Application: Chinese Food Finder

Xiaoming Ji (xj9)

Introduction

Today, when people decide to dining-out, they need choose where and what to eat. Usually, this is done through searching on Google or Yelp. For people who is not familiar with a cuisine, for example, Chinese food, it's hard to conduct effective search to find the answer. This is because search engines lack of 2 important information to make such decision: what are popular dishes for a cuisine and what is the best restaurant to serve a dish.

Chinese Food Finder is a web application by leveraging the most advanced NLP technique and Yelp reviews data, to help user find the most popular Chinese dishes and decide where to take these dishes. It is a web based and user-friendly application that works well on both desktop and mobile browsers.

Technology

Application URL

https://chinesefoodfinder.azurewebsites.net

The application is hosted on Azure App Service.

Tested on Browsers

- Desktop: (latest) Chrome, Safari and Microsoft Edge.
- Mobile Phone: Safari mobile on iOS (it should also work on Chrome on Android).

Web Application

- The backend application is built on Python Flask web framework. Restaurant and Review data are pre-loaded to memory in order to improve query performance.
- UI is built on BootStrap 4.0 , BootStrap UI-Ecommerce and JQuery 2.0. These frameworks help to build beautiful and responsive web UIs that support both desktop and mobile browsers.

NLP

- The Chinese dish names are mined from Yelp reviews in task 3 using AutoPhrase. 190 high confident Chinese dish names are chosen for this application.
- Most popular Chinese dishes and restaurant recommendation are built from task 4 using customized algorithm. Based on our Yelp data, reviews for 998 Chinese restaurants are analyzed.

Functionalities and Implementation Details

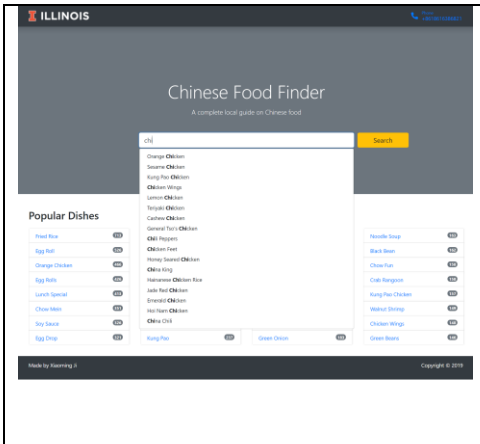

Figure 1

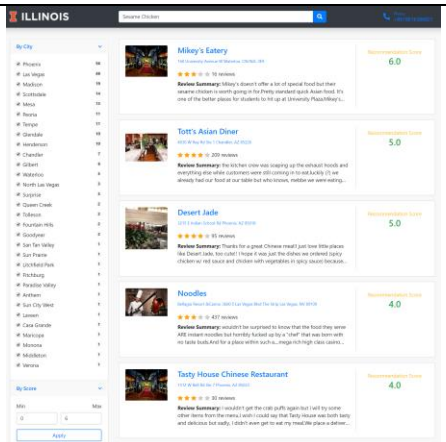

Figure 2

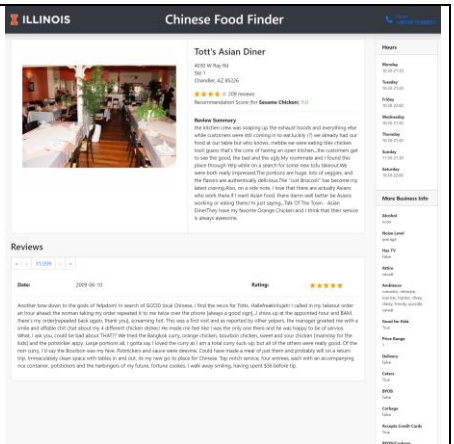

Figure 3

Home Page (Figure 1)

This is landing page of the application. User can search dish name in this page. Since Chinese dish name is hard to remember, I implemented auto-completion in the search-box to help user quickly find the dish

name by just typing any matching 2 characters. A JSON API for searching dish name is built to support this auto-completion.

The page also lists most popular Chinese dish names in this page as short-cuts. The numbers shown after dish names are restaurant (that have review mention such dish name) count that serve such dish.

Listing Page (Figure 2)

Once a dish name is entered and searched, the search results are rendered on this page. Search of non-existing dish name will redirect back to the home page.

This page lists top-20 restaurants that match the searched dish. The list order is based on the recommendation score calculated in Task4. It takes 2 factors into consideration: whether the restaurant is popular (more reviews) and whether the reviews on this dish is favorable by reviewers. And I compute the recommendation score using this formula:

Recommendation_Score = [Sum of all Sentiment_Score of dish reviews for this restaurant]

In the listing, an interesting feature is, instead of showing description for a restaurant (which I don't have such data), I replaced it with review summary. The summary was built on NLP technique called Text Summarization. A review summary was built by concatenating all reviews of a restaurant and find the most informative sentences. I used Gensim and PyTeaser to build this summary.

The review data comes for many cities and it is very common user only consider the city she lives. Thus, a city filter (on the left) is provided so that user can choose which cities that she is interested to explore. User can also apply recommendation score as another filtering condition to narrow down the searching scope. A JSON API for searching dish name with filtering is built to support the filtering smoothly.

Once the user wants to know more about a restaurant, she can click on the restaurant name to see the details page. Or if she wants to find out other dishes, she can do the search directly in this page.

Note: restaurant pictures used in this application are fake and randomly chosen for demonstration only.

Details Page (Figure 3)

This page list detailed information about a restaurant including name, address, recommendation score, review summary, business hour and other facility information. User can also check every review for this restaurant.

Mobile Support

Thanks to BootStrap framework, this web application works well on mobile browser and this is very convenient since most user have the needs to check restaurant in their mobile phones. The UIs are illustrated in the figures 4, 5 and 6.

Further Improvement

I can consider several approaches to improve the application to make it more useful.

1. The review data analysis and web application are separated processes. It takes hours and a lot of manual process to analyze new reviews data and publish to the web application. For practical application, we need to make the data analyze, model building and evaluation, and data publishing to serve online user in a wholly integrated pipeline. By doing this, we can make the most up-to-date dish names and restaurants available to our users.
2. Explore more on these reviews to find more insightful information. For example, we can do aspect opinion analysis on find information about price, service, hygiene and taste. This can better help our user to make decision.
3. Extend this application to support other cuisines. It is not technically hard to do so but need a more efficient way as mentioned in #1 to support more data mining work.
4. Build a standalone APP that are more user friendly.

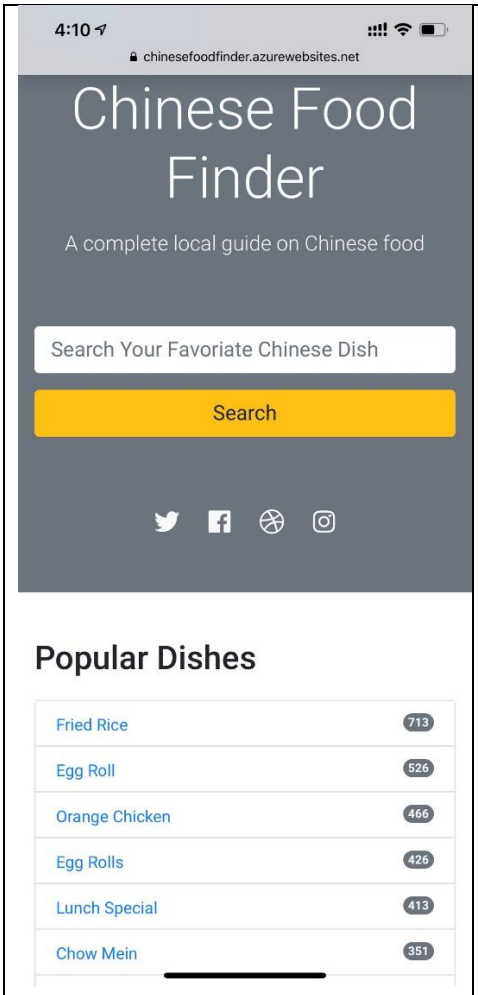

Figure 4

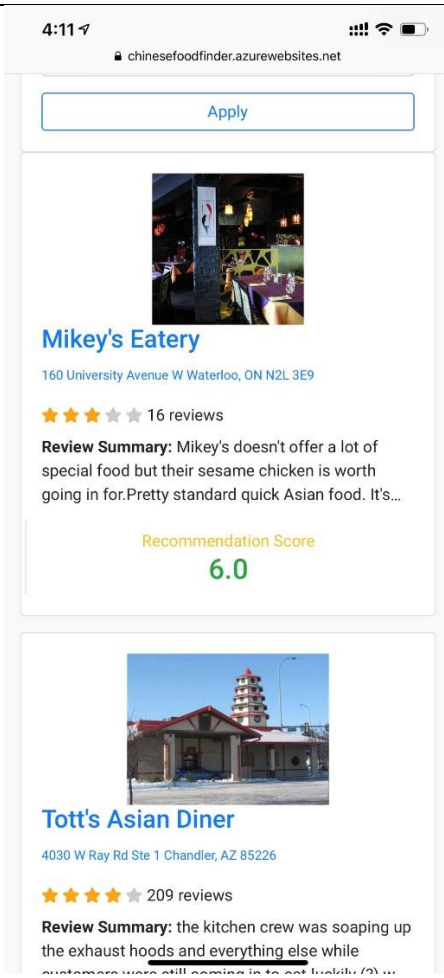

Figure 5

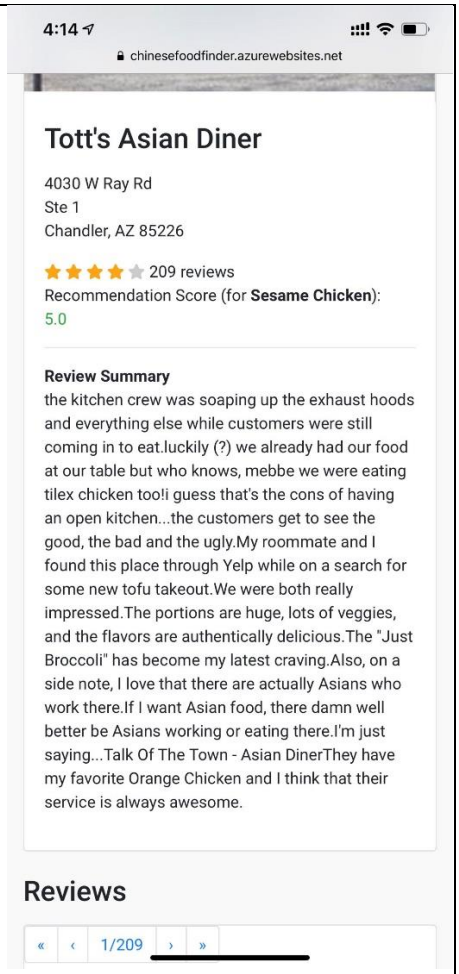

Figure 6