Interactive Map and Review Classifier based on Yelp

Hongyi Deng hdeng@ucdavis.edu

Chengkai Shi CKShi@ucdavis.edu

Abstract

Focusing on the Chinese restaurants in Sacramento on Yelp, we develop an interactive map with multiple new functions, inspired by the original Yelp filtering interface. The map helps to filter desired restaurants and locate them according to different attributes. This work offers a more flexible and user-centric tool for discovering desired restaurants. Moreover, we train a review classifier based on reviews and ratings of Chinese restaurants. Some dominating features are extracted and the accuracy of this classifier is analyzed.

Keywords:API, web scrapping, interactive visualization, classifier.

Contents

1 Introduction

1.1 Background

Yelp has been one of the most popular platform that focuses on food and other aspects of life. It offers all sorts of information from millions of restaurants, including addresses, ratings, reviews from customers or menus, etc. Fortunately, Yelp offers a public API called Yelp Fusion API which we can access and retrieve data from. For our project, we only pay attention to Chinese restaurant in Sacramento. According to the API, there are totally 495 Chinese rest with the label "Chinese". For our project, we are going to improve a function of Yelp and deal with the information provided by these Chinese restaurants.

1.2 Motivation and Significance

1.2.1 Interactive map

First of all, we notice that the filtering interface of Yelp can be improved. This filtering interface enables users to select certain restaurants according to their features, distances, ratings and other attributes. There is also a map to show the distributions of filtered restaurants, as is in Figure 1. However, this interface still has limitations. Here are some typical problems and how we are planning to fix them.

1. **Dilemma of ratings and reviews:** It is evident that the filtering tool of Yelp only allows users to filter restaurants based on either rating or number of reviews, but not on both simultaneously. This presents a dilemma: a restaurant with a perfect 5-star rating might only have a handful of reviews, whereas a restaurant with a 4.3-star rating could have over a thousand reviews, making the latter more trustworthy due to its larger sample size of feedback. Due to this dilemma, a user might be trapped in high-rated restaurants, and fail to make the best choice.

2. **Limitation on time option:** Furthermore, Yelp's filter options are somewhat limited in attributes like "open now". This can be inconvenient for users who are planning future dining experiences, as they have to manually check the operating hours of each restaurant for specific future dates. Our solution includes a feature for filtering restaurants by their operating days, allowing users to make more refined selections based on their intended dining times.

3. **Dilemma of ratings and reviews:** Considering that the type of cuisine is also an important factor for choosing a restaurant, we will incorporate a filter for cuisine types in the interactive map.

1.2.2 Classifier

Apart from an interactive map, we also take a look at the reviews and ratings of each Chinese restaurant, such as Fortune Chinese Food . These messages are given by the exclusive pages of those restaurants, as is shown in Figure 2. They are natural data points from which we are able to conduct sentimental analysis. By collecting all of them and organizing them into a dataset, we can establish a classifier that differentiate positive and negative reviews, and then analyze customers' sentimental orientations. Fortunately, every review on Yelp comes with a rating from 1-star to 5-star above it. This structure facilitates our analysis in that we can directly label the reviews according to ratings.

1.2.3 Significance

The two parts of our work are of real-life significance. Our interactive map can be considered as one step further of the existing interface on Yelp, with some new user-friendly functions. This work seeks to further tailor the search experience according to individual preferences, and enhance the overall utility of the platform for users who seek more specific dining experiences. On the other hand, the review classifier picks up certain features from reviews, and provides an insight into customers' emotions.

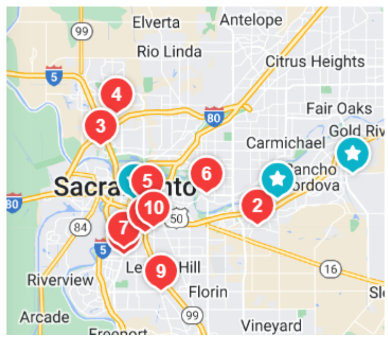

Figure 1: The map with filtered restaurants

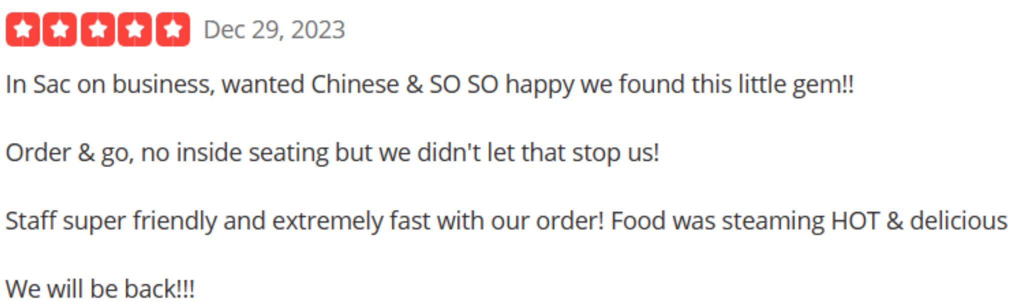

Figure 2: The structure of rating + review

2 Interactive Map

2.1 Basic Idea

To handle with the limitations mentioned before, we are going to equip our interactive map the following functions:

1. An interface that enables users to choose both ratings and number of reviews simultaneously.

2. A new filter to select restaurants by their operating days, containing the seven days of a week.

3. A new filter for different cuisine types, such as Cantonese food, Taiwanese food, etc.

2.2 Data Acquisition

We began by leveraging Yelp's API to collect basic data of Chinese restaurants and conducted an initial data quality check. Notably, we identified inaccuracies in the addresses of some restaurants. To rectify this, we employed a filtering mechanism based on the distance of each restaurant's location from the sample mean and the standard deviation. Additionally, we explored alternative sources for acquiring geographical information about the restaurants. Besides directly accessing Yelp's API, we also turned to website liek Nominatim and OpenCage for geolocation services, with the latter one demonstrating superior accuracy in our assessments.

2.3 Results

We succeed in realizing the functions as we planned. Here they are displayed.

The left picture in Figure 3 is a marker on the map for a restaurant with an incorrect geographical location. The right figure shows the markers for all the restaurants that meet the criteria, where the rating threshold is 4.0 and the review count threshold is 100, meaning that all the marked restaurants have a rating greater than 4.0 and more than 100 reviews.

Figure 4 in the next page are the markers for fast food Chinese restaurants displayed when the rating threshold is set at 1.0 and 2.5, respectively. The number of reviews is greater than 50.

Figure 5 in the next page shows the markers for Taiwanese and Shanghainese restaurants with a rating greater than 3.5, more than 100 reviews, and that are open on Wednesdays.

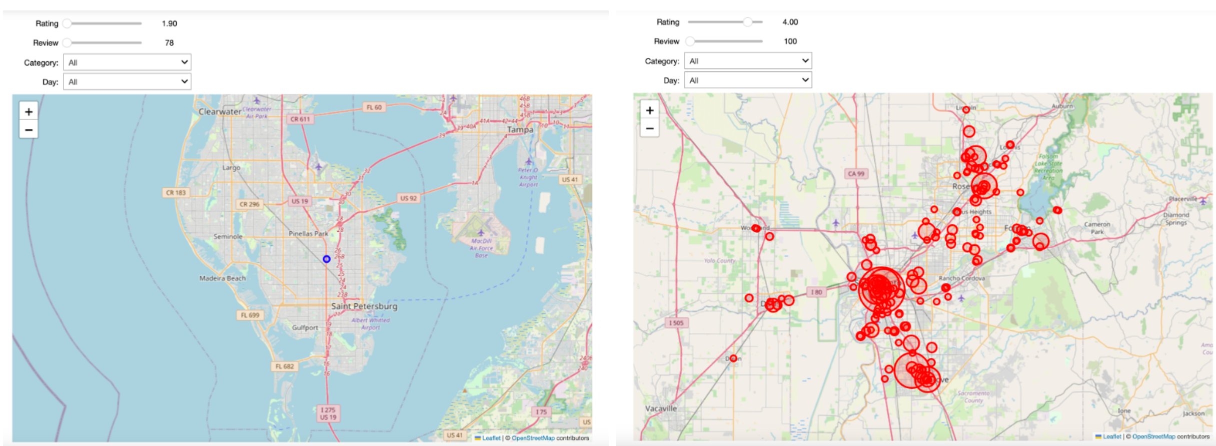

Figure 3: Change the rating and number of reviews simultaneously

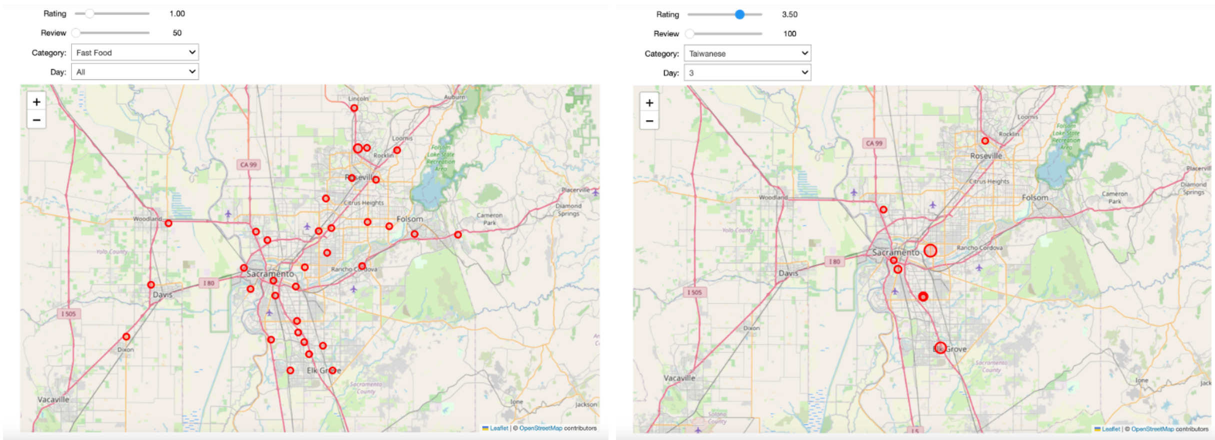

Figure 4: Change the rating and number of reviews simultaneously (2)

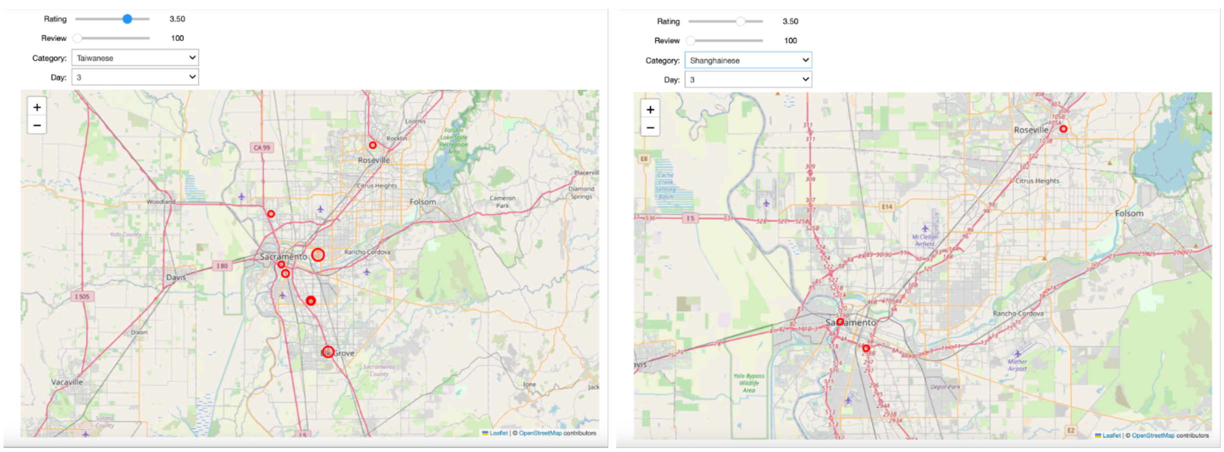

Figure 5: Select by cuisine types and operating days

3 Review Classifier

3.1 Basic Idea

We divide all reviews into two categories: “positive” and “negative”, and then establish a Naive Bayes classifier to distinguish between the two types. For each review, a label is tagged according to the corresponding rating. For a review with 3-star rating or higher, we tag it as “positive”. For 1 or 2-star rating, we tag the review as “negative”. After that, this classifier can be trained on our text set attached with features and labels.

For the classifier, a tokenizer is also needed for reorganizing the given text and extracting features from it. Here we apply a N-grams method to define our tokenizers, with “N” representing the number of adjacent words in this tokenizer. For instance, a “bag-of-word” tokenizer only generates features with one single word, while a "bigram" tokenizer provides features with two successive words. In this way, we can define tokenizers with any N that does not exceed the length of the longest review.

Notably, we have to delete those most frequently used words (stop words) in English, such as "we", "I", "is", etc. This helps make sure that the features are relevant to food or customers' sentimental orientation.

3.2 Data Acquisition

We retrieved all reviews, along with their ratings, through web scrapping from pages of Chinese restaurants on Yelp. Since the page is unstructured, we had to extract the review text and the number of rating stars from it. For the classifier, we need one set of reviews, and one set of the corresponding ratings. It is really easy to be blocked by Yelp, so we applies a time sleep of over 2 seconds, and finally succeeded in obtaining the sample sets we need.

Over 60000 reviews are collected in company with the ratings. To search for a pair including one review and its rating, we only need to input the same index in both sets because they are in one-to-one correspondence. We equally treat every review, as well as its rating, as a data point. In addition, we divide the set of text into a training set and a testing set for the classifier.

3.3 Results

3.3.1 Top features

Table 1 lists the top 10 "positive" features extracted by the first 4 tokenizers, with $N = 1, 2, 3, 4$. Meanwhile, Table 2 lists the top 10 "negative" features. The features are sorted by their log prior probability $P(feature|label)$.

From the top features, we can find customers' likes and dislikes. For positive ones, many of them are the names of some Chinese food. Moreover, some shows customers' willingness to come to this restaurant again, and some are words of compliments. For negative ones, food names also account for a large proportion of them. Besides, some features are customers' complaints about a particular dish or the environment. Generally speaking, the features extracted by N-gram tokenizers do reflect some characteristics of positive and negative reviews.

Table 1: Top 10 “positive” features

Tokenizer	bow	bigrams	trigrams	fourgrams
1	food	dim sum	honey walnut shrimp	fujian pan fried rice
2	good	chinese food	beef chow fun	baked bbq pork buns
3	chinese	fried rice	best chinese food	pan fried rice noodle
4	place	chow fun	come back try	steamed salted egg yolk
5	rice	honey walnut	dim sum place	steam golden shrimp rice
6	chicken	rice noodle	sweet and sour	chow mein, fried rice
7	service	orange chicken	hot sour soup	fried rice, chow mein
8	great	come back	salt and pepper	mein, fried rice, egg
9	restaurant	chinese restaurant	will come back	bbq pork duck clay
10	order	chow mein	go back try	yue huang chow fun

Table 2: Top 10 “negative” features

Tokenizer	bow	bigrams	trigrams	fourgrams
1	food	fried rice	dim sum place	asked owner was, said
2	chicken	dim sum	honey walnut shrimp	fried rice much soy
3	restaurant	chow mein	ordered orange chicken	roll good blow mind
4	poor	of the	rice much soy	ripped got general batter
5	order	orange chicken	owner was, said	roasted duck bun, scallion
6	like	first time	chicken fried rice	roast duck instead. cheated
7	one	tasted bad	restaurant new ownership	roast duck. siu mai
8	back	food came	many chinese restaurants	roast duck. bet cost
9	bad	chicken tasted	hot sour soup	decor. carpet filthy. hot
10	time	white rice	asked owner was	decor seemed like traditional

3.3.2 Accuracy

Accuracy is equal to the ratio of reviews that are correctly classified to all reviews. We collect 10 distinctive testing sets, in order to examine the accuracy and robustness of our classifier. Generally speaking, the accuracy of a certain tokenizer across all testing set tends to be stable without significant differences across all sets. Figure 6 illustrates how the average accuracy is associated with N.

For $N = 1$ or 2, the classifier performs well, with its accuracy over 0.8, which indicates that a tokenizer with no more than two words are suitable for the NaiveBayes method. However, it is obvious that the accuracy decreases sharply after $N > 2$ and buttoms out at $N = 5$. One possible reason is that, 3 to 5 successive words is neither short enough to make it easy to match, nor long enough to contain enough information for the classifier. As the N keeps growing, interesting things happen. For $N > 6$, the accuracy starts to recover and achieve another peak at $N = 150$. This is probably because a feature with so many words contains enough information on customers emotions. However, the accuracy does not continue increasing when N reaches 150 or more.

We can conclude that, a "bag-of-word"($N = 1$) or a N-gram ($N > 100$) tokenizer is the best choice for a classifier that dividing reviews into "positive" and "negative" categories. However, we are usually unable to collect a review with more than 100 words. N-gram ($N > 100$) tokenizers are not as pragmatic as they are expected to be.

Apart from changing testing sets, we also set up 10 different training sets and train the classifier on each of them. The accuracy of a certain tokenizer still changes little across each training set. This also demonstrates the robustness of the classifier.

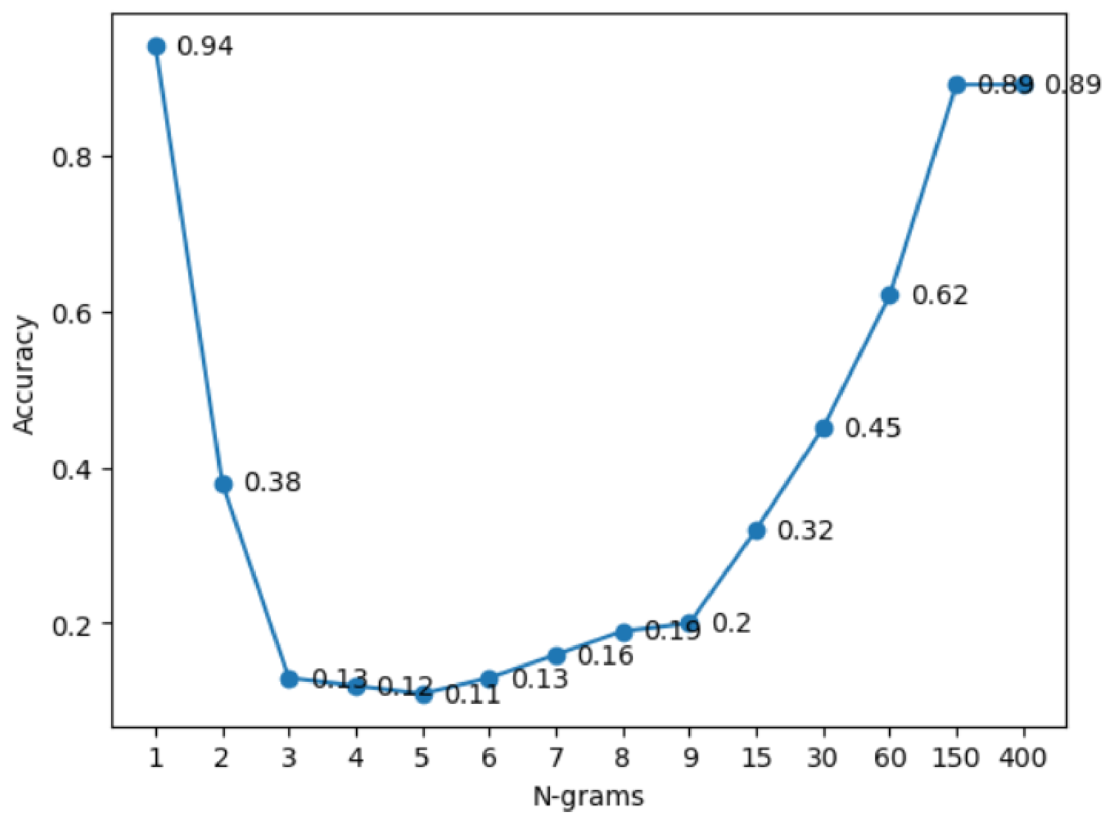

Figure 6: Accuracy of different tokenizers

4 Challenges and solutions

We went through a lot of challenges during our work. In particular, they took place frequently in our process of creating the interactive map.

1. In our study, the daily limit on Yelp's public API posed a significant challenge. Yelp restricts public accounts to only 500 API calls per day, which imposed a constraint on the number of samples we could process. Initially, our ambition was to examine Chinese restaurants in San Francisco. However, the abundance of establishments in that area meant that acquiring comprehensive data within the confines of the API's daily limit was unfeasible. Consequently, we adjusted our sample location to Sacramento. This strategic pivot allowed us to gather complete data on all Chinese restaurants within the city, adhering to the API call limitations. This modification not only ensured that our study remained within feasible operational parameters but also underscored the adaptability required in conducting research with third-party data sources, particularly when faced with unexpected technical constraints.

2. The design and presentation of the interactive map presented its own set of challenges. To enhance the user experience, we dynamically adjusted the map's zoom levels to ensure that the majority of restaurants displayed would always remain centered. This approach mitigates the need for excessive panning and zooming, providing a more intuitive navigation experience.

3. To visually represent the impact of both the number of reviews and ratings on a restaurant's perceived reliability and quality, we innovated by linking the size of a restaurant's marker on the map to both these factors. This design choice means that restaurants with a higher number of reviews and better ratings are more prominently featured on the map. Such a visual differentiation makes it easier for users to identify standout establishments at a glance, thereby facilitating a more informed dining decision-making process. This method not only adds a layer of depth to the map's informational value but also enriches the interactivity by allowing users to discern quality and popularity through visual cues.

5 Conclusion

5.1 Findings

1. We successfully develop an interactive map with some new functions beyond the original filtering interface. In our map, a user can filter his/her ideal Chinese restaurants by changing the threshold of overall rating and number of reviews simultaneously, or choosing from cuisine types and operating days.

2. The Naive Bayes classifier performs best under "bag-of-word" and N-gram ($N > 100$) tokenizer. That is to say, either $N = 1$ or $N > 100$ provides the highest accuracy of the tokenizer.

5.2 Limitations and Possible Improvements

1. The operating time can be further accurated to hours, such as "10 a.m to 5 p.m".

2. We treat all reviews of a restaurant equally. However, we can attach different weights to them, instead. For example, a review with more comments below it should be considered more important. Of course, it is hard to quantify and highlight such importance. For a Naive Bayes classifier, it might be helpful if we regard those important reviews as multiple data points in the training set.

3. There do exist some malicious low ratings that cannot truly reflect the sentimental orientation of the customer, or the real service level of the restaurant. If those malicious reviews can be identified and removed correctly, the classifier can be further improved.

References

[1] https://github.com/kramlinger/STA220_WQ24/blob/main/lectures/06-scraping.ipynb

[2] https://github.com/kramlinger/STA220_WQ24/blob/main/lectures/14-nlp.ipynb

[3] https://github.com/kramlinger/STA220_WQ24/blob/main/lectures/15-visualization.ipynb

[4] https://github.com/kramlinger/STA220_WQ24/blob/main/lectures/16-visualization.ipynb

[5] https://www.yelp.com/biz/fortune-chinese-food-sacramento-2?osq=Chinese

[6] https://www.yelp.com/search?find_desc=Chinese&find_loc=Sacramento%2C+CA&open_now=4136

Appendices

Python codes for our project are all in this Github link. Please have a check of it. https://github.com/BadAppleD/STAT-Project/tree/de6a727279d311af36b0cd032e9de573c808c862/STA%20220.