Task 4 and 5: Mining Popular Dishes and Restaurant Recommendation

Xiaoming Ji (xj9)

Overview

The general goal of Tasks 4 and 5 is to leverage recognized dish names to further help people making dining decisions. Specifically, Task 4 is to mine popular dishes in a cuisine that are liked by people; this can be very useful for people who would be interested in trying a cuisine that they might not be familiar with. Task 5 is to recommend restaurants to people who would like to have a particular dish or a certain type of dishes. This is directly useful to help people choose where to dine.

We use the **Chinese** dish names built from task 3 and search through Chinese cuisine reviews to determine the popularity of the dishes. Sentiment analysis is also used for these 2 tasks. This is especially helpful for task 5 as we not only count how many restaurants have reviews for a specific dish, but also whether this dish is favorable by the reviewers.

To visualize the result, we programmed an interactive D3 web page to illustrate the results.

Data Preparation

Dish Names

In task 3, we were able to use AutoPhrase to extract very good Chinese dish names from Chinese cuisine reviews. We then choose 196 high confident dish names for these 2 tasks. Here are few dish names we got from task 3:

> *chow mein, bean curd, chow fun, fried rice, peking duck, kung pao chicken, pad thai, beef chow fun, siu mai, hot pot, beef noodle soup, shaved ice*

Review Data

All Chinese cuisine reviews are extract from the original data set for analysis.

Sentiment Analysis

For sentiment analysis, we would like to have fine-grained opinion on the dish itself instead of the whole review. Thus, we will need segment the review to separate sentences. This has been done through *nltk.sent_tokenize.* To classify the sentiment of a sentence, we use the pre-trained VADER sentiment analysis tools in NLTK. *SentimentIntensityAnalyzer.polarity_scores()* gives 4 valences of a text. We further compute the sentiment scores using this formula:

*Sentiment_Score = (polarity_scores ['pos'] + 1e-10)/ (polarity_scores ['neg'] + 1e-10) * 0.5*

The Sentiment_Score is then round to [0, 1]. “1e-10” is to avoid divide-by-zero error.

Mining Popular Dishes

Popularity by Total Reviews

We search through how many reviews in total for a specific dish name. If one review contains many times of the dish name, we only count 1. The results are illustrated in Figure 1 (only contains 60 dishes due to space limitation). The color represents averaged sentiment score (on all reviews) of a dish. From warm color (Red) to cold color (Blue). Yellow means neutral. The sentiment score is also displayed after dish name.

Popularity by Reviewed Restaurant

A dish is more popular if it exists on more restaurants, thus it may be more reasonable to count how many restaurants have reviews for a specific dish name. The results are illustrated in Figure 2. The sentiment score is averaged among all restaurants.

Results Analysis

These 2 approaches give us very similar results especially for the most popular dishes. E.g., “fried rice”, “egg roll”, “egg rolls” and “orange chicken” are top 4 dishes on both figures. There are some small differences in ordering. For example, “egg foo yung” only exists in Figure 2.

The sentiment scores for a specific dish are also close and none of them have negative (<0.5) sentiment score. E.g., “fired rice” scores 0.68 in Figure 1 and 0.69 in Figure 2.

By manually exam the results, I would say both results are reasonable and useful to help user determine the popular Chinese dishes.

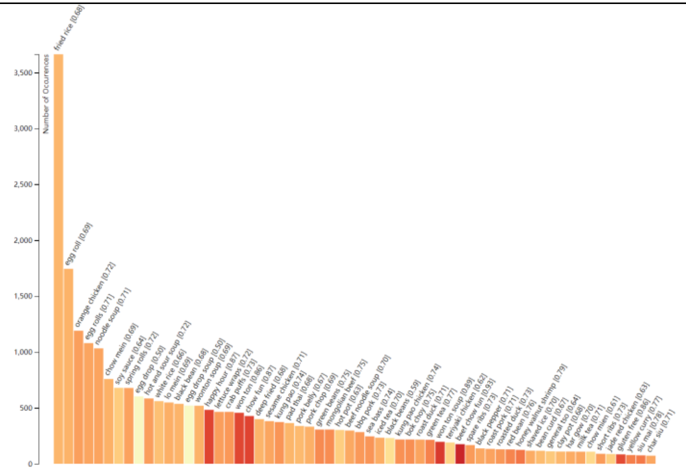

Figure 1

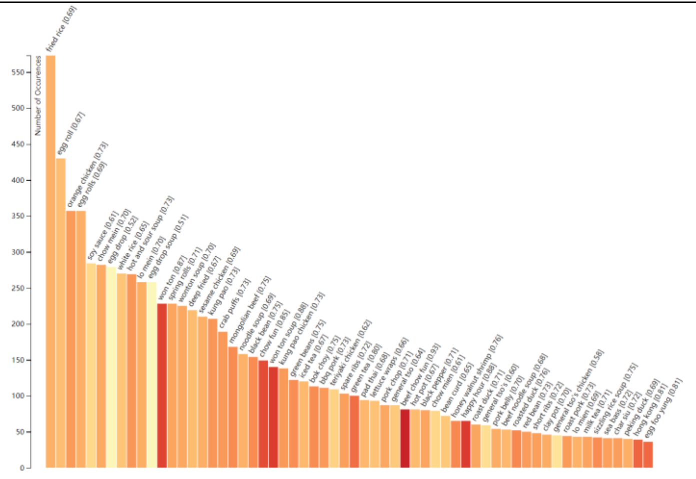

Figure 2

Restaurant Recommendation

To recommend good restaurants for a dish, we consider 2 factors: whether the restaurant is popular (more reviews) and whether the reviews on this dish is favorable by reviewers. Thus, we compute the recommendation score using this formula:

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

We build an iterative D3 web page so that user can easily navigate among different dishes to find the recommended restaurants (by comparing scores). Mouse over the bar will bring up the tooltip to show the computed polarity/sentiment_score and number of reviews for this restaurant. Sentiment score is also mapped to bar color.

Figure 3 and figure 4 illustrates results of 2 dishes. It’s hard to tell whether the results are good because I don’t know either of them. However, since the popularity is counted as major factor, high score is more likely to mean the restaurant is more visited. Thus, I would argue the recommendations won’t go too wrong.

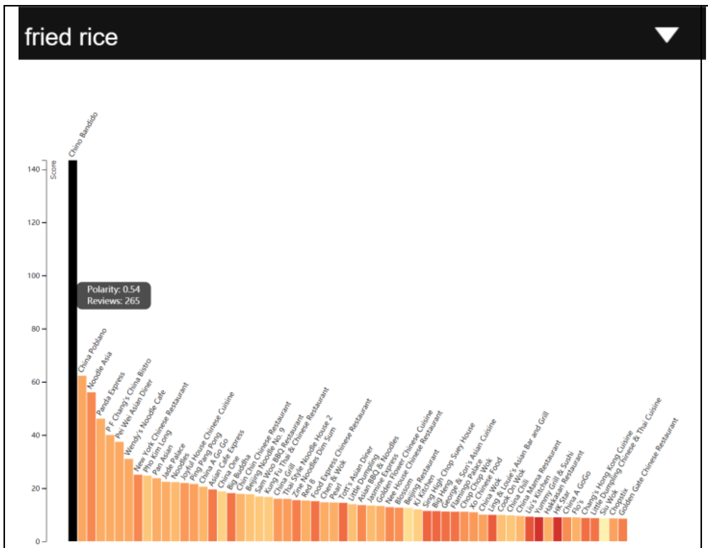

Figure 3

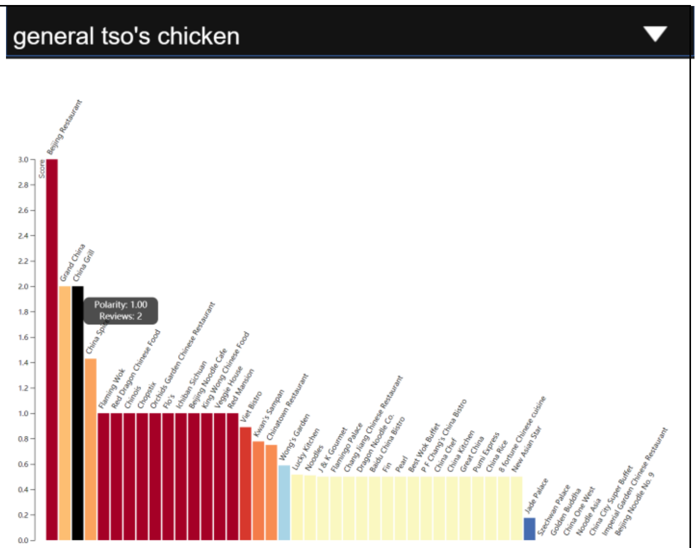

Figure 4

Further Discussion

We see by leveraging dish names built in task 3, simple string search and sentiment analysis. We can build reasonable popular dish mining and restaurant recommendation application.

We do have some weakness in this application that can be further improved.

- De-duplicate dish names. “egg roll” and “egg rolls” are one dish name, “roast duck” and “roasted duck” are one dish name either. Their reviews should be combined and thus increase their popularities. Stemming could be a simple solution.
- We use NLTK sentence segmentation function to split review to sentences. The results are better than regular expression match. However, we do see some sentences are semantically wrongly segmented. For example, “*I ordered general tso’s chicken and fried rice. They are above average but the fried rice is beyond my expectation*” are split to 2 sentences. Thus, this favorable sentiment is classified as neutral based on the 1st sentence. As another example, “geeral tso’s chicken taste good but the fried rice is horrible” has 2 aspects and will cause our approach fail to detect the true sentiment. Solve this problem will need more sophisticated NLP techniques. For example, grammar and semantic analysis.
- We use pre-build VADER tool for sentiment analysis which works OK. This tool is more general purpose and we can improve performance by building a sentiment classifier based on restaurant review dataset. This will need some manual labelling work though.
- To improve recommendation, we can leverage other information of a restaurant. For example: price, rating etc. Collaborative filtering and other recommendation techniques can also be used to give the user more personalized results.