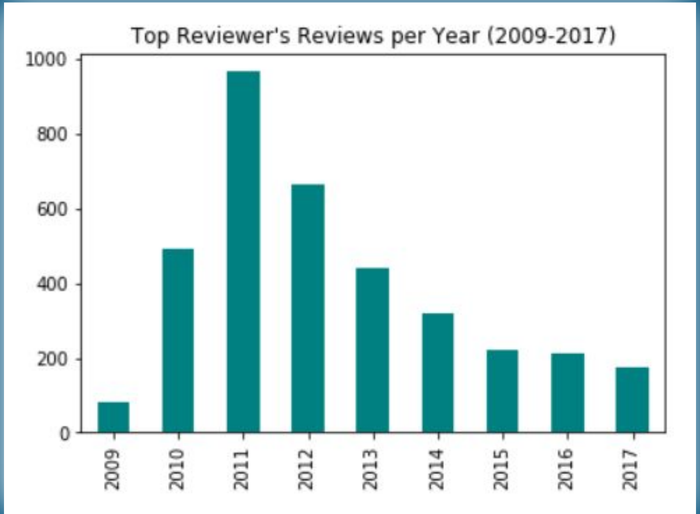
We Like Food
(very much)
An Analysis of Yelp Data

Yuqian Gong (Nancy), Leoson Hoay,
Lily Li, Alex Maiorella

The Data

- Yelp Academic Dataset
  - 174k businesses
  - 5.2 million reviews
  - Dated 2004 - 2015
  - 1.3 million users
  - 11 metropolitan areas



Our Questions

- Building a pair-based restaurant recommendation system
- Predicting business success via a variety of factors

Restaurant Recommendations: Doc2Vec

- Comparisons of reviews to determine <u>most similar other user</u> for each user
    - **Gensim library: Latent Semantic Analysis**
    - MRjob steps:
        - Build vector space, dictionary, and corpus from entire set of data
        - Calculate review cosine similarity for each pair of users
        - Pair each user with another user which is the “most similar”
            - This serves as the preamble for considering recommendable restaurants

Restaurant Recommendations: Procedure

Build dictionary of words from review text

Maps each word to an ID.

```
{'accepts': 0, 'add': 1,
'apples': 2, 'arrived':
3, 'aside': 4,
'banchans': 5, 'beef': 6,
'better': 7, 'bigger': 8,
'bite': 9, 'bottomless':
10, 'bowl': 11, 'centre':
12, 'cheap': 1, ...}
```

Build corpus (initial vector space)

Maps each ID to a count of occurences.

```
[[(0, 1), (1, 1), (2, 1),
(3, 1), (4, 1), (5, 2),
(6, 1), (7, 1), (8, 1),
(9, 1), (10, 1), (11, 1),
(12, 1), (13, 1), (14,
1), (15, 4), (16, 1),
(17, 1), (18, 1), (19,
1), (2z0, 1), (21, 1),
(22, 3), ... ]]
```

Calculate between-user similarity and find most similar other user

Build latent semantic vector space, compute pairwise user review similarity, outputs most similar user and the similarity score.

```
[0.7435081604506978,
"nNpkJm7bzaw0NN-UpNzMsw"]
,"CB0oWs49o7b8kvvXvx9eJw"

[0.7503735998997743,
"B2orbgTiEc2UfM-qQbTn5w"]
,"nNpkJm7bzaw0NN-UpNzMsw"

...
```

Output rec score of potentially recommendable restaurants

For each pair of users, calculate unique restaurants' rec score, based on restaurant similarity and weighted distance.

Still being implemented: Get top recs by some threshold

Or perhaps just the one with the highest "recommendability"?

Restaurant Recommendations: Scoring

- For every non-overlapping restaurant from the most similar user, calculate the ‘recommendability’ of each restaurant
  - MRjob steps:
    - Recommendation Scoring: **Average Restaurant Similarity x Inverse Average Haversine Distance** (pairing each restaurant from the most similar user with all restaurants from the original user)

$$Score_{rec} = \frac{\sum sim_{res}}{n_{ij}} \times \frac{1}{\log \frac{\sum Dist_{mn}}{n_{ij}}}$$

Where i = a particular user and j = the user most similar to i.

Restaurant Recommendations: Limitations

- Uni-dimensional definition of similarity
- Lack of resources to compare different similarity/scoring models
- Non-granularity: Had to aggregate reviews, perform subsetting
- Final recommendation score is still somewhat arbitrary

Business Success: Formulas

- Success scores:
  - (Average star rating) x (number of reviews) x (average sentiment score of reviews)
- Similarity scores:
  - Sum of pairwise similarity across other businesses within 50 miles
    - Pairwise similarity: (inverse exponential distance) x (category similarity) x (rating similarity)

Business Success: Procedure

Used NLTK in MapReduce to calculate average sentiment scores of reviews

Used SQL to build a database connecting a business to all its relevant attributes

Calculated success scores and similarity scores for each business using custom algorithm in MapReduce

Produced graphs that examine our hypothesis

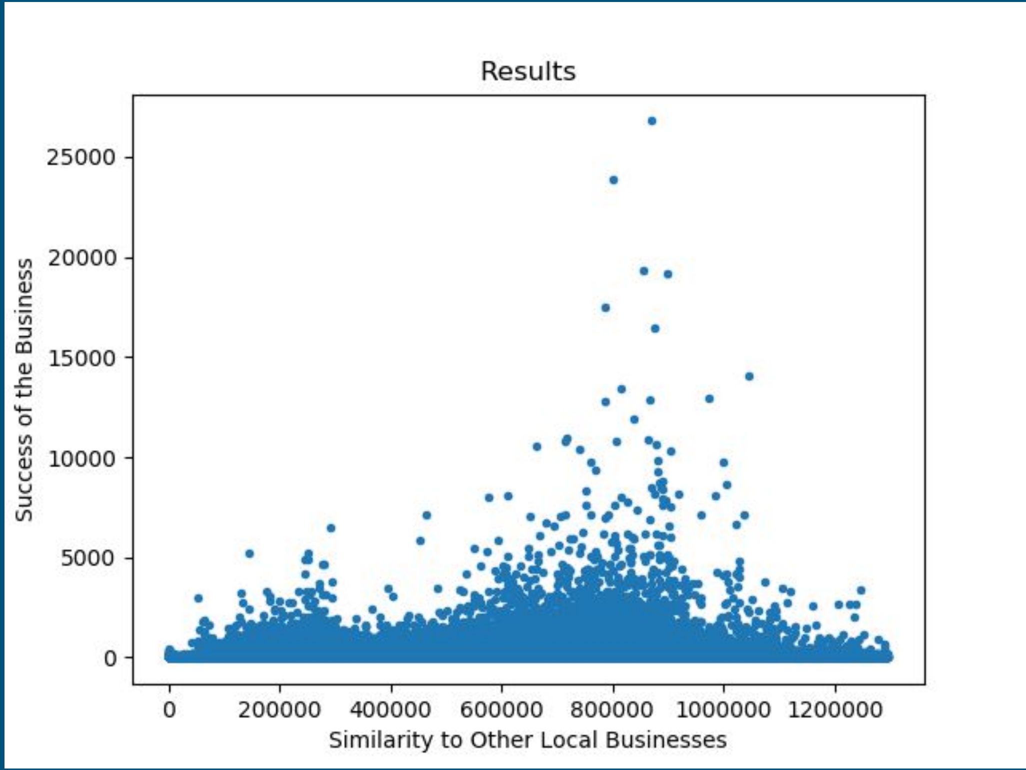
Maps each business to an average sentiment score of its reviews.


```
{'2zufEbSI_D8t3WPCgShw-w'
: 0.719625,
'3-oOqpfXKnLy-BnrS2FGDA':
0.723366667,
'30PzafMyWvh5fYZ8yV_qBg':
0.67554222}
```

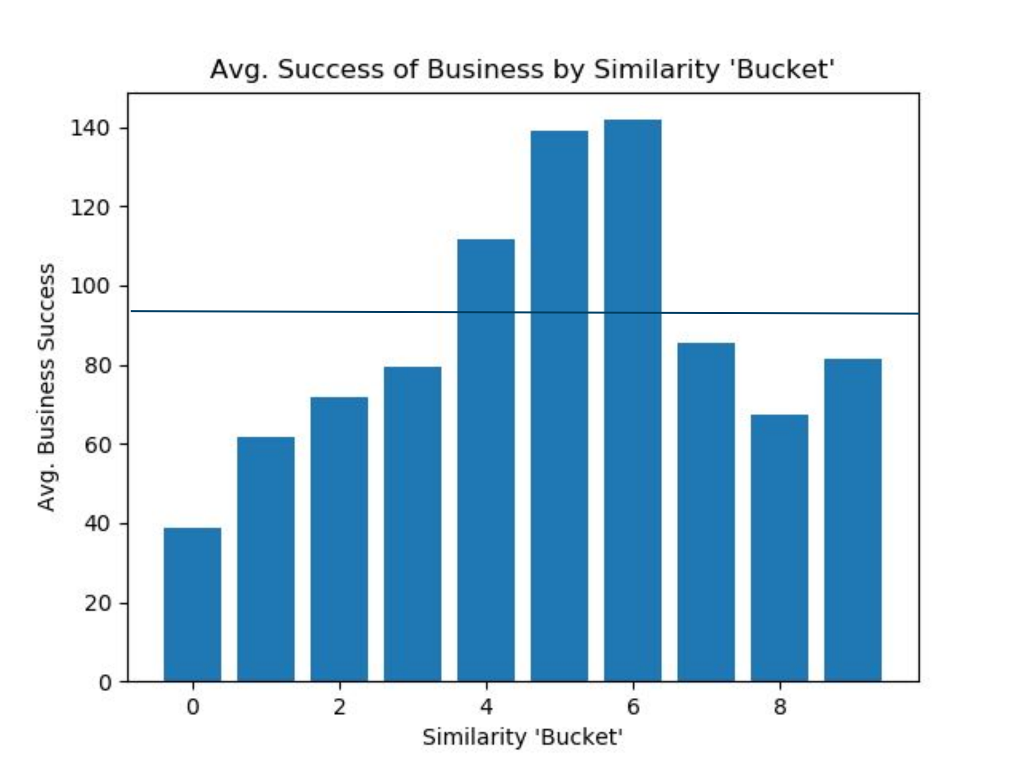

Maps each business to a (similarity score, success score) tuple.


```
{'7czyeIDxM2qrAm7N3UaLag'
: (311833.4379, 64.9072),
'8HycRNzRmhwEkZiMD8X9kw':
(341690.7183, 20.96525),
'9RkvonrX4kM26J0KxhXz6g':
(109372.1856, 6.912)}
```

Business Success: Results



Business Success: Results



Business Success: Conclusions

- Insights for entrepreneurs:
  - Geographical positioning and similarity appear predictive of business success
  - There exists an ideal balance between mutual benefit & competition
  - This ‘sweet spot’ emphasizes mutual benefit (Think: Kimbark Plaza or Harper Court)
- Limitations and further questions:
  - Which components of ‘business similarity’ were most predictive of success?
  - Unclear causality relationship (To what extent does success bring in new businesses, leading to higher similarity scores?)
  - Somewhat arbitrary metrics used to measure success

Challenges (and Solutions)

- mrjob could not handle newline characters in reviews text
  - Solution: cleaned reviews data and replaced line breaks with spaces before running it in MapReduce
- Importing packages and NLTK/gensim corpus in mrjob
  - Solution: bootstrapping
- Transforming mrjob output into reasonable format for future steps
  - CSV, SQL
- Trouble running MapReduce on whole dataset
  - Solution: MORE CORES!!! (125 to be exact)
  - Also split dataset up into chunks when possible

Questions?