CS 410 Group ZZZRQ

Project Progress Report

Zetian Li, zli123
Xinhe Xu, xinhexu2
Rick Wang, haoran14
Qiwei He, qiweihe2
Zhichao Fan, zhichao8

Data Collection and Cleaning:

We downloaded the review data from yelp's official review database. We have filtered the reviewed data from 3 different timeframes:

1. 2018-08-01 to 2018-08-31. This period presents the general date review data.
2. 2021-08-01 to 2021-08-31. This period presents the covid-19 date review data.
3. The Thanksgiving period from 2018 to 2021.

We then constructed the relation between business names and business ID's by filtering the review text json and then picked the first 3 frequency reviews json which contains review_id, business_id and user_id, text, star and named them based on the business names. All python scripts within this process are uploaded to our github repo for review.

Model Training:

We studied and utilized the APIs of Azure Sentiment Analysis model for analysis review in batch and conducted two rounds of training data collection. For the first round, using random selection, we selected 200 reviews across the entire set of review data and isolated the review words and star review entries to treat as data points for model training. We then trained a custom text classification model for predicting the rating of a review based on the input text. However, it turned out that the model created using this training set resulted in a model that produced unreliable results, due to the training set being biased towards extreme star values. For the second round of training data collection, we extracted a dataset with 100 of each 1, 3, and 5 star reviews for future training due to the inaccuracy of the current model. We are in the process of tagging this training set through the API of Azure Sentiment Analysis.

Pending Tasks:

For the next steps, we will need to refine the model using the newly extracted training set. We will then need to conduct a round of data analysis according to the result from the model based on the dates and write some more python scripts to work with the Azure Sentiment Analysis model API and for extracting the results for data visualization. Finally, we will need to present the results in a dashboard fashion and evaluate our results to come up with a conclusion.

Challenges:

Because the original dataset is too large, which will cause a memory error, we cannot use these data directly. So we split the dataset into several files of 20000 review entries each and applied the query to each file. Also, the Azure Sentiment Analysis model appears not to be very scalable due to requiring manual importing and tagging. We are looking into automating the import process. As noted earlier, the first training set, even though employing random selection, resulted in a data set that was skewed towards the extreme scores: either 1 or 5 stars. We decided to have our extraction script to still employ random selection but keep on selecting until 100 of each 1, 3 and 5 stars reviews are retrieved.