Task 2: Cuisine Clustering and Map Construction

In this task data from Yelp reviews are mined to discover knowledge about cuisines. Specifically, a cuisine map is created that allows us to identify similarities between cuisines. Multiple approaches are used to improve the cuisine map including using different term weightings as well different representations of cuisine similarity. Finally, clustering is applied and the various cuisine maps and their approaches are compared.

Preparing the Data

To create the data set for this task the summarized reviews provided by UIUC for this Capstone project was downloaded from: https://d396qusza40orc.cloudfront.net/dataminingcapstone/Task2Cuisines/cuisines.tar.gz. The files were reduced by choosing those that were obviously ethnic in nature (i.e. Italian, Russian, Tai, etc.) then further minimized by eliminating all files under 100 Kb in size (cuisines with the fewest amount of reviews. This produced a set of 69 cuisines (as shown in the plots)

Having selected the cuisines, each cuisine file was imported into R for processing. For each cuisine all reviews were aggregated into a single document. The resulting corpus contained 69 documents – one document for each cuisine. Pre-processing on the corpus included converting text content to ASCII, converting text to lower case, removing stop words, removing numbers, removing extra white space, removing words less than three characters and applying stemming.

A document-term matrix was created from the corpus using the DocumentTermMatrix() function. The mean term frequency-inverse document frequency (tf-idf) was then used to refine the vocabulary. This measure removes terms that have low frequency as well as those occurring in many documents. We only include terms that have a tf-idf value of at least 5.0e-06 which is a bit more than the median and ensures that very frequent terms are omitted. Finally, a cosine similarity matrix was computed using term frequency weighting. The similarity values of this and all subsequent matricies were scaled column-wise to a value between 0 and 1 using the scale() function. For visualization the corrplot() function was used. Smaller and lighter-colored circles represent lower similarity between cuisines. Larger and darker circles represent higher similarity.

Visualizing the Cuisine Map

On first review of the cuisine map (Figure 1), the cuisines with very high similarity stand out visually - Asian-Fusion/Japanese, Hot Pot/Fondue, Chinese/Szechuan, Indian/Pakistani and Tapas Bars/Spanish. These combinations seem obvious. Next, looking a familiar cuisine - Greek - and its most similar cuisines we see the following: Afghan, Halal, Lebanese, Mediterranean, Middle Eastern and Turkish. These provide some interesting insights as Greek is a fairly well known cuisine. It is obvious that it would be related to Mediterranean and Middle Eastern food. However one might not be familiar with cuisines such as Afghan or Turkish. So it seems that this cuisine map effectively identifies similarities between cuisines as well as some useful insights.

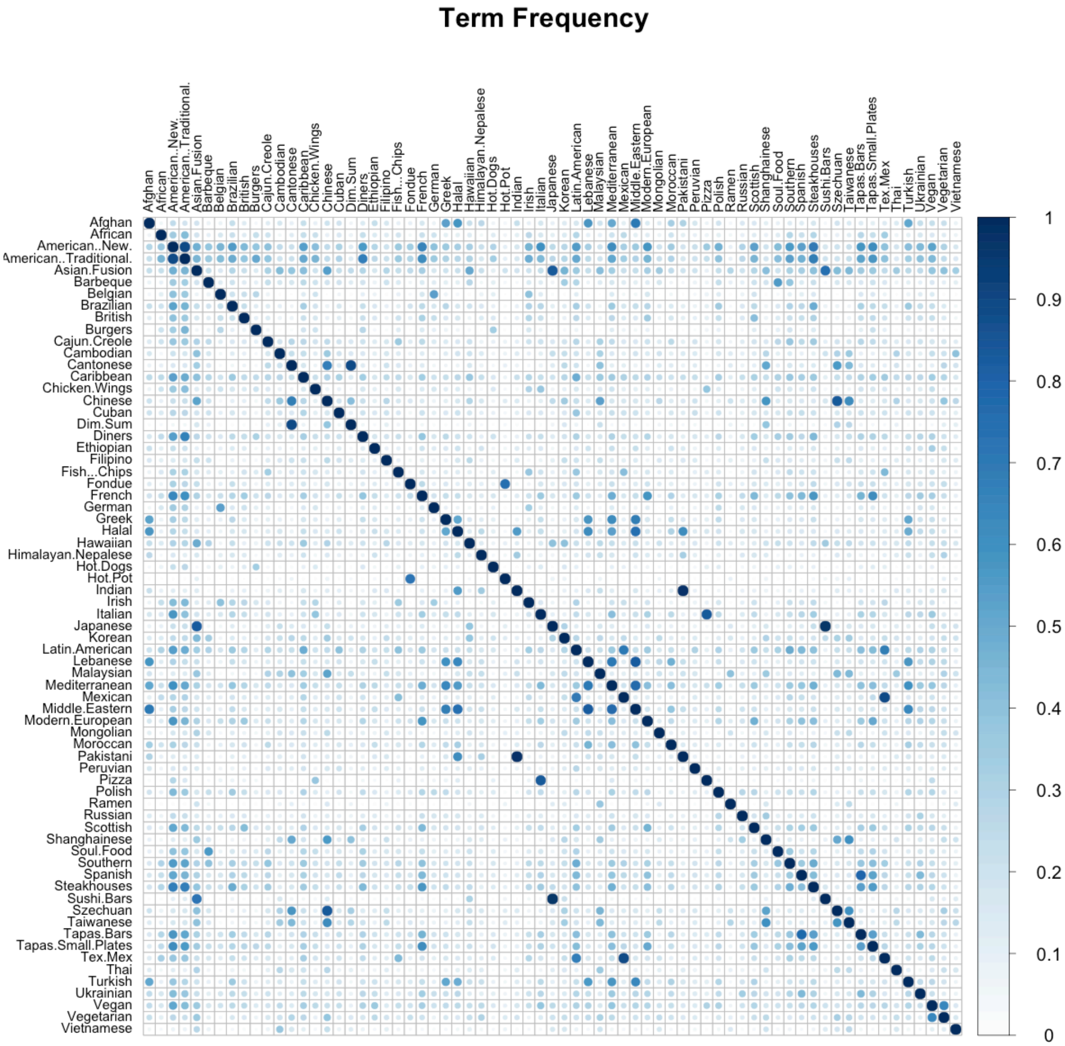

Figure 1 Cuisine Similarity - Term Frequency

Improving the Cuisine Map

Two approaches were investigated to improve the cuisine map. First, term frequency-inverse document frequency weighting was applied to the original document-term matrix and the cosine similarity matrix was computed. The second approach applied Latent Dirichlet Allocation (LDA) on the original document-term matrix and the posterior topic distribution for each cuisine was used for the similarity metric. To compute the similarity matrix based on LDA the dist() function was run with the topic distribution data using the Euclidean Distance method. The resulting dissimilarity matrix was converted to a similarity matrix using the as.simil() function. The results s are shown in Figures 2 and 3.

The TF-IDF cuisine map maintains the strong similarities noted in the TF map. For instance we still see a strong relation between Asian-Fusion/Japanese, Hot Pot/Fondue, etc. We also continue to see a strong relationship between Greek and Afghan, Halal, etc. What TF-IDF has done, however is removed a significant number of less significant similarities between cuisines. This can be attributed to the effectiveness of TF-IDF to remove very common words and thus filtering “noise” from cuisine map. It’s not clear, however, whether or not TF-IDF can eliminate too many terms and thereby removing some cuisine similarities that may be valid. It does appear though that TF-IDF does provide an improvement over TF.

The LDA results are somewhat similar to the TF cuisine map. Many of the similarities from TF are also in the LDA map. There also appears to be many insignificant similarities (i.e. noise) as well. There are some slight differences between these cuisines maps. For instance in the TF map Pizza had a strong similarity with Italian. That similarity does no appear in the LDA map. On the other hand Shanghainese has a stronger similarity to Chinese and Szechuan in the LDA map than it does in the TF map. So, LDA provides similar results to, but not necessarily any significant improvement over Term Frequency.

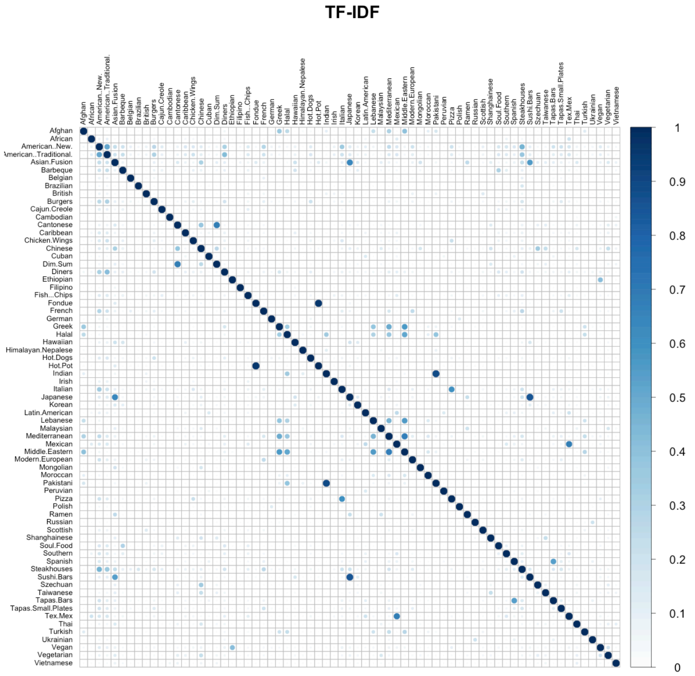

Figure 2 Cuisine Similarity - TF-IDF

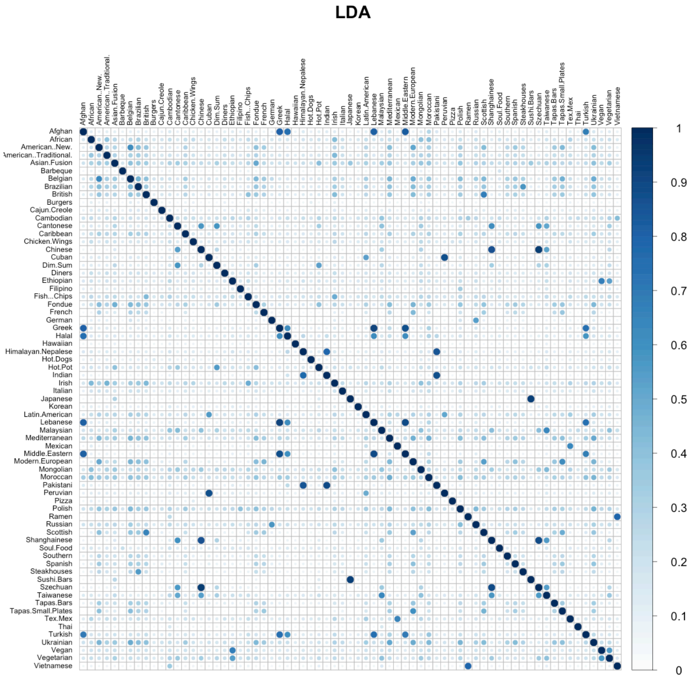

Figure 3 Cuisine Similarity - LDA

Incorporating Clustering in Cuisine Map

Two clustering algorithms were applied to each of the TF, TF-IDF and LDA cuisine data – K-Means clustering using kmeans() function with the "MacQueen" method and Heirarchical clustering using the hclust() function with the "average" method. Both algorithms were run with 10, 17 and 25 clusters.

For both algorithms 10 clusters seemed to produce grouping that were too broad (too many dissimilar cuisines in a cluster) and 25 clusters produced grouping that were too narrow (many groupings of a single cuisine). 17 clusters (roughly ¼ the number of cuisines) seemed to produce grouping that were logical and cohesive. For simplicity the following evaluation will focus on the differences between the two algorithms as applied to the TF-IDF and LDA maps, each using 17 clusters (Figures 4 -7). Rectangles highlight the clusters on each map. Of the four maps shown it is notable that for Hierarchical Clustering / TF-IDF the clusters make very little sense and don't seem to include any similar cuisines. The other maps all are effective at clustering similar cuisines. Each has its pros and cons, but K-Means/TF-IDF produces the most balanced and cohesive cuisine map. Table 1 summarizes pros and cons of each map.

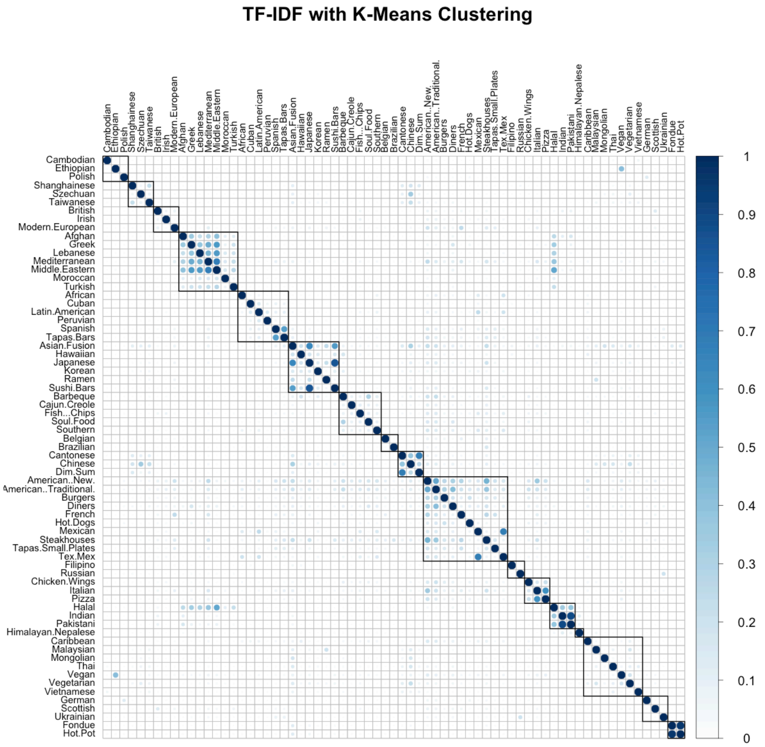
TF-IDF with K-Means Clustering

Figure 4 TF-ID with K-Means Clustering

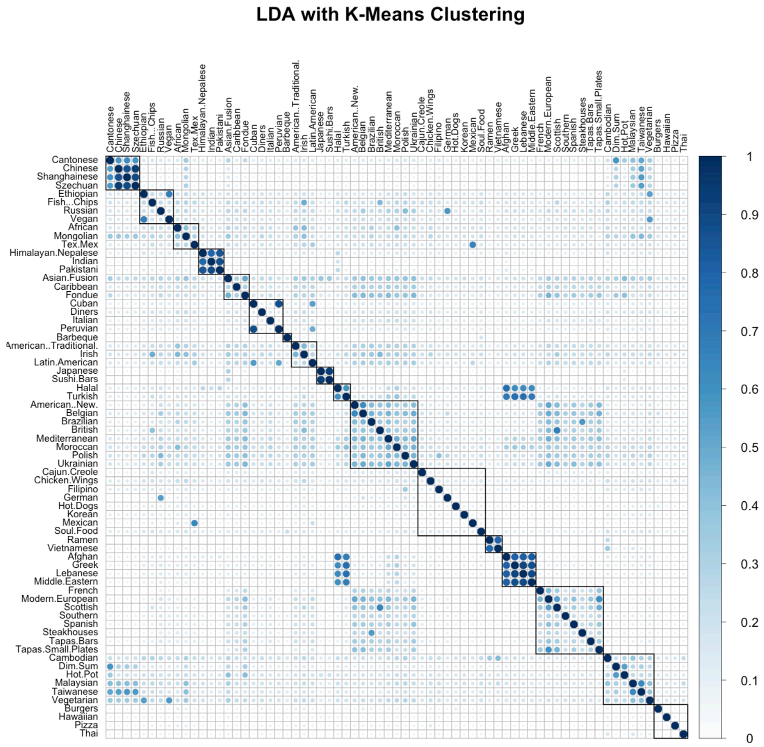
LDA with K-Means Clustering

Figure 5 LDA with K-Means Clustering

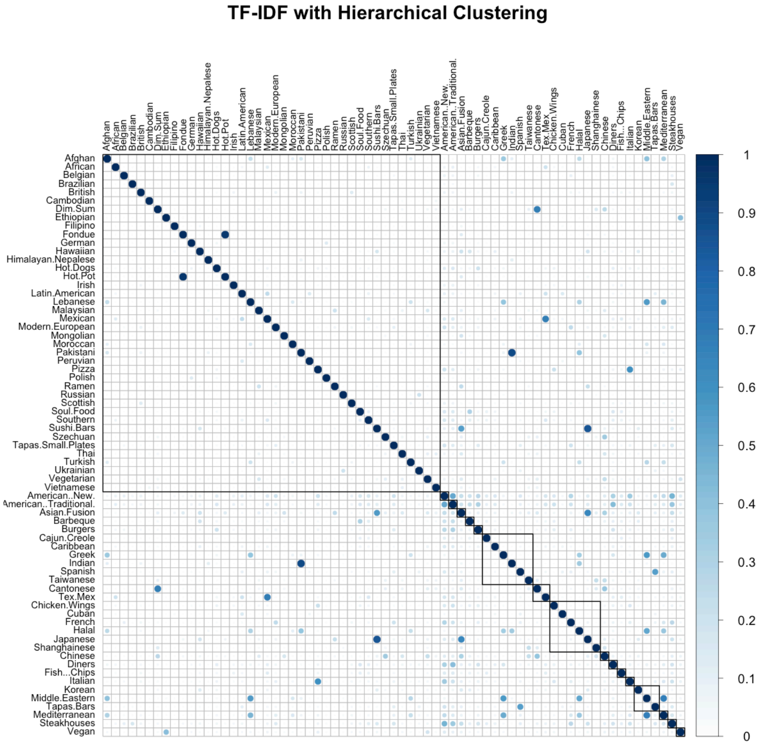
TF-IDF with Hierarchical Clustering

Figure 6 TF-IDF with Hierarchical Clustering

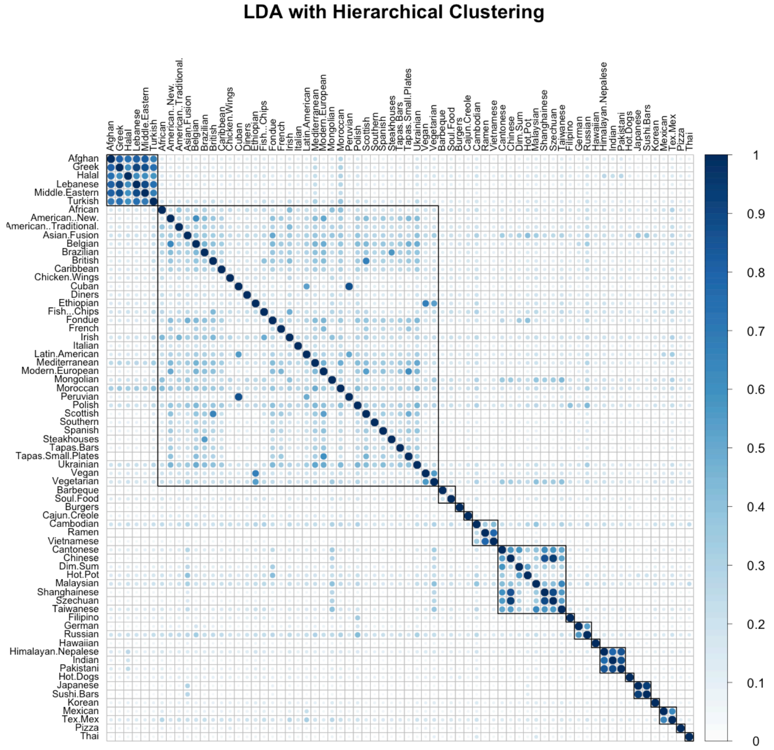
LDA with Hierarchical Clustering

Figure 7 LDA with Hierarchical Clustering

Algorithm	Pros	Cons
K-Means / TF-IDF	Balanced clusters (size ranges from 2 to 10 cuisines). Logical, cohesive clusters for the most part.	One of the larger clusters has a mix of cuisines that appear not to be strongly similar.
K-Means/ LDA	Balanced clusters (size ranges from 2 to 8 cuisines). Logical clusters for the most part.	Clusters don't seem as cohesive as K-Means / TF-IDF. There are a number of similar cuisines that fall outside of a cluster.
Hierarchical / TF-IDF	None apparent	Unbalanced clusters (size ranges from 1 to 40). Clusters do not make sense
Hierarchical / LDA	Unlike K-Means/LDA clusters are very inclusive of all similar cuisines (i.e. you don't see similarities that are outside of a cluster.	Unbalanced clusters (size ranges from 1 to 33).

Table 1 Summary of Clustering Algorithms