Hungry Panda User Guide

How To Use The Chrome Extension And How It Works

Team: Hungry Panda

(Captain) Tingyu Zeng - tzeng11

Sicheng Meng - meng29

Yiwei Kuang - yiwei9

Jiachun Tang - jiachun5

Fangsheng Yang - fyang28

Overview

When Yelp users rate a restaurant, they tend to give the rating based not only on taste and quality of the food, but also other factors such as service, prices, comfort, atmosphere, parking, and location, etc.

As a group of Chinese students, our favorite dim sum restaurants usually don’t have a high rating on Yelp, partly because traditional dim sum places focus on taste of food, but many of them don’t focus on service, known as “cantonese hospitality”. You have to read each review if you want to know how good the food is.

We don't want to let bad service overshadow good food. Through text retrieval and mining techniques, we think it would be a great idea to analyze reviews that focus on food only.

With that interest, we created a Chrome-based extension to give an overview of what kind of food are frequently mentioned in the reviews and based on the review content what should be the sentimental score for this restaurant.

Team

Name	**Netid**	**Contribution**
Tingyu Zeng	tzeng11	Architecture
Sicheng Meng	meng29	Extension: data crawling
Yiwei Kuang	yiwei9	Backend: info retrieval and sentiment analysis
Jiachun Tang	jiachun5	Backend: info retrieval and sentiment analysis
Fangsheng Yang	fyang28	Extension: data presentation

User Guide

How to Install

Frontend

1. Clone the repository,
2. Go to the Extensions page by entering chrome://extensions in a new tab. (By design chrome:// URLs are not linkable.)
3. Enable Developer Mode by clicking the toggle switch next to Developer mode.
4. Click the Load unpacked button and select the extension directory.

Backend

There are 2 ways of installing the current dependencies:

- **[Recommended]**: if you have `conda` installed on your local machine, use `conda env create -f environment.yml` at the root directory. Activating the virtual environment can be done by `conda activate hungryPanda`.
- Using `pip`: `pip install -r requirements.txt` (note that you may need to configure the virtual environment yourself and if the installation does not work, you may need to configure your Python version based on runtime.txt.)

If you need to add a new dependency to the project, use the command: `pip install <lib_name>`, e.g, `pip install flask`. After the installation, update the txt file by `pip list --format=freeze > requirements.txt`.

How to Run

1. Run the backend via `flask --app backend/app run --reload`.
2. Go to a Yelp website in Chrome and click on the extension icon.

Implementation Details

Getting the Data

Instead of bulk scraping Yelp pages which requires a separate database to store data, we use JavaScript to extract a restaurant's page content in place, and save the a restaurant's reviews as list of objects.
This is also a more proper method for Chrome extension, because (1) it can extract whichever merchant's page on Yelp that the user is viewing; (2) the data retrieved is guaranteed to be updated. The drawback is that since Yelp change the structure of their website constantly, the JavaScript codes need to be altered once in a while in order to get the right data.
You can find the codes through this path: hungry-panda/extension/content.js
We first inspected a random restaurant page on Yelp to find its user reviews session. Under the reviews session, through this function `childNodes.length` we can tell that each page has ten (10) users' reviews (childNodes). We use `querySelectorAll` method of childNodes to achieve the data we are interested in. More specifically, the data we retrieve includes user's name, location (city, state), rating (number of stars), and reviews. We save each review and the user's information as an object. A restaurant page will return a list of ten (10) objects. The list is saved in `arr`.

Test Retrieval algorithm

Most frequently mentioned food(dishes) Analysis:
There are two main features that have been implemented. Once the text of json format coming from the front-end is passed. We sort out the body of the review, slicing the words of the sentence and matching them with an existing dictionary we made as dish recipes. In the function, `Food_showing_in_review,` we extract out the food shown in our food dictionary, which is a built in dictionary ,`food_dictionary,` including the most covered dishes and food.By doing so we can record what are those dishes frequently mentioned in reviews, and sum them. The number of result will be reflected on the front-end tool webpage

Sentiment Analysis:
Another major feature we've done is sentiment analysis. After removing reviews that did not contain food, we calculated the sentiment for each review.
Based on what we have learned in CS 410 class, we cut long sentences into words and calculated the score of comments according to the sentiment dictionary. Obviously, "This dish was terrible" would get a negative score, while "This food was pretty good" would get a positive score. Also, we took it a step further, and a comment like "Great! I've never tasted such a delicious goulash. Amazing!" would get a higher score than a simple "The goulash tastes good" - because the last one has stronger emotions. We think adding a sentiment analysis score would make Yelp ratings somewhat more accurate, because, to rate a restaurant on Yelp,

people can only choose between 0-5, not too bad food and not too demanding diners will always make this store easily get a 5-point rating.
After calculating the sentiment scores for each review, we averaged the them as the final score for the restaurant.