Using Social Data for Personalizing Review Rankings

Vaishak Suresh*, Syeda Roohi*, Magdalini Eirinaki
Computer Engineering Department
San Jose State University, CA, USA

Iraklis Varlamis
Department of Informatics and Telematics
Harokopio University of Athens, Greece

ABSTRACT
Almost all users look at online ratings and reviews before buying a product, visiting a business, or using a service. These reviews are independent, authored by other users, and thus may convey useful information to the end user. Reviews usually have an overall rating, but most of the times there are sub-texts in the review body that describe certain features/aspects of the product. The majority of web sites rank these reviews either by date, or by overall "helpfulness". However, different users look for different qualities in a product/business/service. In this work, we try to address this problem by proposing a system that creates personalized rankings of these reviews, tailored to each individual user. We discuss how social data, ratings, and reviews can be combined to create this personalized experience. We present our work-in-progress using the Yelp Challenge dataset and discuss some first findings regarding implementation and scalability.

Keywords
Personalization; recommendation Engine; feature ranking; sentiment analysis;

1. INTRODUCTION
With more and more businesses selling their products or advertising their services online, customers rely on word of mouth in the form of customer reviews to make their decisions. Most of the current websites that feature product/business/service reviews list these reviews in reverse chronological order, or by employing heuristic metrics (e.g. ranking higher reviews of "super users", i.e. users with many reviews, or those with the most "helpful" votes). However, such a generic ranking requires from users to read or at least scan the tens or hundreds of reviews for one product/business/service.

Moreover, different people value different aspects of the same product/business/service. For example, when searching for a digital camera, one might be interested in the price and size, whereas another user may value the ease of use. Similarly, when searching for a good Italian restaurant, one user might value the ambience and wine list of a place, while another might prefer restaurants that are family-friendly. Ideally, users would like to be presented with only these reviews that highlight the qualities of a product/service that they value.

In this work, we present a system framework that addresses the above issue. In a nutshell, we create user profiles that reflect each user's preferences for specific restaurants and restaurant qualities (e.g. food, ambience, etc.). The profiles are created using the rating data as well as implicit preference as identified by applying aspect-based opinion mining to the reviews. Using these profiles, we identify similar users and rank their reviews for new restaurants higher. We also integrate the social network of the user, identifying those friends who have similar preference patterns with the active user, and highlight their reviews.

Proceedings of the 6th Workshop on Recommender Systems and the Social Web (RSWeb 2014), collocated with ACM RecSys 2014, 10/06/2014, Foster City, CA, USA.

Therefore, for the same restaurant, two different users will see a different list of reviews. The system is accompanied by a user-friendly interface that also highlights the main aspects of each review such that the user does not have to read the full text. To achieve this, we employ aspect-based opinion mining and neighborhood-based collaborative filtering techniques and integrate them in our system.

We also present a system prototype, built using the Yelp dataset[1] to demonstrate a first approach to this interesting problem [7]. Without loss of generality, we focus on restaurant review recommendations, however our approach is easily extended to any other product/business/service as long as reviews, ratings, and an underlying social network are available. The personalized presentation of the reviews is a subjective matter and therefore very hard to evaluate without involving real users, however we provide some first empirical results. We should stress that this is a work in progress and our focus in this paper is to introduce this mash-up idea along with an initial approach to the problem, as well as our thoughts on how such a system could be further enhanced.

The rest of the paper is organized as follows: we present our system's design in detail in Section 2. We provide a first-cut approach on extending the proposed model using social network connections and feedback in Section 3. Some discussion on the prototype implementation and evaluation is included in Section 4. An overview of the related work is provided in Section 5 and we conclude with our plans for future work in Section 6.

2. SYSTEM DESIGN
The system architecture is shown in Figure 1. It comprises two main modules, an offline processing module, where the user profiles are being generated and the feature extraction and rating happens, as well as an online module, that generates real-time recommendations.

2.1 Offline Processing
There are two phases of offline processing: namely aspect summarization and user preference generation.

2.1.1 Aspect Summarization
This module aims at extracting the important features from each review, along with their polarity weight. To perform this we employ the subjectivity lexicon [8] in order to map weak and strong positive and negative words to numeric values (ranging from -4 to +4). Using a master list of positive and negative opinion words from an opinion lexicon [5] we created a list of negation words (not, no, nothing etc.) which inverse the sentiment, and intensifiers (too, very, so, etc.), which increase the intensity of the sentiment (these are referred as "TOO words" in our algorithm). More specifically, words of each review are tagged using the default POS (parts of speech) tagger from

* Vaishak Suresh and Syeda Roohi are currently affiliated with Intuit Inc. and HP respectively.

[1] http://www.yelp.com/dataset_challenge/

NLTK[2], a natural language processing Python package. This is done using the Treebank corpus. The text augmented with tags is then split into sentences and then into words. Each word is then examined to determine its type.

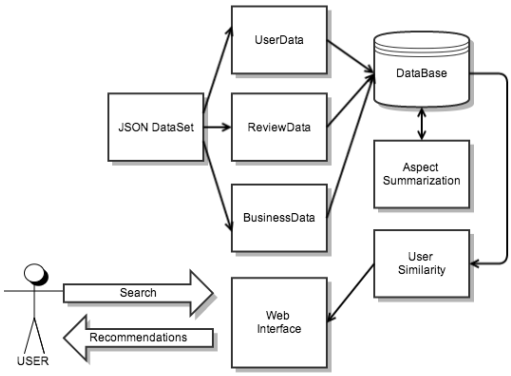

Figure 1: System Architecture

If the word is POS-tagged as an adverb or an adjective, it is considered as an *opinion* word. If the opinion word is POS-tagged as superlative or comparative the score is set to the maximum (+4) or minimum (-4) based on the polarity. During this process, the words that modify the polarity (e.g. "not") and degree (e.g. "too", "very") are also considered for scoring the opinion word. The presence of these words can inverse or increase the sentiment score of the aspect respectively. The words POS-tagged as nouns are potential candidates to be the *feature* words. Apart from using the pre-defined feature look up file, these words are also tested to find any synonyms using the WordNet interface in NLTK.

Once the features and opinion words in a sentence are determined, a mapping is made to a feature and opinion word based on the distance between them. The aggregated opinion score for each feature is calculated for all the sentences in the review as mentioned above and the review document is updated with these values in the system's database. The algorithm performing this process is outlined in Figure 2.

2.1.2 User profile generation

In order to generate personalized review rankings, we follow a neighborhood-based collaborative filtering approach. Given a user $\mathbf{u}$ and the set of businesses they have rated and/or reviewed $\mathbf{B_u}$, each user is represented by a profile vector $\mathbf{U} = \{\mathbf{p_1}, \dots, \mathbf{p_k}\}$ where $\mathbf{p_i}$ denotes the preference of a user for a business i and k is the total number of businesses in the system. We define $\mathbf{p_i}$ as follows:

$$p_i = \begin{cases} s_{ui} & if \quad i \in B_u \\ 0 & if \quad i \notin B_u \end{cases} \qquad (1)$$

where s_{ui} represents the cumulative preference score of a user u for a business i, calculated using their overall rating or opinion on specific aspects of the business that are identified by their review for it. We introduce three alternative ways of calculating the

```
For each sentence in the review
 For each word in the sentence
  if the POS of the word is Adverb (RB) and is a TOO word
        Save the TOO word position
  if the word is 'and'
        Continue the too rule
  # opinon word
  if the word is in the subjectivity lexicon or in the master list
        if the POS tag is a superlative adverb (RBS) or adjective (JJS)
                        Set the superlative flag
        if the POS tag is a comparative adverb (RBR) or adjective (JJR)
                        Set the comparative flag
        if word in subjectivity lexicon
                        Set the word score
        else if word in positive master list
                        Set the word score to +1
        else if word in negative master list
                        Set the word score to -1
        if too exists and is adjacent
                        if word is positive
                                        increase the score by 1
                        if word is negative
                                        decrease the score by 1
        if superlative flag Set
                        if word is positive
                                        Set the score to +4
                        if word is negative
                                        Set the score to -4
        if comparative flag Set
                        if word is positive
                                        Set the score to +3
                        if word is negative
                                        Set the score to -3
        if the opinion word is in negative context
                        negate the sentiment of the score
  Save the opinion_word_position and score
  # Feature word
  if the POS of the word is a Noun (NN)
        if the word is in feature list or a synonym of a feature
                        Save the feature and position
 Apply the opinion score to the potential feature in the sentence
Aggregate the score for each feature
```

Figure 2: Opinion score assignment algorithm

preference score, namely using only the rating of the user, using the specific review opinion scores, or weighing them by the overall preference/dislike of the user for each aspect, as shown in Equations 2, 3 and 4 respectively:

Rating-based preference score

$$s_{ui} = r_{ui} \qquad (2)$$

where r_{ui} denotes the star rating of user u for business i.

Business-based preference score

$$s_{ui} = \sum_{a \in R_{ui}} o_{ua} \qquad (3)$$

where R_{ui} denotes the set of aspects included in the review of user u for business i and o_{ua} is the opinion score calculated for aspect a in this particular review.

Review-based preference score

$$s_{ui} = \sum_{a \in R_{ui}} w_{ua} . o_{ua} \qquad (4)$$

where w_{ua} denotes the overall preference/dislike of user u for aspect a, as expressed by their opinions on all reviews they've written. This can be calculated as the normalized sum of all the scores o_{ua} in all the reviews R_u.

[2] http://www.nltk.org

Once the user profiles are created, we employ a user-based collaborative filtering technique to find similar users. In our implementation, we have used the Pearson correlation coefficient and the open source libraries provided by Apache Mahout.

2.2 Online Recommendations

This step is used to rank and recommend reviews in real-time, as the user navigates the system and searches for new restaurants. When a given user searches for a specific restaurant, the recommendation engine computes the similarity of the current user with all the reviewers of the particular business and ranks and presents the related reviews in descending order of similarity. As a result, each user will be presented with a different set of reviews for the same business.

Moreover, the interface allows the end user to get the gist of the reviews without the need to read the entire review text. For each review, the overall star rating as well as the most important aspects of each review, are prominently shown. The aspects are intuitively marked as strong/weak positive/negative, by using colors and thumbs up/down images. We should stress that the same aspect might appear in more than one reviews and one review might contain more than one aspects.

3. SOCIAL NETWORK FEEDBACK

When available, information related to the user's social network can be incorporated in our model. There are two alternative ways this can be done, either at the last step of the process, or during the profile generation.

In the first case, the similarity between the user and their friends is calculated when the user searches for the restaurant. The friends' reviews for this restaurant are separately ranked and presented in a different list so that they are easily identifiable.

In the second case, the user preferences are weighed by the user's friends' opinion scores. To incorporate the social network feedback in the model, we extend Equation 1 as follows:

$$p_i = \begin{cases} w_{F_u i} \cdot s_{ui} & if \quad i \in B_u \\ w_{F_u i} & if \quad i \notin B_u \end{cases} \qquad (5)$$

where F_u is the set of friends of user u and $w_{F_u i}$ can be defined as follows:

$$w_{F_u i} = \frac{\sum_{f \in F_u} s_{fi}}{|f|} \qquad (6)$$

Equation 6 can be easily extended to incorporate the similarities between users.

Note that in this extension, we also address the cold-start problem since the user profile can be filled by social network feedback, even when the user has few, or none reviews/ratings in the system.

4. PROTOTYPE EVALUATION

We have already implemented a prototype based on our system design described in the previous sections using the Yelp dataset. Our prototype implements the business-based preference profile, assuming that the product aspects are predetermined. A screenshot of our prototype is shown in Figure 5. Each review is accompanied by some metrics showing the calculated polarity and subjectivity of the review as well as the similarity of each reviewer to the user. The end user may further refine the personalized list of reviews by filtering only those that come from his/her friends or by feature (e.g. location, food, etc.). More technical details on the implementation are included in [7][3].

We have load-tested the prototype, deployed on Tomcat Server on a machine with the following configuration: Intel i5-2410M CPU @2.30 GHz, 64-bit OS with 4 GB RAM. As shown in Figure 3, the response time increases linearly with the number of users and can handle multiple simultaneous requests in real-time (the system crashed after 175 simultaneous requests, as MongoDB can't handle that many connections).

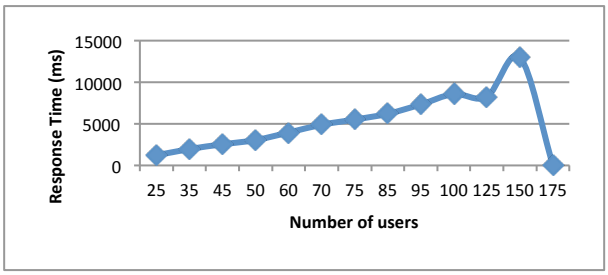

Figure 3: Response Time per number of concurrent users

We also performed an empirical evaluation of the recommendations using the following methodology: we randomly picked 50 users and generated top-5 recommendations for a specific restaurant. We then asked human evaluators to rate each recommended review on the following scale: 1 = "irrelevant", 2 = "somewhat relevant", 3 = "very relevant". To assign the rankings, the evaluators were asked to identify 2-4 aspects highlighted in each user's review[4]. If the recommended review included >50% of the aspects, then it received a 3, if it was very uninformative or did not include any aspects it received an 1 and everything else received a 2. We employ precision as our evaluation metric and define *Prec3* and *Prec2,* measuring how many recommendations received a "3" or a "2 or 3" rating respectively.

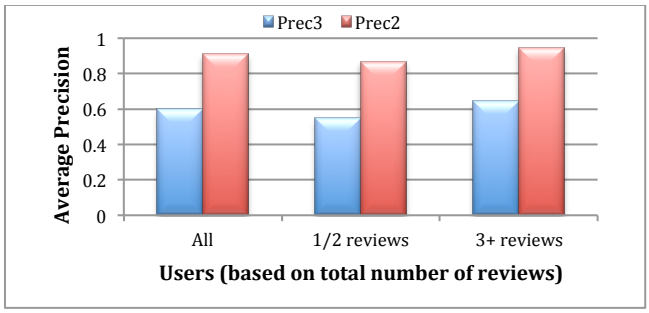

Figure 4: Average precision

We observe that the system manages to recommend 60% or more very relevant recommendations, while the accuracy reaches to 100% when the somewhat relevant recommendations are included. The accuracy increases more when the "cold-start" users (i.e. users with only 1 or 2 reviews contributing to 48% of the subset) are removed. We noticed that most of the times the system failed to generate useful recommendations was when the style of the review was sarcastic and/or focused on non-trivial issues (e.g. servers engaged on a fight). Moreover, as the aspects currently used are very high-level, the results did not capture specific food

[3] A screencast of the prototype is available at: http://youtu.be/vMz5CobpIw4

[4] The aspects were not identical to the ones used by our prototype. Instead the evaluators were asked to identify anything that stood out (e.g. user favors short reviews, values price/service/food, etc.)

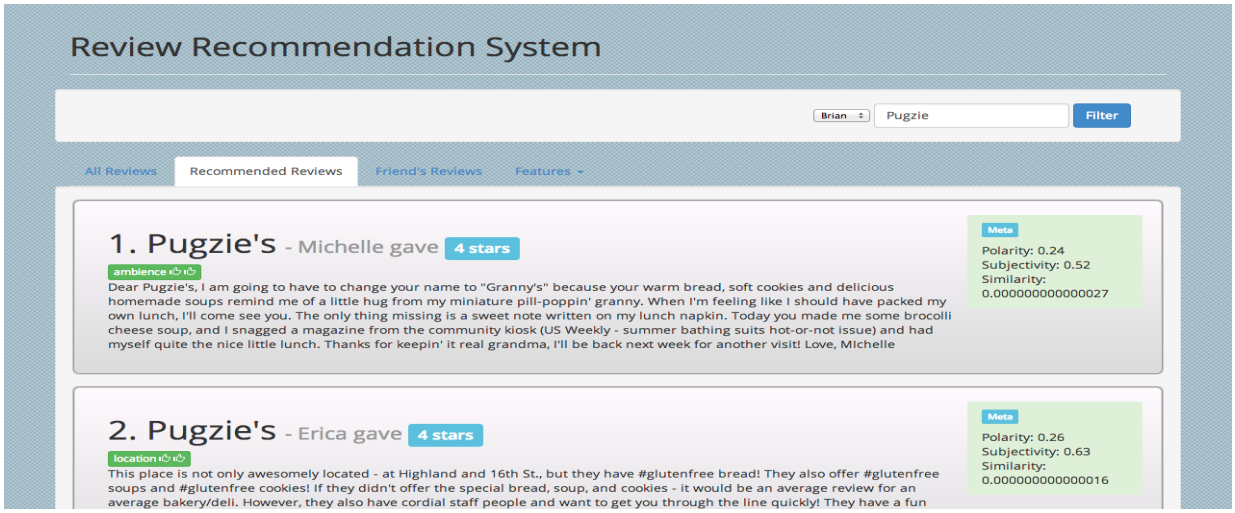

Figure 5. Client application – personalized recommended reviews

preferences of the users (e.g. vegan vs. meat lover). On the other hand, the algorithm has been quite successful in identifying priorities such as the atmosphere/service quality/drink options etc. As a reference, the number of individual user reviews for this subset ranged from 1 to 36 (mean = 4.7, median = 3).

5. RELATED WORK

Many interesting works exist that focus on extracting the opinions from the customer reviews [5]. The most recent ones employ features as an additional tool in representing the semantic orientation of a review [1, 2, 4]. This is an important line of work that provides very useful input in the creation of the rich user profiles of our system. The algorithm we introduce in this paper is along the same lines, however we should note that any similar approach could be easily integrated in our system.

None of the major web sites that include reviews as an indispensable part of their business provide aspect-oriented personalized review rankings. For instance, Amazon ranks reviews by helpfulness (number of "helpful" votes received) without providing any summary of the reviews, other than the overall star rating. Netflix's rating system is also mainly based on the star ratings, whereas Google shopping allows users to create a list of pros and cons in addition to the review, but ranks them based on the review date. Finally, Yelp, whose dataset we are employing in this study, ranks reviews by helpfulness. It also provides an overall summary for each business in terms of several aspects (e.g. friendly for kids, romantic, etc.), as well as a short summary of the most common comments in the reviews. The last two companies have some underlying social network that is not, however, utilized in re-ranking or personalizing the reviews.

Similarly, not much work has been done in the research community. The problem of using helpfulness as a way to rank results is discussed in [3]. The authors conclude that for experience goods, users prefer a brief description of the "objective" elements of the item and then a subjective positioning, described by aspects not captured by the product description. Our work not only addresses these findings, but also proposes ways of personalizing the rankings for each user, taking into consideration their social network as well. Helpfulness is also used in [6] as a way to filter out interesting reviews. This work addresses the same problem in a somewhat different way. The authors employ the feedback given by the community in terms of how helpful one's reviews are, along with several other content-, social-, and sentiment-based features in order to classify a review as helpful or not. The main differences with our approach are that the sentiment is based on explicit sub-ratings given by the users to several predetermined aspects of a service as well as the fact that theauthors assume that a "helpfulness" vote exists for each review in the dataset.

6. CONCLUSIONS

The amount of online reviews for products and services has grown to such extend that often makes it impossible to read all of them. In this work we propose a system that personalizes the order in which the reviews are shown and provides an intuitive interface that allows the users to see the important aspects of each review in a glimpse. An initial evaluation shows promising results. As part of our future work we plan to integrate further these two types of recommendations and enhance them by introducing trust-based and reputation metrics. We also plan to perform a more extensive evaluation of the usefulness of such reordering.

7. REFERENCES

[1] X. Ding, B. Liu, P. S. Yu, *A holistic lexicon-based approach to opinion mining*, in Proc. of WSDM '08

[2] M. Eirinaki, S. Pisal, J. Singh, *Feature-based Opinion Mining and Ranking,* J. of Computer and System Sciences (JCSS), 78(4), pp.1175-1184, July 2012

[3] A. Ghose, P. Ipeirotis, *Designing Novel Review Ranking Systems:Predicting the Usefulness and Impact of Reviews*, in Proc. of ICEC '07

[4] H. Guo, H. Zhu, Z. Guo, X. Zhang, Z. Su, *Address standardization with latent semantic association,* in Proc. of ACM KDD'09

[5] B. Liu, Sentiment Analysis and Opinion Mining, Morgan & Claypool Publishers, May 2012

[6] M.P.O'Mahony, B. Smith, *A classification-based review recommender,* Knowledge-Based Systems, 23(4), pp. 323-329, May 2010

[7] S. Roohi, V. Suresh, M. Eirinaki, *Aspect based Opinion Mining and Recommendation System for Restaurant Reviews,* demo paper, in Proc. of ACM RecSys 2014

[8] T. Wilson, J. Wiebe, and P. Hoffmann, *Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis.* In Proc. of HLT-EMNLP-2005.