Latent Aspect Rating Analysis Implementation on Yelp

Yue Hu
Department of System Information Engineering
University of Virginia
Charlottesville, Virginia, 22903
yh7bp@virginia.edu

Yuting Wang
Department of Computer Science
University of Virginia
Charlottesville, Virginia, 22903
yw4hy@virginia.edu

Yin Zhang
Department of Statistics
University of Virginia
Charlottesville, Virginia, 22903
yz4an@virginia.edu

ABSTRACT

Data redundancy is one of the biggest challenges we face in Yelp review of restaurants, one of the largest communication based restaurant platform in the world. As current review format only provide overall rating, which might not reveal the true information of every aspect of that restaurant, we proposed aspect rating at the level of topical aspect to be implemented to return a non-biased rating environment where user can focus on the preferred area of their own rather than influence by the coarse overall rating.

To achieve this level of availability, a regression based two-stage probabilistic rating algorithm, named Latent Aspect Rating Analysis(LARA), is implemented combined with aspect set testing experimentation, proving that even the slightest preference of user comment has significant impact on overall judgement. Consequently, embedded the algorithm inside normal retrieval system, a relatively subjective Yelp review synchronised search engine is designed and contributed to the open source community.

Categories and Subject Descriptors

H.3.3 [**Information Retrieval**]: text mining

General Terms

Algorithm, Yelp

Keywords

Review mining, bootstrapping, aspect rating, sentiment analysis

1. INTRODUCTION

The development of Internet and varies applications on the smart phones, has introduced us a brand new perspective of life built on the communication between individuals. As a result, gloried the communication platforms in every subarea, consider, the most popular restaurant communication tool in the US- Yelp, or the other version in China named Public-Comment. All these kind of software around the world shares a common characteristic: offer useful reviews which could be referred to by others before making their own judgement, for, like whether go to the the specific restaurant or not. This review data, not only is important for users to refer, but also for those business man to improve their own services. However, as the growing usage of Yelp, the review format no longer fit this big data century. Since there is only one overall rating for every restaurant, so if the user want to have a specific knowledge on their preferred section, they have to go through the reviews one by one, which is time consuming for great volume of reviews.

If we assume that the user got time to go over the numberless reviews, they can not have a precise judgement on others' tone of comment based on various background. For example, one may regard the restaurant having a 'poor' service comparing to Michelin-starred restaurants, while others may comment it as 'great' comparing to self-service. And also, the overall rating can not be simple an average of those sub-aspect rating, instead, a regression model is embedded in our algorithm in consideration of the relationship between those sub-aspects. Consider, a user may comment the same quality of service worthwhile for a meal of 10 dollar per person, while relatively unsatisfying if it cost 100 dollars per person. Therefore, while rating an review, it is more precise to reveal the real rating of the restaurant while putting different weight on different aspect.

Actually, there were several existing work on the LARA implementation, however, aimed at analysis of hotel reviews on TripAdvisor, which is totally different from the rating aspect of restaurants in Yelp, which owns both mobile embedded system and website services, making it more difficult to calculate the huge amount of synchronised data in different platforms. Therefore, we come up with an novel experiment, comparing the rating of a same restaurant with different aspect set. Consider, an restaurant ranked first when putting great weight on 'taste' aspect in the review while ranking the last when 'service' is regarded as the most important among the aspect. Obviously, the aspect set testing experiment will give us the best match of aspect words with aspect weights as well as the most concise aspect words, consequently more precise rating on various restaurants, other than just randomly going through the reviews and make up the aspect set which may cause great bias and influence the user experience. Additionally, since the previous works were built on JAVA, an open source Python version of rating platform will also provide an easier way for users other than developer to make an improvement or modification on.

To address the problem of the existing mining systems, we proposed a regression model based two-stage rating approach, specifically analysis the data crawled from the synchronized platform of Yelp. This platform provides an overall pro-

cess of information retrieval from crawling, indexing, analyzing, ranking, all the way to an well-organized UI interface display with as concise Python and .NET code as possible. (All codes are distributed on GitHub:`github.com/yhu04/CS6501-Final_Project.git`, the Python package [2] and [3] are used in this implementation). First of all, the test result with the raw data set crawled from Yelp app and website is not as satisfiable as expected, so several indexing and document analyzing step is addressed for the collection of a well-turned data set specifically for our system usage. Then, ranking was performed by mainly two stages: The first stage of our model is to identify the aspect set and segment reviews by bootstrapping, where aspect set experiment is implemented to test the fitness of those aspect set. The second stage is to come up the aspect rating by Latent Rating Regression(LRR) based on the review content. Finally, an user-friendly interface was presented maximizing the convince of users.

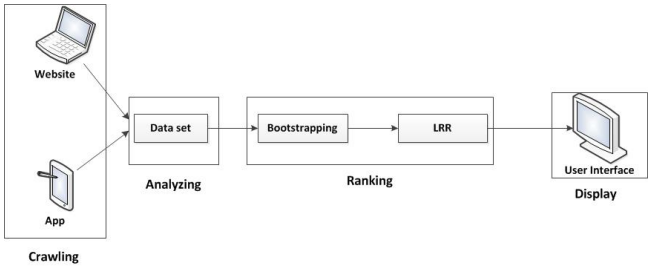

Figure 1: Basic Work flow

The paper is structured as follows. Section 2 presents the problem definition and Section 3 presents the related work. Section 4 presents the system design of method and Section 5 describes the experiment result. Section 6 concludes the paper and Section 7 addressed the future work.

2. PROBLEM DEFINITION

We want to design an algorithm to obtain ratings of different aspects. There are two factors that can be considered for the aspect analysis, overall rating and reviews. Thus, basically, we can build a connection among overall ratings, aspect ratings and reviews. Afterwards, a user interface with generated rating results can be created.

The input for the algorithm are the reviews from yelp and the overall rating of a restaurant. Reviews are composed by sentences and each sentence is possible to describe an aspect of this restaurant. The primary challenge is to extract the aspects from the reviews and overall ratings. With this information, an algorithm for generating aspects and labelling the reviews that are related to aspects needs to be defined. After that, a predictive algorithm can be created to calculate aspect ratings for different restaurant.

3. RELATED WORK

The concept of aspect analysis is not new. It is proposed by [4]. In this work, the authors mentioned the two-stage

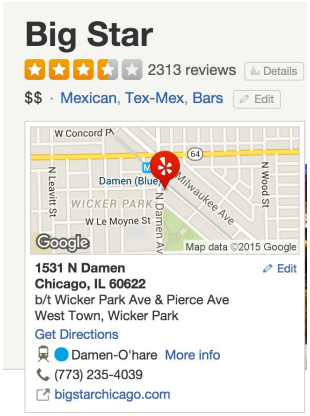

Figure 2: A sample overview of a restaurant on Yelp

Figure 3: A sample review of a restaurant

algorithm generate aspect ratings. The first step is aspect segmentation. Firstly, they manually specify the aspects that are related to reviews and overall ratings. By using bootstrapping method, they can label sentences with different aspects and generate seed words for aspects. The second step is Latent Rating Regression Mode. They use the concept of Bayesian regression and maximum likelihood to estimate parameters. The novel work solve the problems mentioned above. In our work, we implement the algorithm in Python and use the data set from yelp for aspect rating of restaurant review.

4. METHODS

In this paper, we implemented LARA proposed by [4] on the yelp review data in Python. As pointed out in [4], there are two challenges in solving the LARA problem: one is that we do not have detailed supervision about the latent rating on each aspect, and the other is that it is difficult to discover the relative weight placed by a reviewer on each aspect. To solve these challenges, a two-phrase method has been proposed: the first step is to use bootstrapping method to extract the aspect keywords given a list of pre-specifed seed words for each aspect; then we use Latent Rating Regression (LRR) model to tie the latent ratings and latent weights with the review and overall rating. It is assumed that the overall rat-

ing is generated as follows: the reviewer first rates on each aspect by using a weighted combination of the words in the review that discusses the corresponding aspect, then his/her overall rating is determined by another weighted combination of his/her all aspect ratings.

4.1 Aspect Segmentation

The idea of aspect segmentation in [4] is to map the sentences in a review into subsets corresponding to each aspect. Hence it is required to assign some keywords to each aspect in the beginning, then the bootstrapping algorithm can be applied to obtain more related words for each aspect based on the dependency between words and aspects.

Specifically, the basic workflow of the bootstrapping algorithm can be described as follows: given the seed words for each aspect and all the review text as input, we assign each sentence to the aspect that shares the maximum term overlapping with this sentence; based on this initial aspect annotation, we calculate the dependencies between aspects and words by Chi-Square χ^2 statistic, and include the words with high dependencies into the corresponding aspect keyword list. These steps are repeated until the aspect keyword list is unchanged or the number of iterations exceeds the limit. The full description of the algorithm is in Algorithm 1. The χ^2 statistic to compute the dependencies between a term w and aspect A_i is defined as follows:

$$\chi^2(w, A_i) = \frac{N \times (A \times D - B \times C)^2}{(A + C) \times (B + D) \times (A + B) \times (C + D)} \quad (1)$$

where A is the number of times w occurs in sentences belonging to aspect A_i, B is the number of times w occurs in sentences not belonging to A_i, C is the number of sentences of aspect A_i that do not contain w, D is the number of sentences that neither belong to aspect A_i, nor contain word w, and N is the total number of word occurrences.

After aspect segmentation, for each review d, we would get $K \times N_d$ matrix W_d, where N_d is number of unique words in review d. The $(i, j)th$ element in W_d is the frequency of word w_j assigned to aspect A_i in the review d normalized by the total counts of words in that aspect.

4.2 Latent Rating Regression Model (LRR)

The second stage of LARA is Latent Rating Regression Model (LRR), which is able to tie the latent ratings and latent weights with the review and overall rating.

4.2.1 Assumption of Review Generation Process

Every review is assumed to be generated as follows: first, the review decides the aspects she wants to comments on; the for each aspect, the reviewer chooses the words to express her own opinions. Then the review forms a rating on each aspect based on the sentiments of word she used on that aspect. Finally, the overall rating is decided by a weighted sum of aspect rating.

4.2.2 The LRR Model

The LRR model treats the feature matrix of each review W_d, which is obtained after aspect segmentation, as independent variables and the overall rating r_d as response variable. In order to model the latent aspect rating s_d and latent aspect weight α_d, the LRR further assume that $-$ is determined aspect rating s_d with different weight α_d, and the aspect rating s_d is determined by W_d with term sentiment weight β. Specifically, the aspect rating is definitely determined by a linear combination of W_d

$$s_{di} = \sum_{j=1}^{N_d} \beta_{iw_j} W_{dij} \quad (2)$$

where β_i is the word sentiment polarities on aspect A_i, and the overall rating follows a Gaussian distribution with mean $\alpha_d^T s_d$ and variance δ^2,

$$r_d \sim N(\alpha_d^T s_d, \delta^2) \quad (3)$$

Thus, from (2) and (3) we can see that the idea of LRR is to bridge the gap between the observed overall rating r_d and the detailed text descriptions W_d through introducing the latent aspect weight α_d and term sentiment weight β, which enable us to model the overall rating based on ratings of specific aspects. Since there is no uncertainty in aspect rating s_d if β is given, we can plug in (2) to (3), thus, we only leave α_d as latent variable.

Further, in order to model the dependency between aspects, we impose a multivariate Gaussian distribution on α_d as its prior distribution, i.e.

$$\alpha_d \sim N(\mu, \Sigma) \quad (4)$$

where μ and Σ are the mean and variance parameters. So far, we have finished formulating the whole LRR Model, whose graphical representation is shown in Figure (), where r_d and W_d are observed data, $\Theta = (\mu, \Sigma, \delta^2, \beta)$ are the set of corpus-level model parameters, and α_d is the latent aspect weight for review d.

Algorithm 1 Aspect Segmentation

Input: A collection of reviews $\{d_1, ..., d_{|D|}\}$, set of aspect keywords $\{T_1, ..., T_K\}$, vocabulary V, selection threshold p and iteration step limit I
Output: Reviews split into sentences with aspect assignments
Step 0: Split all reviews into sentences, $X = x_1, x_2, ..., x_M$;
Step 1: Match the aspect keywords in each sentence of X and record the matching hits for each aspect i in $Count(i)$;
Step 2: Assign the sentence an aspect label by $a_i = \arg\max_i Count(i)$. If there is a tie, assign the sentence with multiple aspects.
Step 3: Calculate χ^2 measure of each word (in V);
Step 4: Rank the words under each aspect with respect to their χ^2 value and join the top p words for each aspect into their corresponding aspect keyword list T_i;
Step 5: If the aspect keyword list is unchanged or iteration exceeds I, go to **Step 6** , else go to **Step 6**;
Step 6: Output the annotated sentences with aspect assignments.

4.3 LRR Model Estimation and EM Algorithm

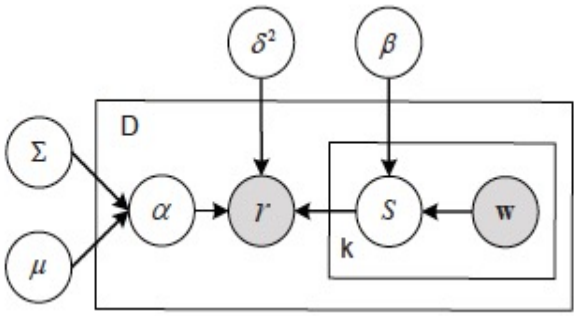

Figure 4: Graphical Representation of LRR.

The probability of observed overall rating in a given review can be written as

$$\begin{aligned} p(r|d) &= p(r_d|\mu,\Sigma,\delta^2,\beta,W_d) \quad (5)\\ &= \int p(\alpha_d|\mu,\Sigma)p(r_d|\sum_{i=1}^{k}\alpha_{di}\sum_{j=1}^{N_d}\beta_{iw_j}W_{dij},\delta^2)d\alpha_d \end{aligned}$$

Obviously, it is intractable to obtain the maximum likelihood estimation of Θ by directly taking derivative of (5) and setting it to zero because of the presence of α_d. Hence, we use the EM algorithm to solve this problem where in the E step, we obtain the estimation of latent variables α_d, given Θ; while in the M-step, we maximize the likelihood to obtain the estimation of Θ, given the α_d from E step.

Specifically, in the E step, given Θ, we obtain the estimation of latent variables α_d, by maximizing its posteriori likelihood function

$$\begin{aligned} P(d|r,\mu,\Sigma,\beta,\delta^2,W) &\propto P(d,r|\mu,\Sigma,\beta,\delta^2,W)\\ &= p(\alpha_d|\mu,\Sigma)p(r_d|\alpha_d^T s_d,\delta^2) \end{aligned}$$

which will lead to

$$\hat{\alpha}_d = \arg\max_{\alpha_d}[-\frac{r_d-\alpha_d^T s_d}{2\delta^2}-\frac{1}{2}(\alpha_d-\mu)^T\Sigma^{-1}(\alpha_d-\mu)] \quad (6)$$

subject to

$$\begin{aligned} \sum_{i=1}^{K}\alpha_{di} &= 1 \quad (7)\\ 0 \le \alpha_{di} &\le 1 \text{ for } i=1,...,K \quad (8) \end{aligned}$$

In the M step, given α_d from E step, we maximize the log-likehood of all overall rating

$$L(D) = \sum_{d\in D}\log p(r_d|\mu,\Sigma,\beta,\delta^2,W)$$

which will lead to

$$\begin{aligned} \hat{\mu} &= \arg\max_{\mu} -\sum_{d\in D}(\alpha_d-\mu)^T\Sigma^{-1}(\alpha_d-\mu) \quad (9)\\ &= \frac{1}{|D|}\sum_{d\in D}\alpha_d \quad (10)\\ \hat{\Sigma} &= \frac{1}{|D|}\sum_{d\in D}(\alpha_d-\hat{\mu})(\alpha_d-\hat{\mu})^T \quad (11)\\ \hat{\delta}^2 &= \arg\max_{\delta^2}\left[-|D|\log\delta^2-\frac{\sum_{d\in D}(r_d-\alpha_d^T s_d)^2}{\delta^2}\right] \quad (12)\\ &= \frac{1}{|D|}\sum_{d\in D}(r_d-\alpha_d^T s_d)^2 \quad (13)\\ \hat{\beta} &= \arg\max_{\beta}\frac{\sum_{d\in D}(r_d-\sum_{i=1}^{K}\alpha_{di}\beta_i^T W_{di})^2}{\hat{\delta}^2} \quad (14) \end{aligned}$$

The E step and M step are repeated until the likelihood value of overall rating converges.

5. EXPERIMENT RESULTS

5.1 Data Processing

We use the data set [1] for MP1 to do the experiment. There are 32048 restaurant reviews from Yelp. After checking reviews on restaurants, we decide to choose 5 aspects that can represent the information reviewers want to deliver. They are, environment, service, value, taste (meat), and taste (other than meat).

We perform the pre-processing on reviews. First of all, we transfer all the reviews to lowercase. And then we use a general stop-word list to remove stop words in the reviews. We use the package from python to do stemming.

Some words may have similar meaning. For example, price, expensive, cheap, or worth all can describe the aspect of value. Thus, we develop a seed word list for each aspect for aspect segmentation algorithm. We set the threshold to be 5 and iteration limit to be 10. The table shows the initial seed words we put.

Aspect	Seed words
value	recommend, price, quality, worth, food
environment	locate, parking, atmosphere, place, room
service	service, minutes, wait, staff, order
taste(meat)	pork, chicken, beef, rib
taste(other than meat)	shrimp, oyster, egg, sauce, bread

Table 1: Aspect with Initial Seed Words

After the bootstrapping algorithm, we get 20 seed words for each aspect. Below is the example of service aspect.

service, minute, wait, staff, order, service, table, hour, friendly, long, seat, us, reserve, line, time, told, waiter, hostess, arrive, party, server, ask, took, take, attend, waitress, mean, flavor, ready, 45, best, 30, get, sit, first, help

Some sentences are not related to any aspect and fail to

be labeled. For those sentences, we choose to discard. After the pre-processing, we get the word frequency matrix and overall ratings for Latent Rating Regression. Below are the statistics of the processed data set.

Number of Restaurants	18
Number of Reviews	32048
Number of Annotated Reviews	32048
Number of Sentences per Review	6.23 ± 4.82

Table 2: Statistics of the Data Set

5.2 Aspect Segmentation Comparison

Even though an overall rating can be a criterion to judge the performance of a restaurant. Reviewers may have diverse preference on restaurants. Also restaurants want to know the satisfaction level from customers. This LLR is capable of meeting demand above. We choose two restaurants with the same overall rating to compare the difference on aspect ratings. After running the predictive model, we obtain the results below.

Restaurant	1	2
Overall Rating	4	4
Environment	4.3	3.9
Service	3.7	3.5
Value	3.5	4.2
Taste(meat)	3.8	4.4
Taste(other than meat)	4.7	4.1

Table 3: A sample aspect rating

Even though these two restaurants obtain the same overall ratings but they perform well on different aspects. The first restaurant is better on environment while the second restaurant is popular because of their taste on meat. This is valuable information for restaurants to improve their satisfaction. Also, it is helpful for the following viewers with different needs.

LLR may be sensitive on parameter setting. Since the aspect segmentation is the crucial part of the two-stage algorithm. We decide to test impacts on the changes of number of aspects and content of aspects. We combine the 5 aspects into 4 aspects. They are service, environment, value, and taste. Also we change the seed words to see whether the initial selection will affect the scores.

Restaurant	1	2
Overall Rating	4	4
Environment	4.2	3.9
Service	3.7	3.6
Value	3.5	4.2
Taste	4.4	4.4

Table 4: A sample aspect rating

After running the new experiment, we find that the aspect ratings do change a lot. Since we combine two aspect, taste(meat) and taste(other than meat) into taste and give a new list of seed words toward it, the ratings definitely change. The first restaurant right now performs as well as the second restaurant on taste. If we check other aspects, changes exist as well. The ratings of environment aspect both are altered on the first and second restaurant.

5.3 User Interface

A user-friendly website is created for result display purpose (see figure 5). The front page is design with .Net framework in Visual Studio 2015 combined with Python Tools and connected to the back-end SQL database(could upgrade to MangoDB is processing big data). One of the things we tried but couldn't complete was to input random sentence which do not contain any of the seed word we produced. It might be better if the sentiment analysis result of query could match to the seed word of aspects, however, We have tried a couple of natural language tool-kits but matching sometime fails with low frequency words currently.

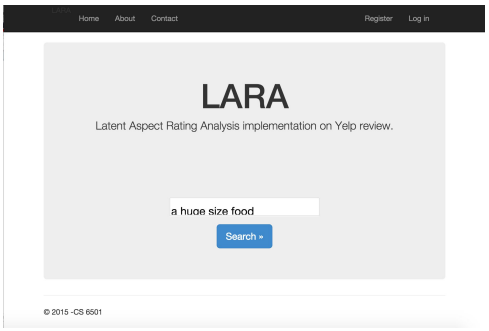

Figure 5: Home page where user input query

We can see from figure 6, when entering the query"a huge size food", our system ranks the restaurant by the 'value' aspect, other than simply rank it according to overall rating. However, other rating is also available if the user would like to reference to. If the user preferred aspect rating is the same, our system would rank the one with highest overall rating, if all the requisition are the same then result allow random selection.

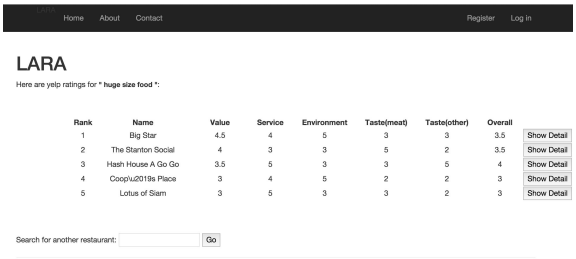

Figure 6: Ranking result

Simple click on "show detail" will allow user have a view of the brief restaurant introduction and a provided redirection to for more details listed on Yelp website.

6. CONCLUSIONS

This paper described the implementation of LARA on Yelp reviews, used for reducing the browsing time for user by

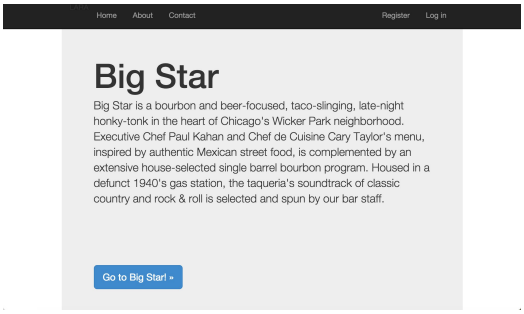

Figure 7: Restaurant detail

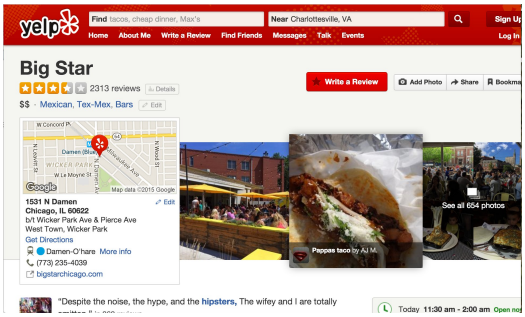

Figure 8: Redirection to Yelp

pruning the redundant data set to a single rating over each selected aspects, which user can directly refer to before making decision. Our system takes Yelp reviews as input data and then performs the pre-processing as the normal information retrieval system, while adding a creative LARA algorithm for topical aspect rating, where the overall rating is based on every sub-aspect rating by regression model. Consider that the slight different of user preference of a specific area might greatly influence the overall rating, we have achieved a relatively subjective search engine both from the review side and query side, providing user a more precise ranking of restaurants on their desired aspect.

Proven by our novel aspect testing experiment, our system has provided the desired levels of availability and performance and has been successful in handling various input queries and review contents. But when talking about aspect sensitivity, the LRR model heavily relies on the selection of aspects. Since in this project, we manually specify the number of aspects to 5 and choose the related seed words. It is an subjective process that may raise bias. Also, after running the aspect segmentation algorithm, we find some possible stop words are in the seed word list. Thus, removing stop words that are specific for restaurant reviews is needed for the further work.

7. FUTURE WORK

In this work, there is underlying assumption that the documents are independent of each other, which is usually violated in real world. Information such as user-user relationship is not utilized in this work. In the future work, we will consider incorporate the user-user relationship to improve the prediction of latent aspect rating, for the reason that users who are friends of each other are likely to have more similar attitude to the same aspect.

In the part of aspect segmentation, we specify 5 aspects and the results are based on the aspects we specify. It is possible that the aspects we specify can't accurately demonstrate aspects in the reviews. Even though from the experiment, the reasonable results are obtained, we still have hard time on deciding aspects. It is better for us to implement an aspect segmentation algorithm with specifying any apse ct. The algorithm can purely generated aspects that are driven by reviews.

In the implementation of parameter estimation, we have problems on β optimization since the optimization of β includes the high-dimensional matrix calculation. It may be the limitation of programming language we use. Moreover, the optimization method we choose is the main reason that limit the speed. Thus, a new optimization algorithm can be tried in the later work.

The user interface we build actually rely on a structured database. Every time when users want to type in queries, they must choose the seed words to obtain the ranking of an aspect. Since our ultimate goal is to build a aspect analysis information retrieval system, an algorithm that can do text mining on the relationship between queries and aspects is needed. The algorithm can firstly know that what kind of aspects the queries refer and obtain the data from database to generate rankings for specific queries.

8. REFERENCES

[1] Yelp review dataset from mp1. [Online; accessed 2015-12-16].

[2] S. Bird, E. Klein, and E. Loper. *Natural language processing with Python.* " O'Reilly Media, Inc.", 2009.

[3] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed 2015-12-16].

[4] H. Wang, Y. Lu, and C. Zhai. Latent aspect rating analysis on review text data: a rating regression approach. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 783–792. ACM, 2010.