CS 4780 Project Proposal: Generalized Review Ranking System with Attribute Matching

Yunlu Li, Ian Walk, Megan Marshall, Yincheng Ren

1 Introduction

User reviews are a common feature of online shopping and searching services, including Amazon, TripAdvisor, Yelp, and many more. They can be posted quite easily by anyone with an account and are not vetted for accuracy or relevance, but are often the only way that a user can compare the quality of advertised goods without turning to other sites and services. As such, evaluating the relevance and accuracy of user reviews is not only important to sites like Rotten Tomatoes and Yelp, which rely nearly exclusively on user reviews for their site content, but also for sites like Amazon. Increasing trust in user reviews can increase the confidence that users have in the purchases that they make, keep users on the site longer, and generally improve the quality of information found on the site.

Despite this, users are rarely provided with tools to help them find the reviews that are the most relevant and informing to them. Reviews can sometimes be sorted by simple metrics such as product rating or most helpful, but these are often vulnerable to abuse through posting false reviews or marking other reviews as unhelpful. One can quite easily imagine a vendor on Amazon creating dozens of accounts to post positive reviews while marking any negative reviews as irrelevant or unhelpful. Furthermore, these metrics are not personalized to the site users or their values. Two users who may be interested in very different characteristics would be presented with the same set of user reviews.

In this project, we aim to increase the relevance of reviews to individual users. We hope to leverage user data in order to discover the aspects of products they find the most important, then rank the available reviews by relevance to the current user. We believe this would make reviews more valuable to users and allow users to better judge how useful a particular product would be to them.

2 Previous Research

There are several relevant research papers.

- In *Real-time personalized twitter search based on semantic expansion and quality model*[1], their framework integrates the semantic features and social attributes which are utilized to make a comprehensive rank for a tweet. This framework has a tweet quality model that is built to distinguish high quality tweets and improve the ranking performance. To deal with the problem of profiles having only a few words and therefore little data to use, this research was also able to identify semantically similar words and include them in the set of words related to a profile. Using this quality distinguish model will help us overcome the information overload brought by a large amount of reviews.
- In *Using Aspect Extraction Approaches to Generate Review Summaries and User Profiles* [2], research is conducted to better identify words that indicate a certain attribute in a review. Methods that used to be the most commonly used made incorrect assumptions about independence and resulted in unrelated words being grouped together. This research expands upon other papers to further improve expanding the set of words that indicate an attribute is being discussed in a review.
- In *Latent aspect rating analysis on review text data: a rating regression approach*[3], authors use a probabilistic rating regression model to evaluate an individual reviewer's opinion based on the different aspects. This involved identifying which aspects a user was referring to in their review as well as the strength and direction of their opinion for that aspect. The research was able to expand an initial set of words to find similar ones

[1] https://doi.org/10.1016/j.neucom.2016.10.082

[2] https://www.aclweb.org/anthology/N18-3009.pdf

[3] https://doi.org/10.1145/1835804.1835903

that may also indicate that the user is referring to a certain aspect. We can use this research to help discover reviewer's latent coverage and potentially improve our model accuracy. This research will also help identify aspects in the reviews. The research exclusively used TripAdvisor data and did not evaluate it on reviews of other products.

- In *More focus on what you care about: Personalized top reviews set*[4], authors proposed a model to use personalization criteria to further improve the rankings of reviews and was influenced by the second paper above. This research used the number of words associated with each aspect in the review to determine the strength of opinion for that aspect and did not look at the sentiment in the review. It was then able to infer the aspects the user cared about and highly rank reviews that mentioned these aspects. This research also involved finding similar reviewers. If a user had not yet left a review for a product, these similar reviewers could then be used to predict which aspects this user would also care about and rank the reviews accordingly. This research focused exclusively on data from Trip Advisor and Yelp.

3 Proposed Work

After reading through previous research, we realized that it only can be applied to single-category website. For example, previous research focused on Yelp which only contains reviews of restaurants and Tripadvisor which only contains reviews of hotels. For our project we could expand the review ranking to a multi-category website, like Amazon. First, we aim to testify the assumption that people hold different preference of attributes for products under different categories. Second, we would like to generalize the review ranking system with attribute matching to multi-category website by applying clustering algorithm.

Workflow

1. For each category, perform attributes extraction on reviews with the help of existing tools. For attributes that appear across categories, we plan to use them to achieve the first goal.
2. For each review, based on its category, we generate a vector form indicating its attribute emphasis. Our tentative plan for calculating emphasis is by counting the number of words talking about the attribute.
3. On the individual level, we retrieve the attribute preference across all categories based on his/her reviews. For reviews under the same category, we add attribute preference up. This means, for each person, for each category, we have a vector form indicating his/her attribute preferences.
4. For each category, we cluster people based on their aspect preferences by machine learning algorithms.
5. When ranking reviews, we first match category and then select reviews written by users from the same cluster.

In this project, we plan to testify the assumption that people's attribute emphasis varies across categories and try to generalize the attribute matching system for review ranking to multi-category website. Furthermore, in previous research, attribute matching was achieved by similarity measurement function, but in this project we tend to apply clustering algorithm to see which methods work more effectively and efficiently.

The dataset[5] that we use is a large crawl of product reviews from Amazon. This dataset contains 82.83 million unique reviews, from around 20 million users. The dataset also contains categorical information which fulfills our purpose.

Evaluation

Finally, we are planning to evaluate our model based on two methods.

- Classical evaluation: split the Amazon data to train set and validation set. After training our model using the train set, we can predict what aspects the user actually care about for the product and validate those aspects with validation set.
- Compare with different review ranking strategy: We can run different ranking strategies on the Amazon data to demonstrate the effectiveness of our model. We can evaluate models by comparing the rank of reviews and the number of up votes each review received. The last paper listed in previous research also performed a case study to manually evaluate the rankings of a few randomly chosen products and users. This could also be used to better evaluate the extend different aspects are included in the review.

[4] https://doi.org/10.1016/j.neucom.2016.10.081

[5] https://nijianmo.github.io/amazon/index.html