4780 Project Report: Generalized Review Recommending with Attribute Matching

Yunlu Li*
yl4df@virginia.edu
University of Virginia

Megan Marshall
mem5ak@virginia.edu
University of Virginia

Yincheng Ren
yr5ka@virginia.edu
University of Virginia

Ian Walk
imw6jy@virginia.edu
University of Virginia

ABSTRACT

As more and more cross-category shopping websites emerge, users usually find it hard to retrieve most relevant reviews in making shopping decision. In helping users to find relevant reviews on cross-category websites, we applied attribute extraction techniques to construct profiles for each user, which shows their preferences regrading different attributes. With use of clustering algorithms, we group users with similar preferences and by thus we recommend reviews written by users in the same cluster for each category. Also, we successfully testify whether uses' attribute preferences have relationship across categories. This method differs from previous methods using similarity measurement function on each user, which could be running on user by group cases. Based on our evaluation methodology, the proposed method achieves satisfactory performance in presenting relevant reviews.

KEYWORDS

Review Recommending, Clustering, Attribute Extraction

ACM Reference Format:
Yunlu Li, Megan Marshall, Yincheng Ren, and Ian Walk. 2020. 4780 Project Report: Generalized Review Recommending with Attribute Matching. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

User reviews are a common feature of online shopping and searching services, including Amazon, Trip Advisor, 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 informative 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 propose an efficient way for increasing the relevance of reviews to individual users on multi-category website. We hope to leverage user review data in order to discover the aspects of products they find the most important, then recommend the available reviews by relevance to the current user by clustering algorithms. 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. The contribution of this work can be summed up as follows. First, we aim to testify the assumption that people hold different preference of attributes for products under different categories; Second, we would like to recommend the review with attribute matching to multi-category website by applying clustering algorithm.

2 PREVIOUS RESEARCH

There are several relevant research papers.

- In *Real-time Personalized Twitter Search Based on Semantic Expansion and Quality Model*[1] [4], 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

*All authors contributed equally to this research. Name is ordered by last name alphabetically.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

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

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.

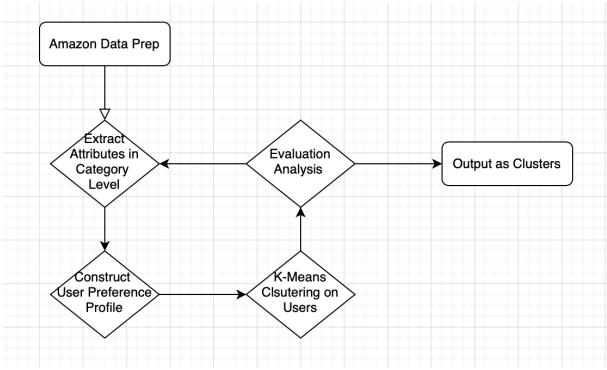
- In *Using Aspect Extraction Approaches to Generate Review Summaries and User Profiles* [2] [1], research is conducted to better identify words that indicate a certain attribute in a review. Methods that used to be the most commonly used made incorrect assumptions about independence and resulted in 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 approach*[3] [2], 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 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] [3], 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 most work focused on single-category website. For example, previous research focused on Yelp which only contains reviews of restaurants and Trip Advisor which only contains reviews of hotels. For our project we could expand the review recommending to a multi-category website, like Amazon. At the same time, we found similarity measurement using in previous work is used in user by user cases, so we proposed a new way to apply clustering tools to group similar users together. Based on the limitation of previous work, as we have mentioned in the introduction, we have made two main contributions. First, we aim to testify the assumption that people hold different preference of attributes for products under different categories; Second, we would like to recommend the review with attribute matching to multi-category website by applying clustering algorithm. The proposed work flow is presented as follows.

Workflow

(1) For each category, perform attributes extraction on reviews with the help of existing tools. For users who wrote reviews 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.
(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 and take the average. This means, for each person, for each category, we have a vector form indicating his/her attribute preferences. We called this vector form as user preference profile.
(4) For each category, we cluster people based on their attribute preferences by machine learning algorithms.
(5) Finally we performed evaluation and analysis for both of our goals by using a customized evaluation methodology, which will be addressed later.



In this project, we plan to test the assumption that people's attribute emphasis varies across categories and recommend reviews with attribute matching on a multi-category website. In previous research, attribute matching was achieved by similarity measurement function in users by users cases, but in this project we intend to apply clustering algorithm to create mappings between users and groups. More details and methodology about each step are discussed as follows.

4 EXPERIMENT DESIGN AND RESULT

4.1 Data Preparation

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

Due to computational limitation of our own laptops, we used a

[2]https://www.aclweb.org/anthology/N18-3009.pdf
[3]https://doi.org/10.1145/1835804.1835903
[4]https://doi.org/10.1016/j.neucom.2016.10.081
[5]https://nijianmo.github.io/amazon/index.html

subset of the whole data set. We only selected 10 categories. The original data format is one-review-per-line in json. We parsed and cleaned the data, keeping 5 columns including "overall", "vote", "verified", "viewerName", "reviewText". We kept the first three columns for potential ranking purpose. The fourth was used for clustering users. The fifth contains our review text data. The text of the reviews underwent stemming, stop word removable, and lower case conversion.

4.2 Category Level Attribute Extraction

Our experiment relied on the assumption that the more a user cares about an attribute of an item, the more likely they are to mention that attribute in the review. These attributes need to be identified so they can later be used to identify users that have similar preferences and therefore, potentially more relevant reviews for the user.

A corpus was created with the processed terms and weightings based on the TFIDF values. From this corpus, Gensim's Latent Dirichlet Allocation method was used to find the most popular attributes. This is an unsupervised technique that is common in topic modelling. These attributes were represented as a sum of the processed terms that best represent the attribute and their respective weights. We used the top 10 most popular attributes as represented by the top 10 terms that were most representative of the attribute. We tried slight adjustments to the parameters involved in the process, such as the number of times the corpus was passed through, but did not find a significant difference in the average magnitude of the weights for the top 10 terms or how frequently terms appeared in the top 10 terms for multiple attributes.

We decided to use Latent Dirichlet Allocation to find the most prevalent attributes in the reviews because we did not feel confident that we would be able to come up with all of the different attributes ourselves. We wanted a method that would examine the data itself as opposed to introducing bias based on our own attribute preferences.

One sample output from this step is shown below. We extracted ten topics from "game" Category, and each topic has ten associated words. These words are top ten words that have highest contribution to this topic, with their contribution shown as coefficients.

```
Topic: 0 Word: 0.008*"play" + 0.007*"like" + 0.005*"time" + 0.005*"stori" + 0.005*"good" +
0.005*"charact" + 0.004*"great" + 0.004*"graphic" + 0.004*"enjoy" + 0.004*"love"
Topic: 1 Word: 0.012*"fallout" + 0.010*"evil" + 0.009*"resid" + 0.008*"horror" + 0.008*"hunt" +
0.007*"mechan" + 0.007*"dumb" + 0.006*"exel" + 0.006*"slash" + 0.006*"penni"
Topic: 2 Word: 0.015*"fighter" + 0.014*"deliv" + 0.013*"remast" + 0.011*"phone" + 0.009*"arcad"
+ 0.008*"persona" + 0.007*"saturn" + 0.007*"street" + 0.007*"dont" + 0.007*"emblem"
Topic: 3 Word: 0.342*"love" + 0.050*"daughter" + 0.030*"cute" + 0.020*"mario" + 0.019*"super" +
0.018*"pokemon" + 0.011*"nintendo" + 0.008*"wife" + 0.008*"mega" + 0.007*"refund"
Topic: 4 Word: 0.458*"great" + 0.057*"cool" + 0.011*"alright" + 0.010*"duti" + 0.009*"madden" +
0.008*"fifa" + 0.007*"microsoft" + 0.007*"wrestl" + 0.007*"telltal" + 0.006*"cheaper"
Topic: 5 Word: 0.158*"good" + 0.049*"awesom" + 0.031*"love" + 0.031*"like" + 0.031*"play" +
0.031*"gift" + 0.030*"kid" + 0.020*"best" + 0.020*"enjoy" + 0.018*"great"
Topic: 6 Word: 0.018*"song" + 0.018*"husband" + 0.015*"lego" + 0.015*"sega" + 0.014*"genesi" +
0.013*"sale" + 0.013*"sonic" + 0.013*"season" + 0.012*"dreamcast" + 0.012*"danc"
Topic: 7 Word: 0.085*"thank" + 0.081*"excel" + 0.072*"work" + 0.068*"product" + 0.053*"great" +
0.040*"perfect" + 0.034*"nice" + 0.029*"describ" + 0.029*"condit" + 0.027*"fast"
Topic: 8 Word: 0.023*"work" + 0.015*"mous" + 0.015*"control" + 0.013*"xbox" + 0.013*"case" +
0.011*"need" + 0.010*"deliveri" + 0.010*"great" + 0.009*"download" + 0.009*"window"
Topic: 9 Word: 0.013*"grandson" + 0.011*"control" + 0.011*"headset" + 0.008*"keyboard" +
0.008*"work" + 0.007*"button" + 0.007*"batteri" + 0.007*"sound" + 0.006*"comfort" + 0.006*"grip"
```

Some of these attributes have fairly clear intuitive meaning. The first attribute represents caring about being immersed in the world of the game. The world is made up by the characters and story lines in the game and is displayed by the graphics. The second attribute appears to correspond to violence in video games and the third may characterize the feeling of the original fighter games and old-school arcades. However, some attributes make less sense. For example, attribute 5 only seems to imply that this product was a gift to a child. This does not actually reflect an element of a product. Attribute 7 may be referring to some of the logistics, including that the game does work and arrived quickly; however, attribute 8 appears to be very similar to that as well. Further analysis into these attributes is conducted in the evaluation step.

4.3 User Preference Profile Construction

After extracting top attributes for each topic, we tried two different method to score each review on each topic. We first utilized Word2Vec to classify reviews. However, due to limited calculation capacity and insignificant result, combined with our assumption that the more users care about an attribute, the more they mention that attribute in reviews, we chose another method which is bag of words to count the number of words talking about the attribute. We first stemmed and removed stop-words for each review. We assigned each review 10 scores, corresponding to 10 attributes. Then we counted the frequency of top 10 words that were most representative of the attribute. The product of the frequency of words and its weights based on the TFIDF values became the score. After applying this process to all reviews in each category, users had a [x,10] vector (x is the number of reviews the same user have in each category) indicating their preference, enabling us to further cluster our users. A sample output from this step is shown below.

0	1	2	3	4	5	6	7	8	9
0.004	0.0	0.0	0.000	0.458	0.018	0.0	0.053	0.010	0.000
0.036	0.0	0.0	0.000	0.011	0.124	0.0	0.000	0.000	0.000
0.000	0.0	0.0	0.000	0.000	0.000	0.0	0.000	0.000	0.000
0.008	0.0	0.0	0.000	0.000	0.031	0.0	0.000	0.000	0.000
0.016	0.0	0.0	0.342	0.458	0.080	0.0	0.053	0.010	0.000
...	...	...	...	...	...	...	...	...	...
0.005	0.0	0.0	0.000	0.000	0.158	0.0	0.000	0.000	0.000
0.000	0.0	0.0	0.000	0.000	0.000	0.0	0.000	0.000	0.000
0.004	0.0	0.0	0.342	0.000	0.061	0.0	0.034	0.000	0.000
0.148	0.0	0.0	0.019	0.057	0.697	0.0	0.178	0.057	0.016
0.036	0.0	0.0	0.000	0.000	0.251	0.0	0.000	0.015	0.011

4.4 Clustering

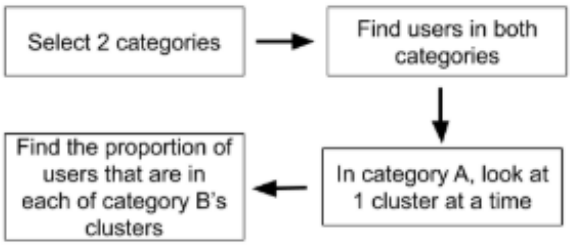
Now that each person's preferences are represented with a vector, we can cluster these people based on the vector of their category preferences. We examined different approaches to clustering, including mean shift and k-means. We found that mean shift grouped the users into few and very unbalanced clusters, where 90% or more

of all people were in a single group, and many of the others consisted of only a single outlying individual. As this did not match the behavior we hoped to see from our clustering, we instead decided to use k-means with a cluster size of seven. This resulted in a more expected outcome with groups that were more equally represented. Unlike mean shift, k-means forms a pre-defined amount of clusters, which we believe contributed to the more balanced results that it produced with this data set.

These clusters can be used by a ranking algorithm to prefer user reviews from users that fall within the same cluster, as similar users are more likely to care and post about similar subjects. We believe this would enable a ranking algorithm to improve it's relevance to individual users.

4.5 Evaluation

Our experiment wanted to test if the attribute preferences a user has for one category were related to attribute preferences for another category. To determine this, two categories were analyzed at a time. Given the large number of pairwise combinations that exist for ten categories, only four categories, automotive, beauty, video games, and digital music, were used. First, users that had written at least one review in both of the categories were found. From there, they were broken down into groups based on their cluster for the first category. Then, for each of those clusters, the proportion of those users that were in each of the clusters for the second category were found. For example, there are 2,982 users that have written a review in both the music and automotive categories. Of these 2,982, 383 users are in cluster 0 for the automotive category. Of those 383 users, for the music category, 85 are in cluster 0, 18 is in cluster 1, 103 are in cluster 2, 48 are in cluster 3, 127 are in cluster 5, and the final 2 are in cluster 6. 0 of the 383 users are in cluster 4 of the music category. This is summarized in the flowchart below:



After these cluster breakdowns were found, they need to be compared to the cluster breakdowns based on all of the users in that category. Originally, a chi square test was considered where the expected values would be based on the counts in each cluster for all users in the category. However, very few users in the data set had reviews in multiple categories and these expected counts were too low to meet chi square assumptions.

A two proportion test was conducted for each of these pairwise proportions. Each category pair involved up to 49 tests as the proportion of users in each cluster in category B given each of the clusters in category A needed to be tested against the general cluster proportion in category B. Virtually all of the category pairs had at least one cluster in category A that had too few users to allow for an acceptable sample size. In this case, all of the comparisons from that entire cluster from category A were omitted from testing. This did not mean that the other clusters in category A were not related to those in category B so the remaining cluster pairs from the category pair remained in the analysis. 77 pairwise tests were omitted for this reason, leaving 511 pairwise tests. Once the clusters with very few users in category A were removed, the sample sizes of these tests was fairly reasonable. Half of the category pairs had over 1000 shared reviewers. These reviewers just were not distributed across categories evenly enough to allow for chi square to be possible.

To account for the multiple tests, the Bonferroni correction was used. This correction allowed the overall significance level to remain at 95% confidence by comparing the p-values to $\frac{0.05}{511}$ or approximately $9.78 * 10^{-5}$. 96 of the 511 tests, or 18.8%, were found to be statistically significant at 95% confidence. Each of the four categories had at least 1 cluster that had a significantly different cluster breakdown for another category so while not all of these cluster pairs were found to be related, these categories were all partially related to each other. This leads to the conclusion that a reviewer's attribute preferences for a category can be related on their attribute preferences for a different category. However, this is situational as it is not true for most category/cluster combinations and even when found to be significantly different, the difference in actual magnitude is small. Any potential benefit from combining categories would be minimal.

Our second goal involved the accuracy of the individual clusters. Since we are using clustering algorithms in our project, it is not as easy as supervised learning to evaluate results. We customized the evaluation procedure in three Steps.

Step 1: Calculate the centroids for each cluster. Based on the centroids, we know the the topic preference emphasis for this cluster.

Step 2: Manually generate some users with reviews talking about specific attribute preference. We know clearly which cluster this user should belong to with the help of the preference given by centroids in the last step. For each category, we aim to create 20 users with 3 reviews each.

Step 3: Run the whole procedure according to workflow and see how many users are in the group that they should be. The percentage is our evaluation metric.

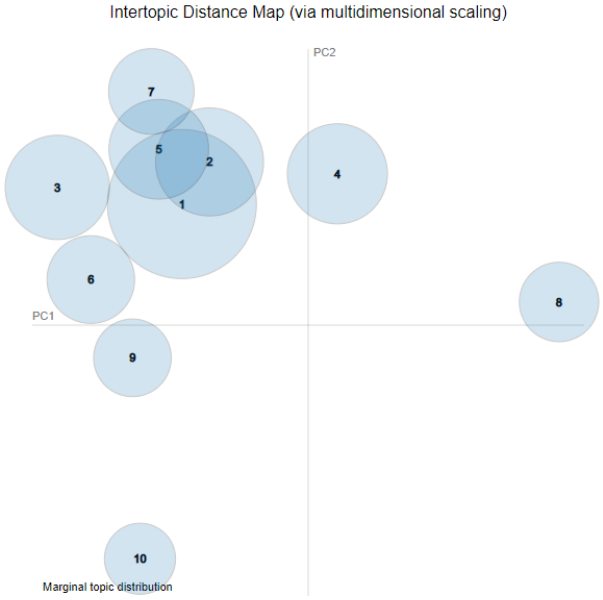
Due to the limit of the paper, we will use "game" category as an example to show our evaluation procedure. We have 6 clusters for "game" category, and there are 3 cluster which show clear distinction in preference. We only use these clusters for evaluation because others are hard to distinguish preference by humans. Based on the centroids, we found that cluster 6 has preference with topic 3 which consists of "character", "fight", "battle", "attack", "enemy" and so on. Then we create some users with reviews showing this preference. Similarly, we have done the same procedure for other

two clusters. In total, we created 20 users with 60 reviews for 3 clusters. After running the whole workflow on them, the percentage of correct clustering is 75% (15 out of 20 users). Among these 5 users, 3 users were clustered to some other clusters than the three that we mentioned above, and 2 users were clustered to one of three clusters but not the correct one.

In analyzing those users which was mistakenly clustered, we found that one cause is that preference shown in their reviews are across different topics. For example, we found one users who mentioned not only mentioned multiple times of "fight" or "battle", but also "multi-play" and "control". These two topics were not clustered together and that's why the clustering algorithms cluster those users mistakenly.

Visual analysis of a various of clusters found that to a human, the differences in some of the clusters seemed to be insignificant. Looking further into the attributes themselves revealed that many of the attributes shared similar words. A graph was created to plot the attributes versus the first and second principal components. Principal components are orthogonal to each other and combine the preexisting variables to create new variables that combine the usefulness of the preexisting variables.

In figure below, the attributes from the cell phone category are plotted using the first two principal components for the category. Multidimensional scaling is used to compress the many dimensions of the attributes. Each circle corresponds to one of the attributes. The size of the circle represents how prevalent that attribute is. There is overlap between four of the attributes and six of the ten attributes all fall in the same quadrant of the graph. We did not expect the attributes to be as close together as they are when looking at the first and second principal components. This proximity between some of the attributes could cause some of the difficulties a person might have in deciphering the differences between some of the clusters.



5 LIMITATION AND FUTURE WORK

First, the evaluation method needs to be further adjusted and improved. Due to the unexpected delay caused by corona virus, we could only evaluate one category. Also, the number of reviews for evaluating is small and may introduce human bias, because the test review writing was done by one person. In the future work, we would plan to rely on some review generator to generate reviews for given keywords. Hopefully this way, we can get rid of bias and has larger test set. In addition, we hope to include more annotators and calculate kappa statistic as the measurement of agreement. This may also allow the attributes to become more dissimilar with each other and allow the chi square test to be utilized in comparing clusters across categories. Furthermore, more metrics and techniques should be introduced. We are planning perform statistical testing and use metrics like F-1 score etc.

Second, instead of using bag of words, we could compute semantic similarity using word vectors, such as those created by the word2vec family of algorithms. This would let us match words based on semantic meaning instead of bag of words. We did not choose word2vec because when calculating the cosine similarity amongst different attributes, the matrix was too large for our computer or online platform (Google Colab) to handle. In the future, we can migrate our model to high performance computing resources such as Rivanna UVa.

Third, we propose to combine our work with more data about products and users. For example, We hope to incorporate users' demographic information. We believe this can help to reveal some pattern associated between attributes and race, age, location and so on. This may be helpful in solving the errors that we talk above in the evaluation section. Besides that, so far in our data set, we only have one column called "asin" which is a unique product ID but do not show any information about the product. We think product information can help us to better understand the attributes in reviews.

Last but not least, we plan to expand our project to implement for comprehensive ranker. We have already proposed a efficient and effective way of recommending reviews by relevance, but our work only gives a set of relevant reviews instead of actual ranking. We are considering adding other factors to determine ranks of reviews, like number of voting which can be a good measurement of helpfulness and trustfulness. Combining relevant data set mentioned above, we could construct user profiles not only on relevance, but also on other aspects. A comprehensive ranker covering all such aspects is the future direction of our work.

6 CONCLUSION

Generalized review recommending with attribute matching studies user's preference towards certain entities based on reviews. This paper focuses on the problem of recommending customized user reviews to individual users. Online product reviews from Amazon.com are selected as data used for this study. The recommendation algorithm has been proposed in workflow, alongside with a detailed description of design and execution of each step. Evaluation on the assumption that people hold different preference of

attributes for products under different categories is performed using proportional test, and evaluation on recommending the review with attribute matching to multi-category website by applying clustering algorithm is performed using manually generated reviews for clustering. Despite limitations, we think our model is effective for providing customized reviews for individual users on multi-category website. [6]

REFERENCES

[1] Avneesh Saluja Christopher Mitcheltree, Skyler Wharton. Using aspect extraction approaches to generate review summaries and user profiles. *Association for Computational Linguistics*, 3:68–75, 2018.

[2] Chengxiang Zhai Hongning Wang, Yue Lu. Latent aspect rating analysis on review text data: a rating regression approach. *KDD '10: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, page 783–792, 2010.

[3] Nikos Mamoulisb Wenting Tu, David W.Cheung. More focus on what you care about: Personalized top reviews set. *Neurocomputing*, 254:3–13, 2017.

[4] Bin Zhou Aiping Li Yan Jia Xiang Zhu, Jiuming Huang. Real-time personalized twitter search based on semantic expansion and quality model. *Neurocomputing*, 254:13–21, 2017.

[6] Our work and codes can be found at https://github.com/walkianm/InfoRetrieval