Task 2: Cuisine Clustering and Map Construction

Xiaoming Ji (xj9)

Software Environment

- Python: 3.6.8
  - scikit-learn: 0.20.2
  - gensim: 3.4.0
  - nltk: 3.4.1
- Visualization
  - d3: v4
  - d3-hierarchy: v1

Data Pre-processing

Before mining the data, the following steps are performed on the dataset.

- Extract restaurant related reviews from the whole dataset and manually removed non-cuisine related categories (For example: Bowling, Nightlife etc). This gives me 147 cuisines in total.
- For each cuisine, concatenate all reviews to make a giant sentence representation of this cuisines.
- Tokenization: Split the sentences into words. Lowercase the words and remove punctuation. In addition, filter out tokens that appear in,
  - Less than 2 documents.
  - More than 50% documents.
  - Words that have fewer than 3 characters are removed.
- All stopwords are removed.
- Words are lemmatized. Words in third person are changed to first person and verbs in past and future tenses are changed into present.
- Words are stemmed. Words are reduced to their root form.

Visualization of the Cuisine Map and Improvement

To generate the cuisine similarity map, 4 models are used,

1. Concatenate all reviews for each cuisine and present these cuisines with their words' term frequency (TF). Then calculates pairwise cosine similarity between all cuisines' representation.
2. Concatenate all reviews for each cuisine and present these cuisines with their words' TFIDF. Then calculates pairwise cosine similarity between all cuisines' representation.
3. Concatenate all reviews for each cuisine and mining 100 topics using Mallet LDA algorithm. Then calculates pairwise cosine similarity between all cuisines' representation. The reason I choose Mallet LDA over genism native LDA implementation is because Mallet LDA perform better in finding similar cuisines and generating better clusters. Mallet LDA also has higher perplexity and coherence (using c_v measure) scores compared with genism LDA implementation.
4. As to vary the similarity function, build a doc2vec model with all individual reviews and represent a cuisine by inferring a vector from the concatenated reviews. Then calculates pairwise cosine similarity between all cuisines' representation.

Visualization

Cuisine similarity map is visualized using D3 as illustrated in the following figures. The map shows the similarity matrix, with every cell corresponding to the similarity between two cuisines. The opacity of each cell is the similarity - with a higher opacity for a higher similarity.

Such visualization method can't show too many cuisines. I selected 50 most popular cuisines (ranked by how many similar cuisines a cuisine has). For easy comparison, the cuisines selected for the first model are used for the rest 3 models.

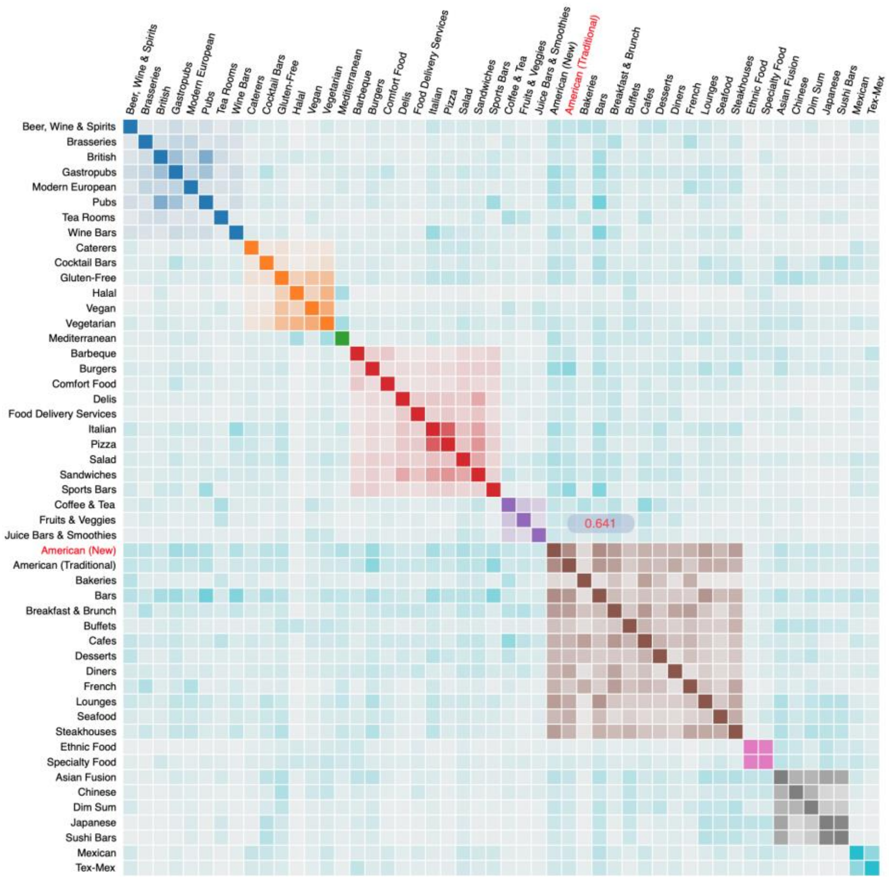

Figure 1: TF (Concatenated Reviews)

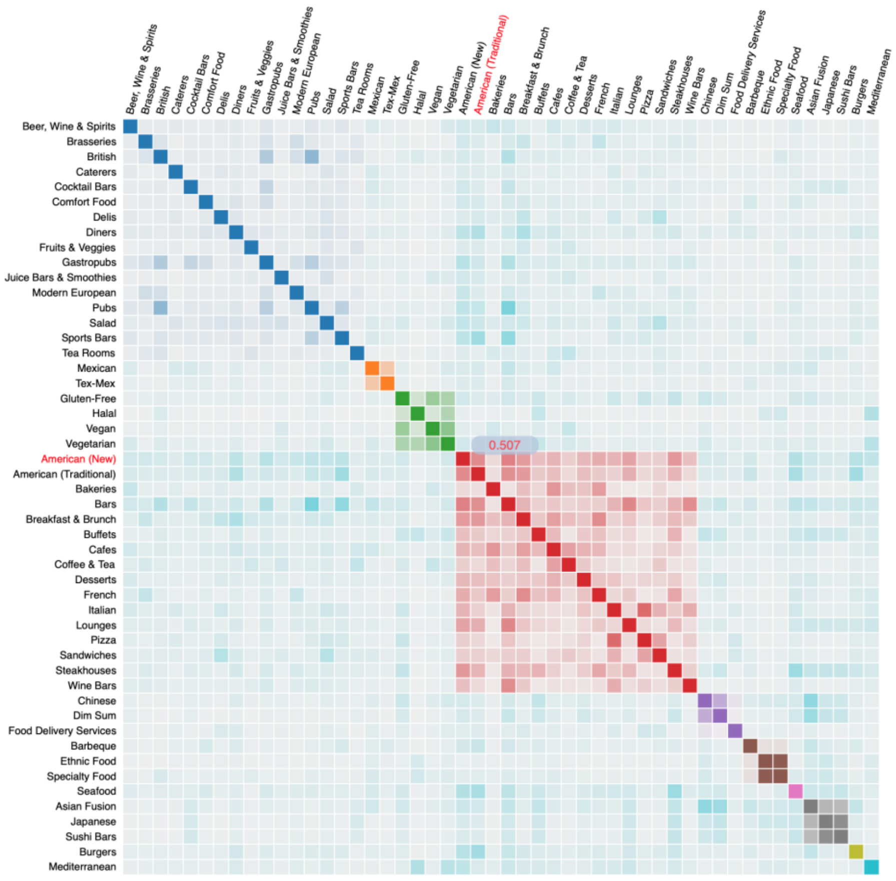

Figure 2: TFIDF (Concatenated Reviews)

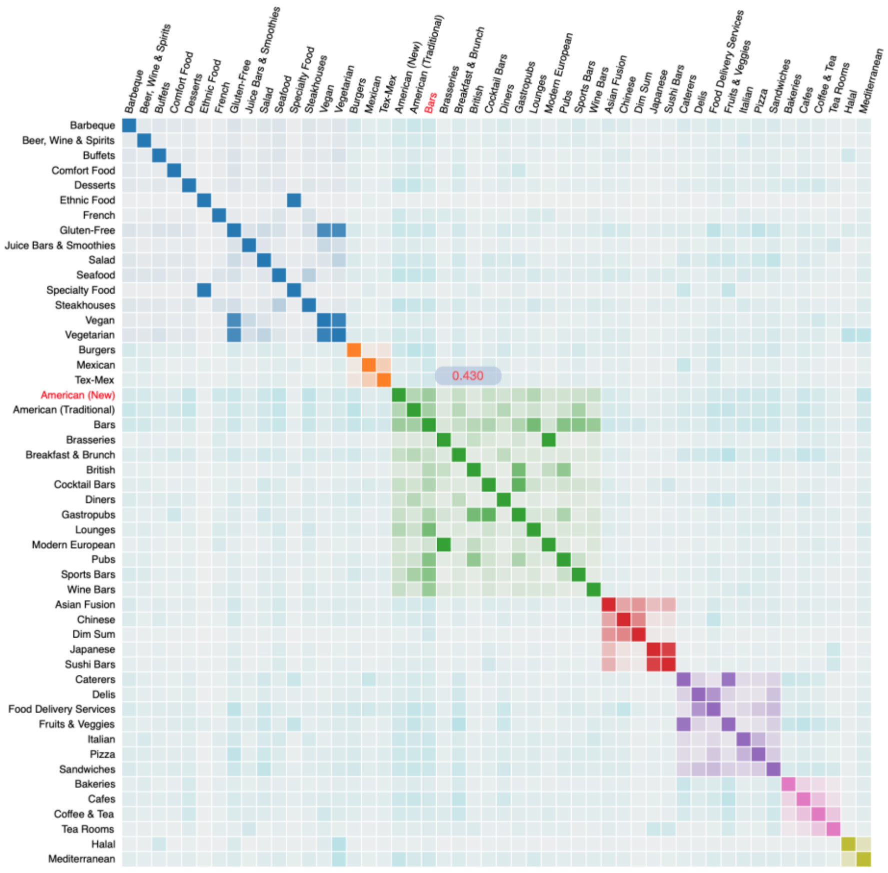

Figure 3: Mallet LDA (Concatenated Reviews)

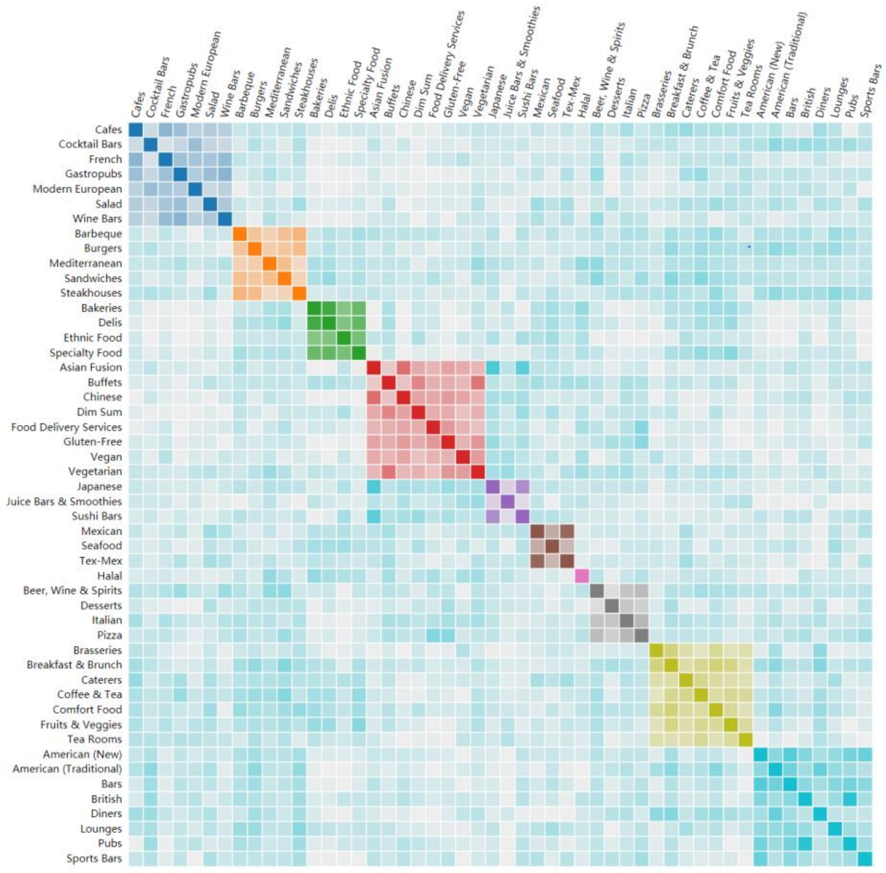

Figure 4: Doc2Vec (Individual Reviews)

This matrix map give user a quick way to find which 2 cuisine are related. To improve visualization and usability,

- Hover over a cell will trigger a tooltip that shows the similarity score. The cuisine names are also highlighted in red so that user can easily find out which 2 cuisines she is comparing.
- For each map, the cuisines are grouped in 10 clusters. These clusters were generated (on all cuisines) using birch algorithm and are marked in different colors.

Further Exploring

All these maps find me very reasonable results. For example, they all show "*Japanese*" and "*Sushi Bar*" are similar and grouped together, "*Mexican*" and "*Tex-Mex*" are similar, "*Italian*" and "*Pizza*" are similar. Since the matrix map can't show all cuisines. I made another cloud map so that user can select one cuisine name from the list box on the top and see more clearly on how other cuisines are related to this one. User can also simply click on the word to see the results. A URL parameter can specify which model should be loaded for investigation.

I can use this tool to compare the results of different models more easily. The following figures show results for "*Chinese*" cuisine.

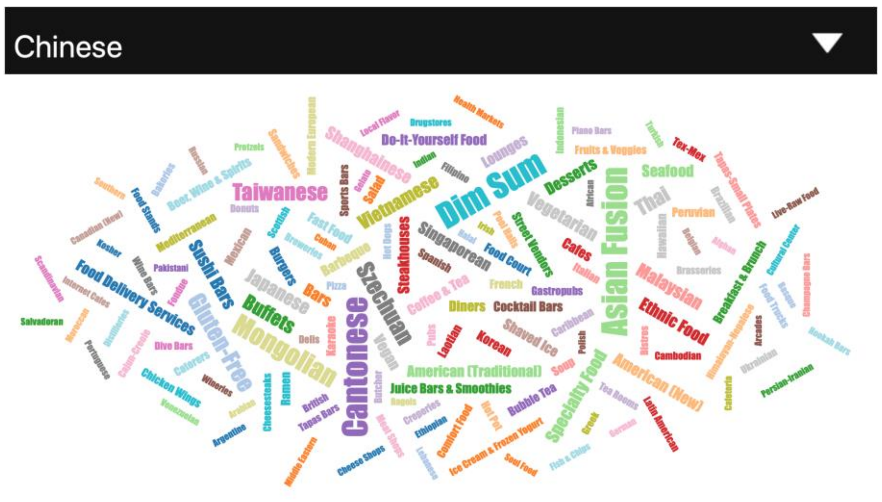

Figure 5: TF (Concatenated Reviews)

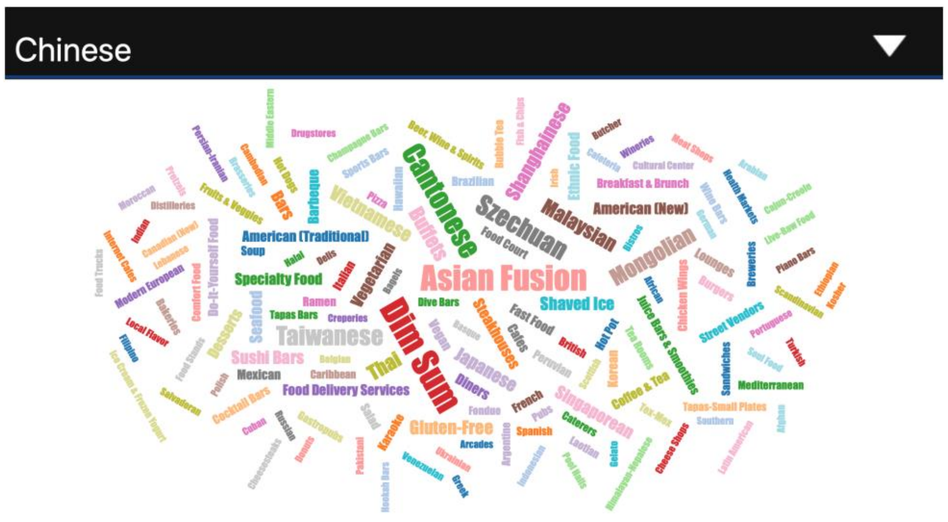

Figure 6: TFIDF (Concatenated Reviews)

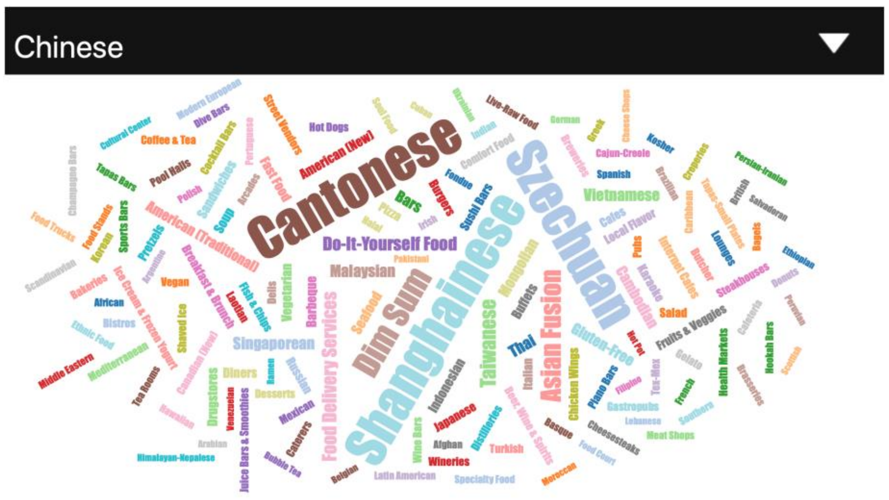

Figure 7: LDA (Concatenated Reviews) - **BEST**

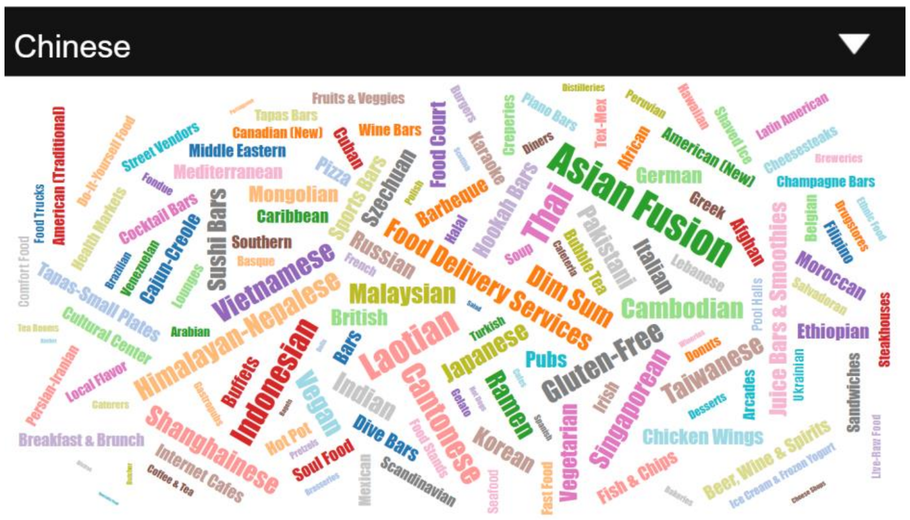

Figure 8: Doc2Vec (Individual Reviews)

In Figure 7, we can clearly see "*Cantonese"*, "*Shanghainese*" and "*Szechuan"* are highlighted to be similar to "*Chinese"*. This is indeed true, as they are sub-categories of "Chinese" cuisine. Figure 6 highlight "*Dim Sun*" and "*Asian Food*" more than "*Cantonese*" and "*Szechuan*". It is also correct. But it seems to me the options from LDA sounds more relevant if I were to build a cuisine recommendation application. As another example, "*Cocktail Bars*" LDA gives me "*Gastropubs*" (which I didn't know what it is before googling it) and "*Local Flavor*" which are very interesting options. While TF and TFIDF don't highlight much. Further exploring other cuisines, I would say **LDA** perform best among these models. TFIDF is better than TF. And Doc2Vec is not very impressive compared with others.

Cluster Analysis

Clustering helps grouping related cuisines for better demonstration and recommendation. K-Mean, Birch and DBScan (by Scikit-Learn) are tried to find the clusters. K-Mean and Birch are further explored to compare the clustering results. Since these algorithms use Euclidean distance to compute similarity but the cuisine similarity maps are generated using cosine similarity, normalization is applied to the cuisine representation data in order to make them equivalent. For each algorithm, 2 cluster groups are generated:

1. Group 1: 10 clusters.
2. Group 2: Clusters that score the highest Silhouette Coefficient.

The results are visualized as following figures. The text color is rendered based on cluster of group 1 (10 clusters).

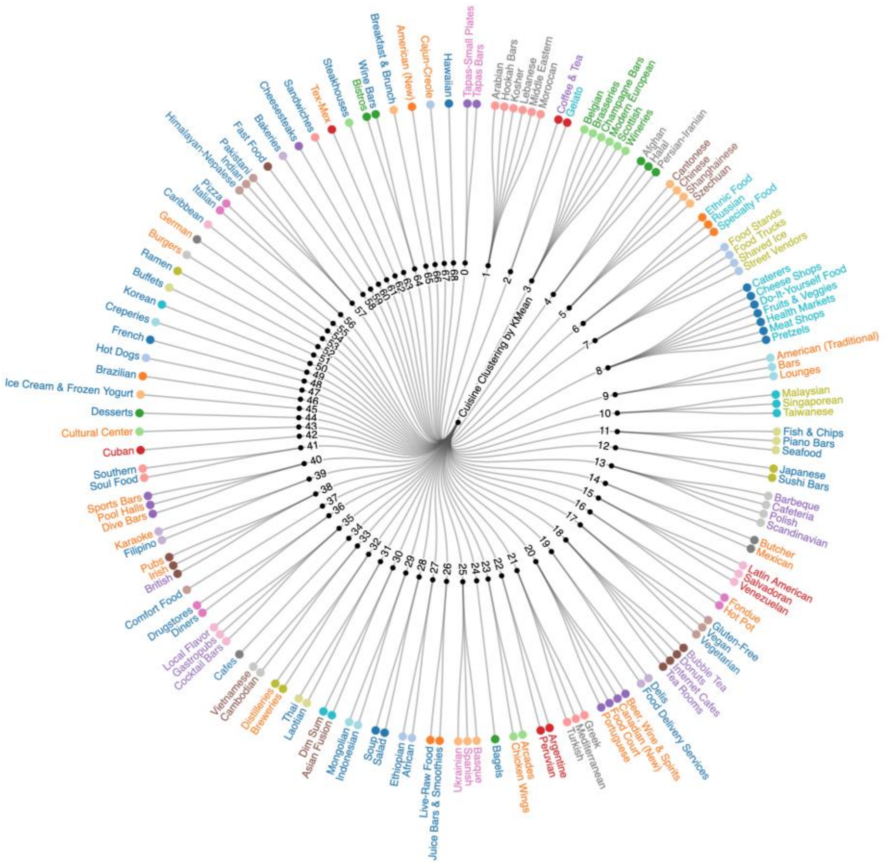

Figure 9: K-Mean on LDA

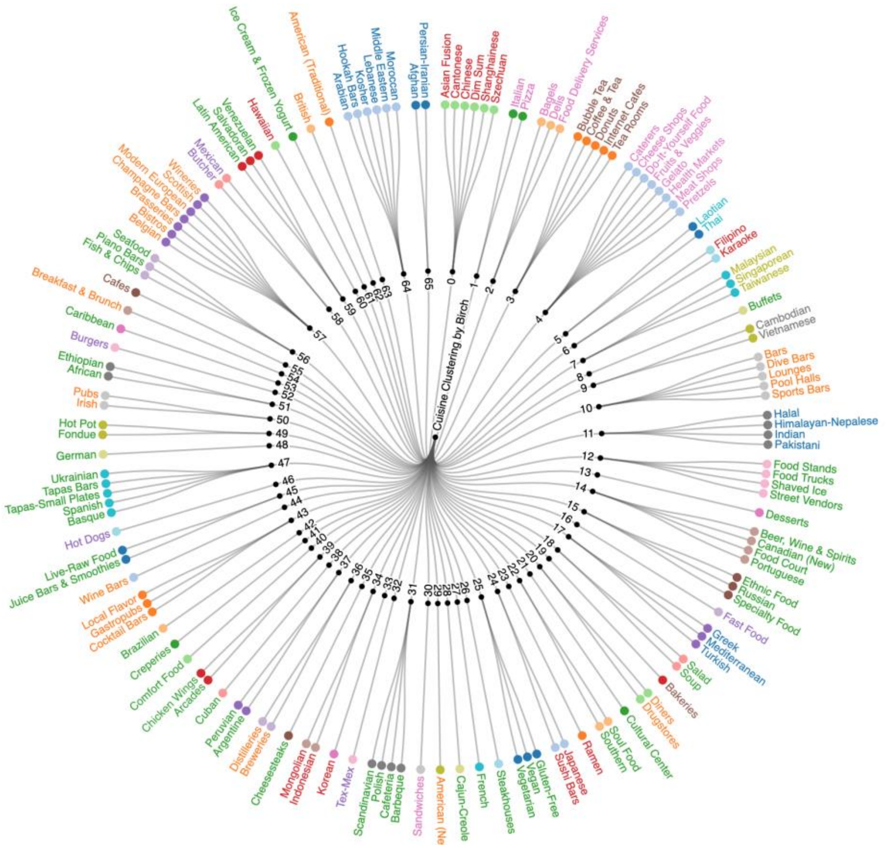

Figure 10: Birch on LDA

Both algorithms give very good and similar results. For example, "*Food Stands*", "*Food Trucks*", "*Shaved Ice*" and "*Street Vendors*" are in one cluster of both cluster groups. To compare,

- Birch gives more meaningful results in the group 1. For example, K-Mean makes "*Food Stands*", "*Food Trucks*", "*Shaved Ice*", "*Street Vendors*", "*Taiwanese*", "*Singaporean*" and "*Malaysian*" in one cluster while Birch split them in 2 clusters.
- Birch use smaller number of clusters (66) to achieve best Silhouette score vs. K-Mean (69) and the has fewer groups that only contain one cuisine.

Therefore, I use Birch to generate a 10-cluster for LDA model and incorporating the cluster info in the cuisine similarity map as illustrated in Figure 1, 2, 3 and 4.

Conclusion

The key to this task is find a way to represent cuisine reviews, thus it's a text representation problem. LDA perform the best among all 4 models in terms finding the meaningful results. It also only uses a 100-size vector to represent a cuisine (TF and TFIDF use the size of vocabulary, and Doc2Vect also uses 100-size vector).

Further improvement could be applied by leveraging more advanced text representation techniques, for example, by leveraging Google's BERT or Microsoft MT-DNN models. But I haven't tried since these models are quite large and require good hardware (GPU with at least 16G of memory) to work.