Yelp Review Analysis and Recommendation

Enrui Liao, Yuqing Guan, Ying Tan, Mengting Wu
Department of Computer Science
Columbia University
{el2756, yg2392, yt2443, mw2987}@columbia.edu

Abstract—**Review analysis has become a critical reference in recommendation and business strategies nowadays. Exploration into the feedbacks of the users can grant us incredible insights. Given such untapped treasure of resources, we aim at harnessing the fusion of the review analysis and recommendation, and try to extract valuable advice for business management.**

Applying the review dataset from Yelp Dataset Challenge, we develop a model of recommendation for customers with the explicit industrial classification, as well as the implicit subtopic clustering by Latent Dirichlet Allocation. Based on the information presented in the review analysis, therefore, we are able to provide recommendations for customers with related businesses, taking multiple facets of their preferences into account.

Besides, in this project, we also extract key factors that customers mainly take interests in for each industry. The keywords extracted from both the positive and negative review analysis can provide significant insights to their business strategies.

Keywords: Part-of-speeach tagging, Classification, Latent Dirichlet Allocation, Clustering, Recommendation

I. Introduction

Currently, the star rating is typically a dominant factor for the underlying analysis of customer recommendation mechanism. However, a great loss of information is inevitable if we merely take a single rating into account in terms of recommendation, since the focus of each customer varies a lot from each other. Besides ratings, users' review text is also a rich treasure of feedbacks. Unfortunately, traditional methods applied in general simply discard the review texts, which leave out many latent factors unrevealed. By delving into the content of the reviews, we can interpret incredible abundance of useful information. And taking advantage of that, it enables us to provide more accurate recommendations for customers based on each facet of the subtopics. And the key factors absorbed from the reviews can be of great significance for the management of different industries.

In this project, therefore, we aim at achieving both the customer-oriented and business-oriented goals. Based on the review analysis, we recommend businesses to customers by taking both explicit industrial categories and implicit subtopics from LDA clustering into consideration, and also try to offer commercial advice for business management by keyword extraction.

II. Related Works

By far, tremendous work has been done related to review analysis and recommendation. However, many works only focus on the rating stars of reviews. Based on the quantified rating information, researchers can easily analyze users' interests and recommend corresponding items to them. The most widely used method is collaborative filtering [1], which employs the rating information to infer users' common interests or experiences. With these inferences, developers can build user-based recommenders and item-based recommenders [2].

Compared to the quantified rating stars, users' review text is relatively ill-structured. Before using it to provide recommendations, researchers have to retrieve hidden topics of industrial categories or ratings from these reviews. Therefore, topic extraction is necessary for the review-based recommendations. According to the explicitness of topic labels, the topic extraction can be roughly divided to classification and clustering. The former one employs explicit labels to generate a model from training data set and then apply it to new testing data [3]. Some widely used classification algorithms are naïve Bayesian classifier [4], support vector machine [5] and k-nearest neighbors algorithm [6].

As for the implicit topic labeling, researchers generally apply the unsupervised learning algorithm of clustering for retrieval. Traditional methods usually use a feature space model, which maps documents to vectors in a very high-dimensional space. These algorithms can be further divided to centroid-based clustering, like k-means [7], density-based clustering, such as DBSCAN and OPTICS [8], and distribution-based clustering. Apart from these vector-based algorithms, researchers also conceived probabilistic models to perform clustering. Latent Dirichlet Allocation is a popular model which describes a document as a mixture of topics [9]. For short text, like posts on social networks, comments and reviews, scientists further conceived models for these special types of model, such as Twitter LDA [10].

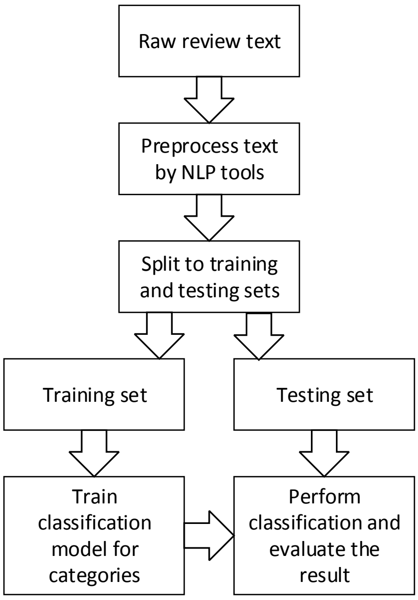
III. SYSTEM OVERVIEW

1. Structures of our systems

Our work consists of two parts:

a) Customer-oriented review analysis and recommendation:

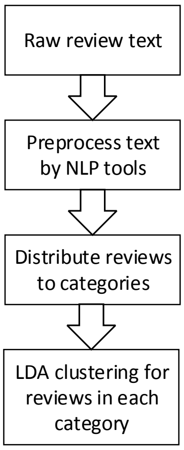
i. Category-based classification: classify review text to explicit industrial category labels



Raw review text
Preprocess text by NLP tools
Split to training and testing sets
Training set
Testing set
Train classification model for categories
Perform classification and evaluate the result

Figure 1: Category-based Classification

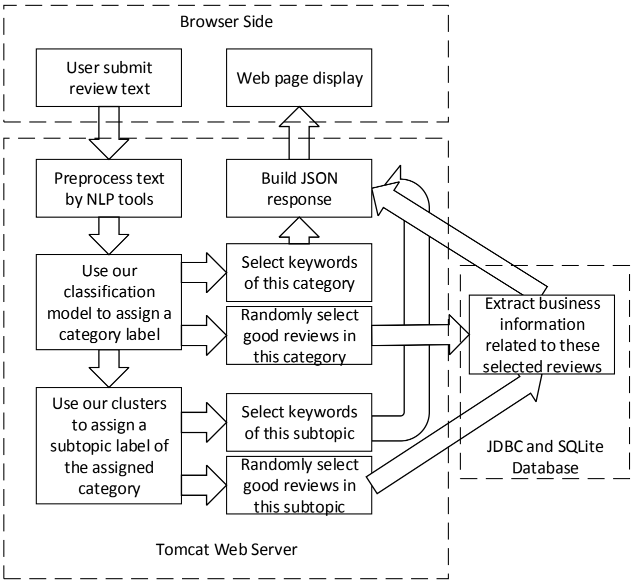
ii. Subtopic clustering: extract implicit subtopic labels from reviews after the classification



Raw review text
Preprocess text by NLP tools
Distribute reviews to categories
LDA clustering for reviews in each category

Figure 2: Subtopic Clustering

iii. Build an online review-based recommendation system based on the results of category-based classification and subtopic clustering

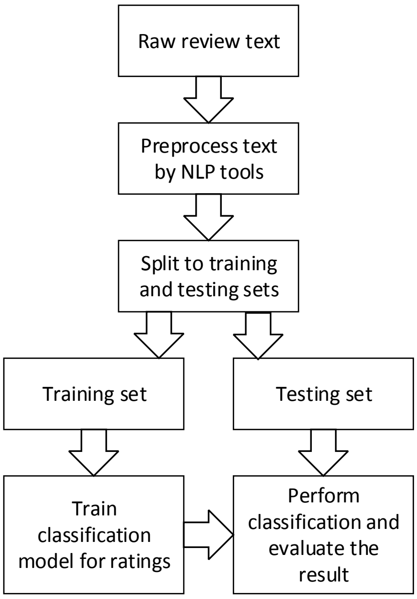


Browser Side
User submit review text
Web page display
Preprocess text by NLP tools
Build JSON response
Use our classification model to assign a category label
Select keywords of this category
Randomly select good reviews in this category
Extract business information related to these selected reviews
Use our clusters to assign a subtopic label of the assigned category
Select keywords of this subtopic
Randomly select good reviews in this subtopic
JDBC and SQLite Database
Tomcat Web Server

Figure 3: Online Recommendation System

b) Business-oriented review analysis:

i. Rating-based classification: classify review text to positive and negative classes by their rating stars



Raw review text
Preprocess text by NLP tools
Split to training and testing sets
Training set
Testing set
Train classification model for ratings
Perform classification and evaluate the result

Figure 4: Rating-based Classification

ii. Extract keywords of positive reviews and negative reviews in each category

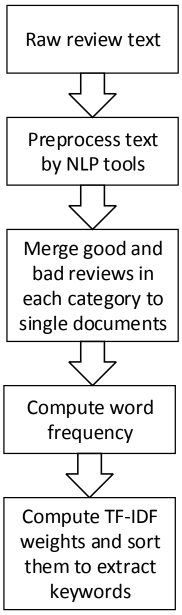

Raw review text
Preprocess text by NLP tools
Merge good and bad reviews in each category to single documents
Compute word frequency
Compute TF-IDF weights and sort them to extract keywords

Figure 5: Keyword Extraction

2. Dataset and preprocessing

We apply our research with the data from Yelp Dataset Challenge, which includes data from Phoenix, Las Vegas, Madison, Waterloo, Edinburgh and etc., covering 42,153 businesses, 320,002 business attributes, 31,617 check-in sets, 252,898 users, 955,999-edge social graph, 403,210 tips and 1,125,458 reviews [11]. All the data is provided in the form of separated JSON objects, where the business object includes information about the type of business, location, rating, categories, business name, and a unique id, while the review object contains a rating, review text, and is associated with a specific business id and user id. Considering the relative size of the data set, we reduce the size and extract 37,326 reviews for business in 10 industrial categories:

- Arts & Entertainment
- Automotive
- Beauty & Spas
- Chinese
- Event Planning Services
- Grocery
- Hotels & Travel
- Night Life
- Restaurants
- Shopping

We mainly deal with review and business objects within the Restaurants category as our prototype.

For preprocessing, we use Natural Language Processing tools, Stanford CoreNLP [12], to analyze the raw Yelp review text. Firstly, we perform sentence segmentation on the text to get separated sentences. We then tokenize the sentences and lemmatize the tokens so that each review can be transformed to a set of standardized words.

Together with the lemmatization, we also use the part-of-speech (POS) tagger provided by Stanford CoreNLP to assign POS tags to each word. These tags represent the roles of corresponding words in the sentences. For example, an adjective word can usually express the customer's emotional attitude to a business. The adjective word 'good' from a review like 'This restaurant provides good food' is critical for us to judge that this review is a positive review. In the other hand, a noun can imply the industrial category of a review. In the aforementioned review, the word 'food' implies this review is related to the 'Restaurants' category. Contrarily, some parts of speech are less important when we try to extract information from reviews, such as possessive pronouns: 'my', 'your', 'his'. We can just remove these words from the sentences to reduce redundant information.

We also planned to perform syntactic analysis on these reviews. We use Stanford Parser included in Stanford CoreNLP tools to build the dependency syntax tree of the segmented sentences. However, the dependency parsing is extremely time-consuming. It took more than 4 days to do dependency parsing on the whole data set. We also find that the syntactic roles of words are very similar to their POS tags. For example, an attribute is usually an adjective word, and most adverbs are also adverbial modifiers in sentences. It makes little sense to analyze the syntax tree of sentences when we have already assigned parts of speech to the words in the review text. Therefore, we find that it makes little difference to employ syntactic analysis and we decide not to cover such operation.

Besides using automated NLP tools, we also preprocess the review text manually. We use a stop word list to remove all stop words from our parsed text. These stop words appear frequently in different reviews but are relatively useless for our review-based recommendation and analysis. Furthermore, we only keep digits, letters and whitespaces in the review text and remove all other special characters. We also tried to use an emotional dictionary to enhance reviewers' positive and negative attitudes. However, it turned out to have little influence on the result of our analysis, which is proved to be unnecessary to be included.

IV. Algorithm

1. Classification algorithm

For category-based and rating-based classifications, we use naïve Bayesian classifier to assign explicit labels to reviews. The naïve Bayesian classification is a generative probabilistic model, which assumes that, for the position of each token, the probabilities of words are independent and identically distributed. Each class has a set of parameters θ , from which we can compute the likelihood $p(X \mid \theta)$ representing the probability of generating a document from the corresponding class. Combining the likelihood with the prior probability $p(\theta)$, we can figure out the posterior probability by the Bayes rule:

$$p(\theta \mid X) = \frac{p(X \mid \theta)p(\theta)}{p(X)}$$

By comparing the posteriors we can determine the best class for the given text document:

$$\begin{aligned} \theta^* &= \arg\max_\theta p(\theta \mid \mathrm{X}) \\ &= \arg\max_\theta \frac{p(X \mid \theta)p(\theta)}{p(X)} \\ &= \arg\max_\theta p(X \mid \theta)p(\theta) \end{aligned}$$

When computing the likelihood $p(X \mid \theta)$, there are two models. The first one is the Bernoulli model, which only checks whether a word appears in one document. x_d indicates whether word d appears in document X .

$$p(X \mid \theta) = \prod_{d=1}^{D} p(W_d \mid \theta)^{X_d} \left(1 - p(W_d \mid \theta)\right)^{(1-X_d)}$$

This model works well for documents with a small glossary. But our reviews include many words, so we use the multinomial model, which also takes the frequencies of words into consideration. x_d is word d 's frequency in document X [4].

$$p(X \mid \theta) = \frac{\left(\sum_{d=1}^{D} X_d\right)!}{\prod_{d=1}^{D} X_d!} \prod_{d=1}^{D} p(W_d \mid \theta)^{X_d}$$

When learning a parameter $p(W_d \mid \theta)$ of the multinomial model, we use the following formula to compute this probability. In case that some words never appear in documents of one class, we add one to the count of each word.

$$p(W_d \mid \theta) = \frac{\sum_{i=1}^{N} x_{id} + 1}{\sum_{d=1}^{D} \sum_{i=1}^{N} x_{id} + D}$$

We use Hadoop and Mahout to perform offline experiments using the naïve Bayesian classification algorithm. And when constructing our online review-based recommendation website, due to the difficulty of deploying Hadoop and Mahout on our Windows-platform server, we implement a naïve Bayesian classifier from scratch.

2. Clustering algorithm

After performing the naïve Bayesian classification, we can already find the explicit categories or rating labels for the given raw review text. Nevertheless, we try to find more hidden subtopic information for the reviews in the same category, hence we also do clustering for these reviews.

A clustering algorithm is an unsupervised learning algorithm which does not require an explicit label for the documents. In our experiments, we use Latent Dirichlet Allocation (LDA), which is mature and widely used in text clustering.

Latent Dirichlet Allocation is also a generative model, which uses a set of parameters to describe the generation of one document [9]:

- α : parameter of the Dirichlet prior on the per-document topic distributions
- β : parameter of the Dirichlet prior on the per-topic word distributions
- θ_i : topic distribution for document i
- ϕ_k : word distribution for topic k
- z_{ij} : topic for the j -th word in document i
- w_{ij} : the j -th word in document i

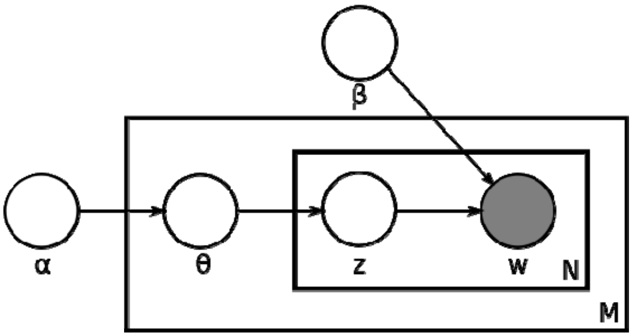


Figure 6: LDA Model [9]

We use JGibbLDA [13] to extract subtopics for reviews in each category. When performing online recommendation, we can use the existing clusters to classify a user's raw text and give an implicit label to this text.

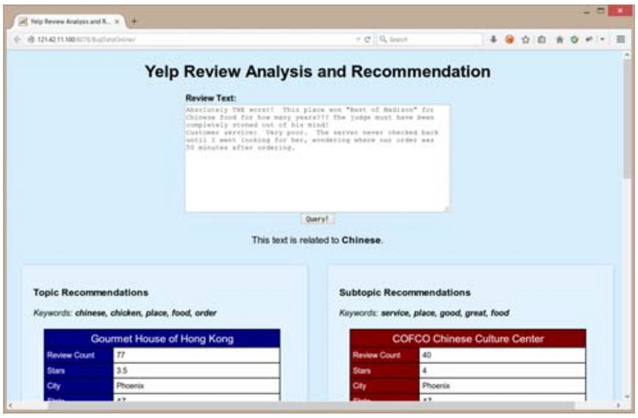
3. Keyword extraction

In our business-oriented task, we want to give advice based on keywords of positive and negative reviews in each category. The basic idea for extracting keywords is to count the frequency of each word and find the most frequent ones. However, for those words that appear in almost all documents, it is highly probable that these words can hardly express any valuable information. For example, the word 'have' can represent possession and is also an auxiliary verb in perfect tenses. It may appear frequently in documents but do not make sense. Therefore, we use TF-IDF weights to deal with this situation, which takes both term frequencies and document frequencies into consideration.

When extracting keywords for positive and negative reviews in each industrial category, we firstly merge all positive reviews in each category into one single document. Similarly, we also merge all negative reviews in each category into one document. We then compute TF-IDF vectors for each merged document. By sorting the weights in each vector, we can find the importance of each word and further select valuable keywords from high-weighted words.

V. Software Package Description

We build our online review-based recommendation tool on a Windows server with Apache Tomcat 7. The address of our server is http://121.42.11.100:8078/BigDataOnline/. Our server is temporary and only available before Jan. 20th, 2015.


Yelp Review Analysis and R...
121.42.11.100:8078/BigDataOnline/
Search

Yelp Review Analysis and Recommendation

Review Text:

```
Absolutely THE worst!  This place won "Best of Madison" for
Chinese food for how many years??? The judge must have been
completely stoned out of his mind!
Customer service:  Very poor.  The server never checked back
until I went looking for her, wondering where our order was
30 minutes after ordering.
```

Query!

This text is related to **Chinese**.

Topic Recommendations

Keywords: ***chinese, chicken, place, food, order***

Gourmet House of Hong Kong	
Review Count	77
Stars	3.5
City	Phoenix
State	AZ

Subtopic Recommendations

Keywords: ***service, place, good, great, food***

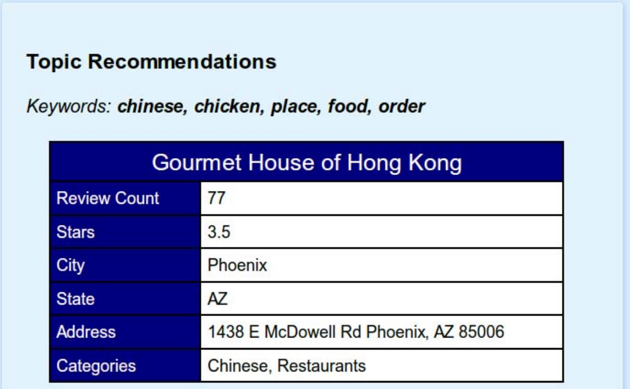
COFCO Chinese Culture Center	
Review Count	40
Stars	4
City	Phoenix
State	AZ

Figure 7: User Interface of Our Website

The screenshot above is the main user interface of our website. A customer can input his or her review text in the text area and then click 'Query!' button to submit this text to the server side.

The review text is submitted to a JSP dynamic page, which firstly use our classifcation model to find an industrial categorial label for this review. Based on this label, it can select keywords of this category and randomly select high-rating reviews to recommend related businesses.

Furthermore, after getting the category, our program will pass the review to the JGibbLDA inferencer, which can infer the most related cluster in the same category. From the cluster, our program can also find keywords and recommend businesses. Finally, the server-side program will package all results of the analysis and recommendation into a JSON object, and return it to the front-end by AJAX. The keywords of the assigned category and one recommended business are shown in the screenshot below:



Topic Recommendations

Keywords: ***chinese, chicken, place, food, order***

Gourmet House of Hong Kong	
Review Count	77
Stars	3.5
City	Phoenix
State	AZ
Address	1438 E McDowell Rd Phoenix, AZ 85006
Categories	Chinese, Restaurants

Figure 8: Displayed Keywords and One Business

Our business information is stored in a SQLite database, which supports common SQL statements to manage and query records. By using the standard JDBC interface, we can easily transport our database to other SQL database management system, such as Oracle, MySQL, Microsoft SQL Server and etc., which ensures the scalability of our database.

VI. Experiment Results

1. Customer-oriented review analysis and recommendation:

a) Category-based classification:

For category-based classification, we use Mahout's naïve Bayesian classifier to classify our parsed review text to 10 categories. We split the text by using 80% of the reviews as the training set and the remaining 20% as the testing set. Then we train the model and evaluate it by the testing set.

The accuracy is higher than 60%. Considering there are 10 classes in this classification, the accuracy is relatively high:

Table 1: Result of Category-based Classification

STATISTICS	
Kappa	0.4563
Accuracy	61.0975%
Reliability	51.7575%
Reliability (standard deviation)	0.2891

b) Subtopic Clustering:

We use JGibbLDA to cluster reviews in each category. Each review can be placed in one of the five subtopics in one category. Due to the limited space in our report, we only show the subtopic proportions for 5 categories:

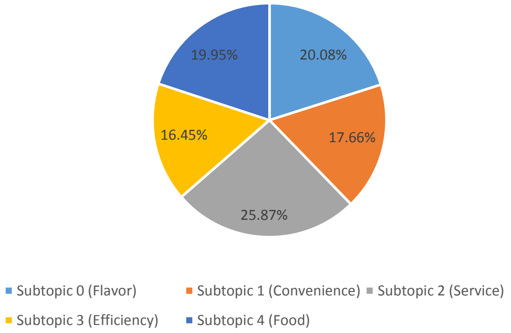


Figure 9: Subtopic Proportions for Restaurants

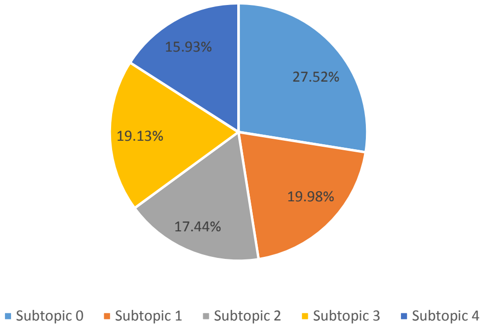


Figure 10: Subtopic Proportions for Arts & Entertainment

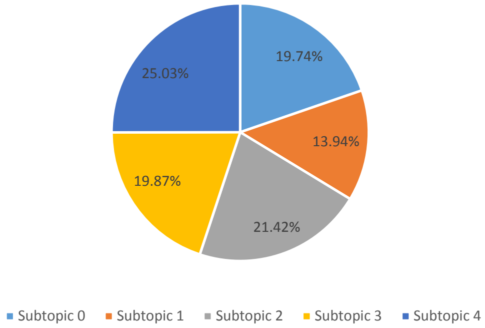


Figure 11: Subtopic Proportions for Automotive

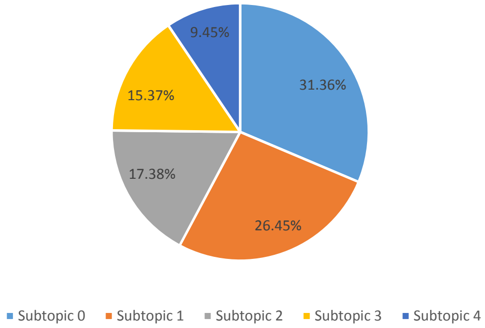


Figure 12: Subtopic Proportions for Event Planning Services

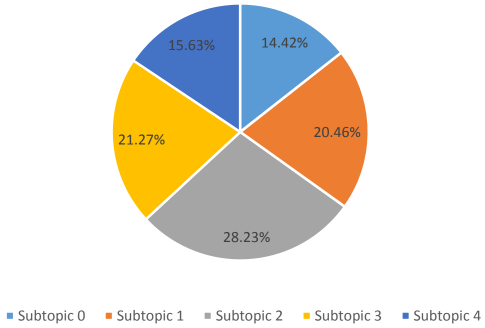


Figure 13: Subtopic Proportions for Grocery

A LDA model also includes the probabilities of words of each hidden topic. Based on the results of LDA clustering, we can sort these probabilities and find keywords for each subtopic of categories. Furthermore, by semantically analyzing these keywords, we can filter out useless keywords and assign meaningful names to the corresponding subtopics.

We take subtopics of Restaurants category as examples:

Table 2: Subtopics and Keywords for Restaurants

SUBTOPIC	KEYWORD	PROBABILITY
Flavor	fresh	1.35%
	hot	1.18%
	taste	0.98%
	selection	0.71%
	delicious	0.63%
Convenience	place	1.64%
	experience	0.94%
	server	0.81%
	order	0.79%
	parking	0.73%
Service	service	3.23%
	time	2.18%
	friendly	1.75%
	staff	1.42%
	atmosphere	1.14%
Efficiency	time	1.34%
	hour	0.98%
	wait	0.79%
	menu	0.76%
	sit	0.75%
Food	chicken	2.28%
	sauce	1.83%
	dish	1.25%
	soup	0.88%
	beef	0.87%

We can find that the reviews for restaurants can be classified to five subtopics, including flavor, convenience, service, efficiency and food. The reviews in the 'service' subtopic will focus on keywords like 'service', 'time' and 'staff', while reviews in the 'food' subtopic will focus on 'chicken', 'sauce' and 'beef'.

2. Business-oriented review analysis:

a) Rating-based classification:

Similar to category-based classification, we use the rating stars as the classes. For a typical Yelp review, the rating star is an integer ranging from 1 to 5. In order to improve the accuracy of classification, we divide the ratings to 2 classes: 1~3 represents a negative review, while 4~5 represents a positive review. We use Mahout's naïve Bayesian classifier to train models for each category and then evaluate them by the testing data sets (20% of the whole data sets).

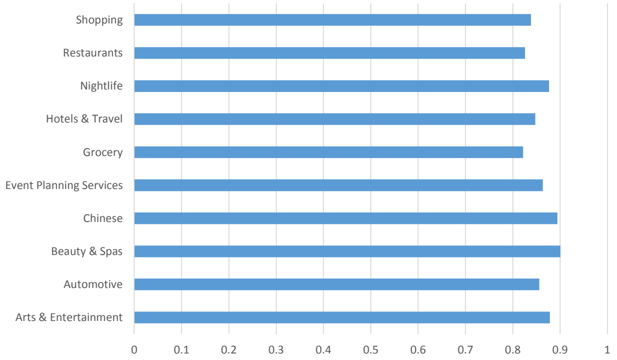


Figure 14: Accuracies of Rating-based Classifications

As Figure 14 shows, the accuracies of rating-based classifications are very high. For business owners, they can use our classification model to check whether a raw review / comment text is positive or negative.

b) Keyword Extraction:

For each category, we compute one TF-IDF vector for positive reviews and another vector for negative reviews. We then manually analyze the sorting results of TF-IDF weights and extract valuable keywords, so that we can give advice to businesses according to these keywords.

Table 3: Keywords for Automotive and Night Life

CATEGORY	ATTITUDE	KEYWORD	WEIGHT
Automotive	**Positive**	car	97.107
		service	39.837
		work	35.221
		tire	28.026
		repair	26.124
	Negative	car	10.348
		wash	6.354
		oil	3.472
		time	2.880
		vehicle	2.051
Night Life	**Positive**	place	122.652
		wine	81.556
		love	69.285
		food	66.923
		bruschetta	58.859
	Negative	place	62.162
		bar	49.177
		food	44.173
		beer	34.627
		time	33.506

Based on the extracted keywords, we can suggest automotive businesses to pay more attention to tires, car repairing and washing, provide better oil and reduce service time. For businesses related to night life, we suggest their owners to locate their businesses in downtown areas, offer good food and drink, such as wine, beer and bruschetta, to their customers.

VII. Conclusion

Based on our naïve Bayesian classification and LDA clustering, we have completed a review-based recommendation according to industrial categories and subtopic matching. Given an arbitrary raw review without any extra information, we could label a category for this review, extract subtopic information and recommend related businesses for the user who presents the review. As for the business-oriented task, in addition, we successfully classify the reviews into the positive and negative groups with over 80 percent of accuracy on average. And by analyzing these two groups of reviews in each category, we extract the keywords that the customers care most about, which cast great insights to improving the management of business in various industries.

In the preliminary stage of our project, we held discussions for brainstorms and collected datasets from various resources. For preparation, we preprocessed the raw reviews with NLP tools and filtered them to remove useless information. In the system development stage, we applied the underlying algorithms of classification and clustering, and developed a website to achieve the corresponding recommendation user interface.

As we already applied review analysis into recommendation, and since the main approach is based on clustering, an unsupervised learning, it would be beneficial to develop a way to determine the accuracy of our recommendation. So it is with the business strategies that we offer. With the help of the user behavior analysis of our website as well as further research into the feedbacks from the industries, we could justify the correctness and accuracy of our model.

Besides, as an extension to the application of our review-based recommendation model, it would be interesting to apply the review analysis into the social networks. With a further step into the natural language processing, we can replace the raw text of reviews from Yelp with either a post, a review or any article from one's blog among arbitrary social networks, by which means, we can expand the application of our recommendation into a broader range. And that's one of our expected applications for this model in the future.

The contributions of each team members are listed below:

- ✧ Enrui Liao (Team leader): 20%, performing customer-oriented classification experiments, writing part of the proposal/final presentation slides and recording the video
- ✧ Yuqing Guan: 40%, performing LDA clustering experiments, building the online website and writing part of the proposal /final presentation slides and part of the final report
- ✧ Ying Tan: 20%, performing business-oriented classification experiments and doing the proposal and final presentations
- ✧ Mengting Wu: 20%, extracting keywords for improving business, writing part of the final presentation slides and part of the final report

Acknowledgment

Yelp Inc.

The Apache Software Foundation

The Stanford Natural Language Processing Group

Xuan-Hieu Phan and Cam-Tu Nguyen, authors of JGibbLDA

References

[1] Loren Terveen, Will Hill. Beyond recommender systems: Helping people help each other[J]. HCI in the New Millennium. 2001, 1:487–509.

[2] Badrul Sarwar, George Karypis, Joseph Konstan, John Riedl. Item-based collaborative filtering recommendation algorithms[C]. Proceedings of the 10th international conference on World Wide Web. ACM, 2001, 285–295.

[3] Ethem Alpaydin. Introduction to machine learning[M]. MIT press, 2004, 9-10.

[4] Andrew McCallum, Kamal Nigam, et al. A comparison of event models for naive bayes text classification[C]. AAAI-98 workshop on learning for text categorization. Citeseer, 1998, vol. 752, 41-48

[5] Corinna Cortes, Vladimir Vapnik. Support-vector networks[J]. Machine learning. 1995, 20(3):273-297.

[6] Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression[J]. The American Statistician. 1992, 46(3):175-185

[7] James MacQueen, et al. Some methods for classification and analysis of multivariate observations[C]. Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. California, USA, 1967, vol. 1, 281-297

[8] Hans-Peter Kriegel, Peer Kröger, Jörg Sander, Arthur Zimek. Density-based clustering[J]. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 2011, 1(3):231-240

[9] David M Blei, Andrew Y Ng, Michael I Jordan. Latent dirichlet allocation[J]. the Journal of machine Learning research. 2003, 3:993-1022

[10] Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, Xiaoming Li. Comparing twitter and traditional media using topic models. Advances in Information Retrieval, Springer, 2011. 338-349

[11] "Yelp Dataset Challenge." Yelp Dataset Challenge. Yelp, n.d. Web.

[12] Stanford CoreNLP. A Suite of Core NLP Tools[Z]

[13] Xuan-Hieu Phan, Cam-Tu Nguyen. Jgibblda: A java implementation of latent dirichlet allocation (lda) using gibbs sampling for parameter estimation and inference[Z], 2006