 movies. Based on this distribution, we set the coverage threshold to 5 meaning that the AC

Algorithm 2 generateSMACs : Generation of SMAC$_{k+1}$ from SMAC$_k$

Require: SMAC$_k$
1: SMAC$_{k+1} = \emptyset$
2: **for each** ($smac1$, $smac2$) with left-parent($smac1$)==left-parent($smac2$) and isListSmaller($smac1$,$smac2$) **do**
3: **if** isSMAC(movies($smac1$) ∩ movies($smac2$)) **then**
4: $smac$ = new SMAC;
5: attvals($smac$) = attvals($smac1$) ∪ attvals($smac2$);
6: movies($smac$) = movies($smac1$) ∩ movies($smac2$);
7: left-parent($smac$) = $smac1$;
8: add $smac$ to SMAC$_{k+1}$
9: **end if**
10: **end for**
11: return SMAC$_{k+1}$

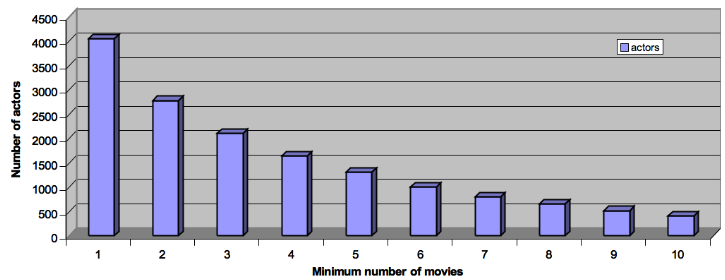


Figure 2: Distribution of actor ACs

defined by the pair (actor, actor_name) is a SMAC candidate if it identifies strictly more than 5 movies.

SM threshold We report the results of our experiments to determine the right SM score definition and threshold for our ACs. There are many possibilities. We experimented with the following criteria to characterize each AC and its corresponding IC:

1. *The total number of movies* in the collection, $|\text{IC}|$;
2. *The total number of reviews* in the IC, denoted $|\mathcal{R}(\mathcal{U}, \text{IC})|$, where $\mathcal{U}$ is the set of all users in the database;
3. *The number of internal co-reviews* that is defined as the number of pairs of movie reviews where both movies belong to the IC and that were reviewed by the same reviewer. We define this set as follows: $|(r_1, r_2) \in \mathcal{R}(\mathcal{U}, \text{IC})^2 \mid \mathit{reviewer}(r_1) = \mathit{reviewer}(r_2)|$, where $\mathit{reviewer}(r)$ is the author of review r.
4. *The number of total co-reviews* that corresponds to the number of pairs of movie reviews where at least one movie belongs to the IC and which were reviewed by the same reviewer. We define this set as follows: $|(r_1, r_2) \mid r_1 \in \mathcal{R}(\mathcal{U}, \text{IC}) \wedge r_2 \in \mathcal{R} \wedge \mathit{reviewer}(r_1) = \mathit{reviewer}(r_2)|$, where $\mathcal{R}$ defines the set of reviews in the whole database.

The last two criteria are similar to the notion of cohesiveness of a cluster (Banerjee, Basu, & Merugu 2007) An alternative to our approach is to use arbitrary item-item clustering methods (see e.g. (Sarwar *et al.* 2001)) and then try to automatically label each cluster with a “socially meaningful

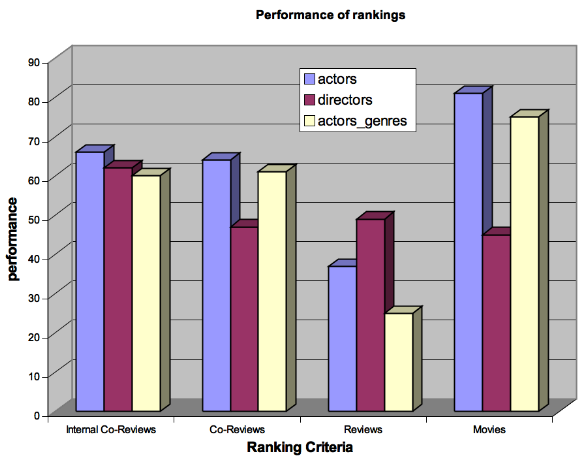


Figure 3: Performance of SM Criteria

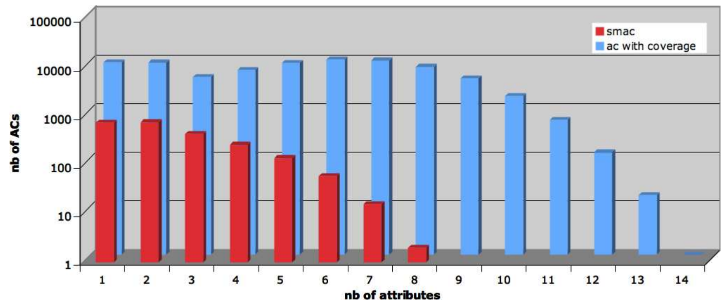


Figure 4: AC-SMAC Distribution

name". Unfortunately the last step is a notoriously difficult problem.

We ran additional experiments to determine the SM function. We used the actor ACs identified previously and ranked 10 distinct actor values by each one of the criteria described above. Table 1 shows the different rankings obtained. We asked 12 users to identify if they know a movie for each actor. The first column of the table contains actor names ranked by the total number of people who recognized them. Figure 3 shows the aggregated size of the intersection between each each user's list of movies and the list obtained for each criterion. The figure contains this information for actor, director and actor-genre ACs.

The assessments show that the number of reviews is not the best SM criterion thereby validating the fact that popularity of movies in an AC doesn't mean that the AC is socially meaningful. It also shows that the number of movies in an AC is the best criterion for actor ACs, but not for director ACs. This argues for an attributes-specific criterion that can only be identified by conducting experiments similar to the ones we discuss.

The best criterion overall seems to be the number of internal co-reviews in the AC. This result justifies that the relationship between reviewers and movies provides more information than popularity (number of reviews in the AC) alone.

Figure 4 shows the number of potential SMACs using the number of internal co-reviews criterion. We note that AC coverage alone is very selective compared to the total number of potential ACs. For example, the number of (director, genre) combinations in the database is 9554 and there are only 19 combinations that satisfy the AC coverage (defined as the directors that made at least 5 movies with enough reviews in the genre). By setting the SM value (number of internal co-reviews) to 50, the number of ACs is further reduced to 4 SMACs.

These observations are explained by power-law distributions in the data (e.g., a large number of movies have a very small number of reviews making coverage very selective and a large number of users have only written a small number of reviews which reduces the number of co-reviews by the same reviewer.)

Interpreting reviews in the context of a SMAC

We discuss how SMACs can be used to interpret reviews in context and describe an algorithm that is used to derive those interpretations.

We are given a SMAC, its collection of items, SMIC, and a set of users $\mathcal{U}_1 \subseteq \mathcal{U}$, we define set of users in $\mathcal{U}_1$ who reviewed an item i as $\mathcal{U}(i) = \{u \in \mathcal{U}_1 \mid \exists r \in \mathcal{R}(i), u = reviewer(r)\}$, where $\mathcal{R}(i)$ denotes the set of reviews associated with i.

Each item belongs to more than one SMAC and each review has potentially a *different interpretation* per SMAC. However, given an item i and a SMAC s, the ability to interpret a review of a given reviewer $u \in \mathcal{U}(i)$ depends on the number of reviews that u has written in the SMIC corresponding to s. Therefore, the identification of *optimal context* to interpret a review is a dynamic process that involves to reason about an individual item and reviewer. We defer this discussion later where we present our review interpretation algorithm and focus first on identifying four classes of reviewers that are defined by the amount of data that is available to interpret a given review of that reviewer in the context defined by s.

1. $\mathcal{U}_{one}(i) = \{u \in \mathcal{U}(i), |R(u, \mathcal{I})| = 1\}$. This set identifies the reviewers who have written only one review in the whole database, namely the one for i. We propose to generate "This is the only review by u in the system". This information, although straightforward, provides additional context to the reader, and presumably reduces reliance on that reviewer's opinion. $\mathcal{U}_{one}$ could also be defined using an arbitrary threshold (instead of 1) in which case, the review interpretation would be of the form "This is one of the only <N reviews> by u in the system" where <N reviews> links to the set of reviews. This type of annotation is fairly similar to existing systems.

2. $\mathcal{U}_{few}(i) = \{u \in \mathcal{U}(i), |R(u, \text{IC})| = 1\}$. This set identifies the reviewers who have only written one review in the SMIC defined by s. We propose to generate "This is the only review by u of a AC item out of his <N reviews>". For instance, "This is the only review by u of a Woody Allen Comedy out of his <20 reviews> in the system".

User Assessments	Number of Movies	Number of Reviews	Number of Co-reviews	Number of Internal Co-reviews
Johnny Depp	Michel Caine	Orlando Bloom	Johnny Depp	Johnny Depp
Robert de Niro	Martin Sheen	Johnny Depp	John Cleese	John Cleese
Bruce Willis	Robert de Niro	Christopher Lee	Orlando Bloom	Orlando Bloom
Eddie Murphy	James Earl Jones	Monica Bellucci	Christopher Lee	Christopher Lee
Cameron Diaz	Whoopi Goldberg	John Cleese	Kirsten Dunst	Samuel L. Jackson

Table 1: Top actors by criterion

Note that *few reviews* can be as significant as *many reviews*, because it suggests a lack of experience with the particular IC. Again the threshold could be higher than 1.

3. $\mathcal{U}_{many}(i) = \{u \in \mathcal{U}(i), 1 < |R(u, \text{IC})| < thresh\}$. This case identifies reviewers who have written more than one review for items in the SMIC defined by s. In this case we have enough other reviews by u that we can present his review in the context of his other reviews: "u rates this item better/worse than K out of the $<|R(u, \text{IC})|>$ AC items he rated". For example if the SMAC defines Woody Allen Comedies, the annotation could be: "This reviewer rates this movie better than 4 out of the 6 Woody Allen Comedies he rated".

4. $\mathcal{U}_{prolific}(i) = \{u \in \mathcal{U}(i), |R(u, \text{IC})| \geq thresh\}$. In this case, we have enough other reviews by u in the SMAC that we can detect whether he has some bias (in a statistical sense) compared to the average. If we detect a bias, we can highlight it: "u rated $<|R(u, \text{IC})|$ items$>$ in AC items he rated; he rates this item better/worse than K of them; he likes these items more/less than the average reviewer". For example if the SMAC defines Comedies, the annotation could be: "u rated 125 Comedies; he rates this movie worse than 85 of them; he likes Comedies more than the average reviewer".

We are now ready to provide a description of Algorithm 3 used to interpret reviews in context. The main takeaway from this algorithm is the process of picking an optimal SMAC given an item and a reviewer.

The algorithm admits a movie and first retrieves the list of reviews to annotate and the list of SMACs which could be used for the annotation. The algorithm scans the list of reviews and annotates each of them. In lines 5-6, the trivial case where r is the only review of u is solved. In the other cases, the algorithm scans the list of potential SMACs and chooses the first one for which u has written at least one other review (lines 9-13). If it doesn't find such a SMAC, it uses all the reviews for the annotation (lines 14-15). Otherwise, it generates the annotation according of the number of others reviews (lines 16-22).

Algorithm 3 Review Interpretation Algorithm

Require: movie m.
1: $L_r =$ list of reviews of m;
2: $L_s =$ list of SMACs associated to m and ordered by specificity;
3: **for** (r in L_r) **do**
4: $u =$ user who has written r;
5: **if** $u \in \mathcal{U}_{one}$ **then**
6: Annotate r with "This is the only review by u in the system"
7: **else**
8: $N = 0$; $i = 0$;
9: **while** ($N \leq 1$ and $i <$length(L_s) **do**
10: $N =$ number of reviews of u in $L_s[i]$;
11: i++;
12: **end while**
13: AC = $L_s[i-1]$
14: **if** $N == 1$ **then**
15: Annotate r with "This is the only review by u of an AC movie out of his <X reviews>".
16: **else**
17: **if** $N > thresh$ **then**
18: Annotate r with "u rated <X AC movies>; he rates this movie better/worse than K of them; he likes these movies more/less than the average reviewer".
19: **else**
20: Annotate r with "u rated <N AC movies>; he rates this movie better/worse than K of them".
21: **end if**
22: **end if**
23: **end if**
24: **end for**

Related Work

In the e-commerce arena, the impact of reviews is quite clear: a recent study (Chevalier & Mayzlin 2006), confirmed also by (Chen, Dhanasobhon, & Smith 2007) shows that reviews impact sales, and that customers appear to read the reviews, rather than just rely on numeric scores. Furthermore, the reviews have the advantage of evaluating products with regard to their ability to match the consumers idiosyncratic usage (Chen & Xie 2007), with the effect that they attract more interest, and are perceived as more trustworthy than vendor-supplied information (Bickart & Schindler 2001).

Sentiment analysis is discussed in detail in (Pang, Lee, & Vaithyanathan 2002; Pang & Lee 2004; 2005) where a machine learning method is used to apply text categorization techniques to extract polar information (thumbs up, thumbs down). The work is based on finding minimum cuts in graphs to summarize subjectivity. The method described in (Popescu & Etzioni 2005) is based on an unsupervised information extraction system which mines product reviews to extract product features and their evaluation by reviewers. Our work is complementary to these approaches: rather than analyzing the review's content, we analyze the review's background.

Review "usefulness" is a vague and relative concept – a review that was useful for a casual consumer is often worthless to an expert. It is actually possible to some extent to predict the usefulness of reviews based on text analysis (Ghose & Ipeirotis 2007; Kim *et al.* 2006) but on the other hand, existing systems often present the most useful reviews first: this has the effect that these reviews continue to garner positive votes and also affect sales, since the first-presented reviews have disproportionate impact (Chen, Dhanasobhon, & Smith 2007). The ability to organize reviews into different buckets (which correspond to our user classification discussed above) will allow us to conduct a live evaluation where users are presented with different review rankings. We plan to do this in the future.

Summary and Future Work

We introduced SMACs, a powerful notion which characterizes a collection of items of interest and we showed how to use them in order to interpret reviews in context. SMACs have two appealing characteristics: (i) they can be easily described (e.g., Woody Allen Comedies, French restaurants in New York City) and, (ii) they are socially meaningful (based on domain-dependent criteria).

We discuss more uses of SMACs that we believe are critical to building a successful collaborative review system.

Personalization. We can use SMACs to connect an item under consideration by a user who has not rated it to the personal history of that user. For instance we can say This Woody Allen Comedy is rated better than 4 out the 6 Woody Allen Comedies that you have previously rated

Presentation. It has been shown that the order in which annotated reviews has a strong impact on user behavior (Ghose & Ipeirotis 2006). SMACs can be used in our application to rank the reviews. A natural order would be to start with $\mathcal{U}_{prolific}$ followed by $\mathcal{U}_{many}$ then $\mathcal{U}_{few}$. Some live experimentation would be required to test this feature.

Participation. While online reviewing systems observe a tremendous growth in content, it is known that most of the valuable content is produced by a small number of users (Rashid *et al.* 2006). We believe that displaying the number of reviews that an author has written for a given SMAC is a good incentive to encourage users to participate. Users could be tempted to show that they are knowledgeable, or if they are rated as “biased” they might be tempted to “correct” mistaken perceptions.

Acknowledgments

We wish to thank our colleagues Deepak Aggarwal, Cameron Marlow, Srujana Merugu, Bo Pang, David Pennock, Duncan Watts, and Cong Yu that provided us valuable information, references, data, and constructive criticism.

Two of the authors wish to thank restaurant *Christophe* in Amsterdam that helped inspire this paper.

References

Agrawal, R., and Srikant, R. 1994. Fast algorithms for mining association rules. In Bocca, J. B.; Jarke, M.; and Zaniolo, C., eds., *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, 487–499. Morgan Kaufmann.

Banerjee, A.; Basu, S.; and Merugu, S. 2007. Multi-way clustering on relation graphs. In *SIAM International Conference on Data Mining (SDM)*.

Bickart, B., and Schindler, R. M. 2001. Internet forums as influential sources of consumer information. *Journal of Interactive Marketing* 15(3):31–40.

Chen, Y., and Xie, J. 2007. Online consumer review: A strategic analysis of an emerging type of word-of-mouth. *Management Science*. Forthcoming. Available from `http://bear.cba.ufl.edu/xie/Publications/2007_Consumer%20Review_3rd%20Revision.pdf`.

Chen, P.; Dhanasobhon, S.; and Smith, M. 2007. All reviews are not created equal: The disaggregate impact of reviews and reviewers at amazon.com. In *Proceedings of the 28th Annual International Conference on Information Systems (ICIS2007)*. To appear. Available from `http://papers.ssrn.com/sol3/papers.cfm?abstract_id=918083`.

Chevalier, J. A., and Mayzlin, D. 2006. The effect of word of mouth on sales: Online book reviews. *Journal of Marketing Research* 345–354.

Ghose, A., and Ipeirotis, P. 2006. Designing ranking systems for consumer reviews : The impact of review subjectivity on product sales and review quality. In *Proceedings of the Workshop on Information Technology and Systems (WITS)*.

Ghose, A., and Ipeirotis, P. G. 2007. Designing novel review ranking systems: predicting the usefulness and impact of reviews. In *ICEC '07: Proceedings of the Ninth International Conference on Electronic Commerce*, 303–310. ACM Press.

Kim, S.-M.; Pantel, P.; Chklovski, T.; and Pennacchiotti, M. 2006. Automatically assessing review helpfulness. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-06)*, 423–430.

Pang, B., and Lee, L. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, 271. Morristown, NJ, USA: Association for Computational Linguistics.

Pang, B., and Lee, L. 2005. Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 115–124. Morristown, NJ, USA: Association for Computational Linguistics.

Pang, B.; Lee, L.; and Vaithyanathan, S. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, 79–86. Morristown, NJ, USA: Association for Computational Linguistics.

Popescu, A.-M., and Etzioni, O. 2005. Extracting product features and opinions from reviews. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 339–346. Morristown, NJ, USA: Association for Computational Linguistics.

Rashid, A. M.; Ling, K.; Tassone, R. D.; Resnick, P.; Kraut, R.; and Riedl, J. 2006. Motivating participation by displaying the value of contribution. In *CHI '06: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 955–958. New York, NY, USA: ACM Press.

Sarwar, B.; Karypis, G.; Konstan, J.; and Reidl, J. 2001. Item-based collaborative filtering recommendation algorithms. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, 285–295. New York, NY, USA: ACM Press.