# STAT 605 Project


| Xiangyu Wang | Zijin Wang | Jingshan Huang | Yicen Liu | Yuan Cao |
|---|---|---|---|---|
| xwang2439 | zwang2548 | jhuang456 | liu943 | cao234 |


## 1. Introduction

As we all know, Yelp is a world-class platform with more than 100 million store reviews worldwide. So we are curious about whether the reviews do affect business market or not. Specifically, our main goal is to explore which aspects are frequently mentioned in the reviews in order to give business owners some feasible tips for them to improve average star. What's more, we also wonder whether there will be any difference shows in reviews among different years which can reflect user habits and preferences. To investigate those questions, we use CHTC to accomplish large scale parallel analysis on "paired words" which will be described in detail later.

## 2. Methodology

### 2.1. Description of data

Our Yelp dataset is obtained from Kaggle (`https://www.kaggle.com/yelp-dataset/yelp-dataset`). Yelp is founded in 2004 and it develops, hosts, and markets website and mobile app, which publish user-sourced reviews and ratings about businesses. This dataset is a subset of the businesses, reviews, and user data from Yelp. It contains 8,021,122 reviews, 209,393 businesses, 1,968,703 users and 1,320,761 tips from users and is stored in 4 separate files, respectively. Also, business attributes like hours, parking, availability, and ambience are included in business data.

### 2.2. Data processing

Firstly, in order to read and filter the data with **awk** more conveniently, we transform the business data and the review data format from .json to .tsv. Secondly, we perform some column transformations in the review data. We reassign the level of stars in the review data as the following rules: reviews with 1-2 stars become level 1 now, 3 stars become level 2 and 4-5 stars become level 3. By doing this, we modify the rating rank from 1-5 stars to simply "bad", "median" and "good". For the date data, we extract only the years by regular expression. Then we merge the column "categories" in the business data into the review data by matching the business_id in both .tsv file. Thirdly, we filter "bad", "median" and "good" reviews given in 2015, 2016, 2017 and 2018, then we split the review.tsv into 3(star rating levels) $\times$ 4 (years) = 12 subfiles.

### 2.3. Computations and findings

For each subfile mentioned above, we run a R script to count the frequency of "paired words". The R script tokenize each review and delet all the stopwords, for example "the", "is", "at", "which", etc. Next, traverse all the review texts in the subfile and search for the consecutively two words, also known as "bigrams", occurred most frequently. The reason we compute frequecy of the bigrams instead of just the single words is straightforward. Comparing with sigle words like "good", the bigram "good

service" obviously conveys more information to the analysists. For each of the 22 categories, the R script output the top 100 bigrams and write the result as a .csv file. Here we use CHTC to run the R script in 12 parallel jobs. Therefore, after the computing, we obtain $12 \times 22 = 264$ .csv files, for instance, "review_2015_1_Food_top100.csv", "review_2016_2_Hotels & Travel_top100.csv", etc.


Figure 1: Demo of our word cloud.
According to review_2015_1_Active Life_top100.csv

## 3. Conclusion

### 3.1. Difficulties

By now, we do encounter some difficulties, the most exhausting one is that computing the frequency of the bigrams requires extremely large memory. The reason is that the computer should memorize all the counts of possible word pairs in the same time and then sort the frenquecy. We solve this problem by requiring more memories(up to 4GB) in CHTC. However, the script still terminate due to lack of memory when it runs based on 2018 data. We split each 2018 .tsv file into 4 new subfiles to run the R script and merge the result respectively.

### 3.2. Findings and future work

Going through different top 100 frequent bigrams, we find the bigrams do vary from years, ratings and categories.

In the coming weeks, we will conduct some statistical test to verify the significant difference among years, rating and categories. Also, we are going to merge all the top 100 .csv files into one large file "final_bigram_counts.csv". Based on the data in this file, we would build up a shiny app. The app would give the top 20 frequent bigram and a world cloud(see Figure1) corresponding to the year, stars and category the app user choose in the panel. Also, we will debug our codes and make sure it gives better performance.