Task 2: Cuisine Clustering and Map Construction

Brief description

I've completed the task using python language and did it in iPython notebook (http://ipython.org/notebook.html). Data loading and preparation was done through Apache Spark SQL (http://spark.apache.org/sql/). The Natural Processing Toolkit (http://www.nltk.org/) was used for text processing, tokenizing and stemming. For TF-IDF extraction, building of cosine matrices, clustering and clustering performance evaluation was used scikit-learn toolkit (http://scikit-learn.org). All visualization was completed with help of matplotlib (http://matplotlib.org/).

Particularly, I've done following things:

1. Extracted all categories that occur with category 'Restaurants' and analyzed them and selected 98% most frequent categories.
2. The dataset was presented as pair of a category and and concatenation of all reviews related to a business from the category.
3. For all categories was built TF-IDF model and cosine matrices and provided visualization of similarities matrices.
4. Following experiments were performed:
    a. Model consists only from TF components
    b. TF-IDF model
    c. TF-IDF model with filtering of terms with TF less than 2 and higher than 0.8 proportion of documents.
    d. TF-IDF based on texts with removed punctuation and applying of Porter stemming algorithm, Lancaster stemming algorithm and WordNet lemmatizer.
5. For experiments I provided visualization in form of a map for similarity matrix and dendograms for result of hierarchy clustering.
6. Mini Batch K-Means algorithm was applied for the obtained similarity matrices for different number of clusters. Since there is no ground trough for the result of clustering were calculated Silhouette score and the relation of score and number of cluster was depicted on the plot.

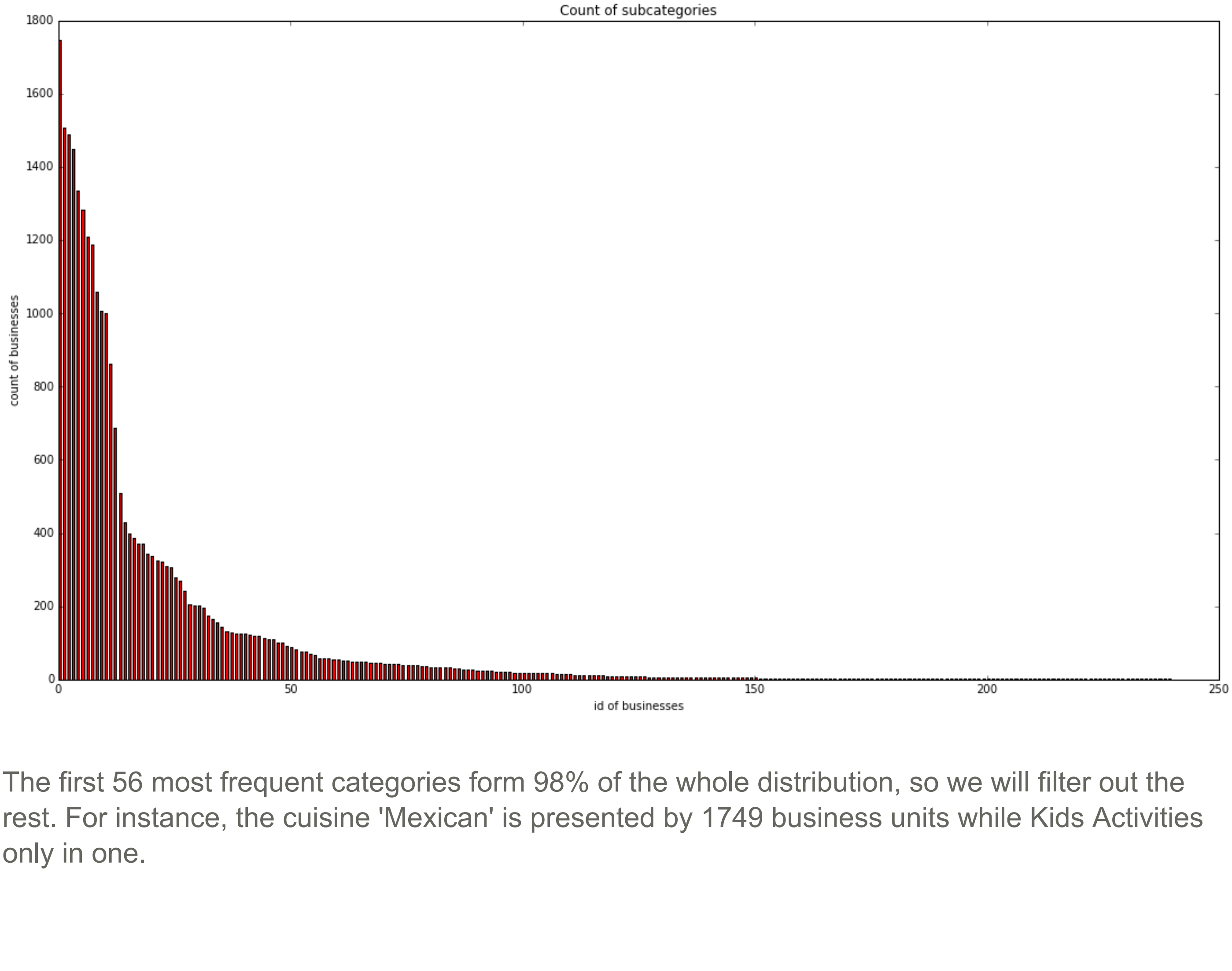
Below you can find more detailed description of each step made by me. The original iPython with source codes can be found in my dropbox: https://www.dropbox.com/s/oc1nnkm9s91ks9c/Capstone%20project%202.html?dl=0.

Data preparation: extraction set of all cuisines

We start the second task in the capstone project with standard step of Data Science: data preparation. Particularly we are going to load JSON file as Apache Spark DataFrame (https://spark.apache.org/docs/latest/sql-programming-guide.html) and extract set of all cuisines that are presented in 'categories' attribute. The total number of categories is **14303**.

The categories contain not only cuisines, but also not related to restaurants categories, for example, there are categories 'Dry Cleaning & Laundry' or 'Art Galleries'. We will filter out these categories using a guess that real cuisines should have many businesses associated to this category.

Below is distribution of categories and business.

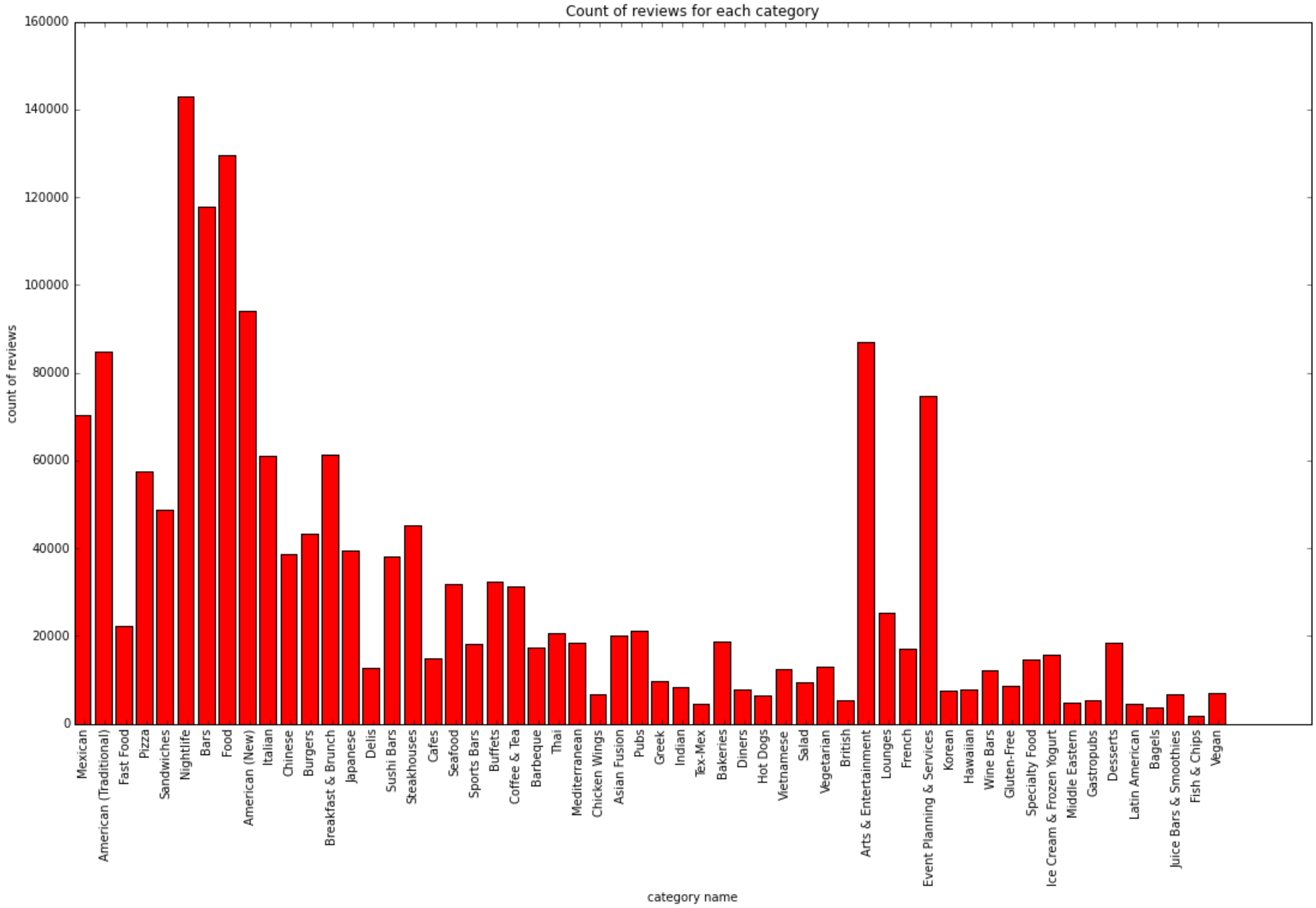


The first 56 most frequent categories form 98% of the whole distribution, so we will filter out the rest. For instance, the cuisine 'Mexican' is presented by 1749 business units while Kids Activities only in one.

Below is the list of selected categories:

```
[u'Mexican', u'American (Traditional)', u'Fast Food', u'Pizza', u'Sandwiches', u'Nightl
ife', u'Bars', u'Food', u'American (New)', u'Italian', u'Chinese', u'Burgers', u'Breakf
ast & Brunch', u'Japanese', u'Delis', u'Sushi Bars', u'Steakhouses', u'Cafes', u'Seafoo
d', u'Sports Bars', u'Buffets', u'Coffee & Tea', u'Barbeque', u'Thai', u'Mediterranean'
, u'Chicken Wings', u'Asian Fusion', u'Pubs', u'Greek', u'Indian', u'Tex-Mex', u'Bakeri
es', u'Diners', u'Hot Dogs', u'Vietnamese', u'Salad', u'Vegetarian', u'British', u'Arts
& Entertainment', u'Lounges', u'French', u'Event Planning & Services', u'Korean', u'Haw
aiian', u'Wine Bars', u'Gluten-Free', u'Specialty Food', u'Ice Cream & Frozen Yogurt',
u'Middle Eastern', u'Gastropubs', u'Desserts', u'Latin American', u'Bagels', u'Juice Ba
rs & Smoothies', u'Fish & Chips', u'Vegan']
```

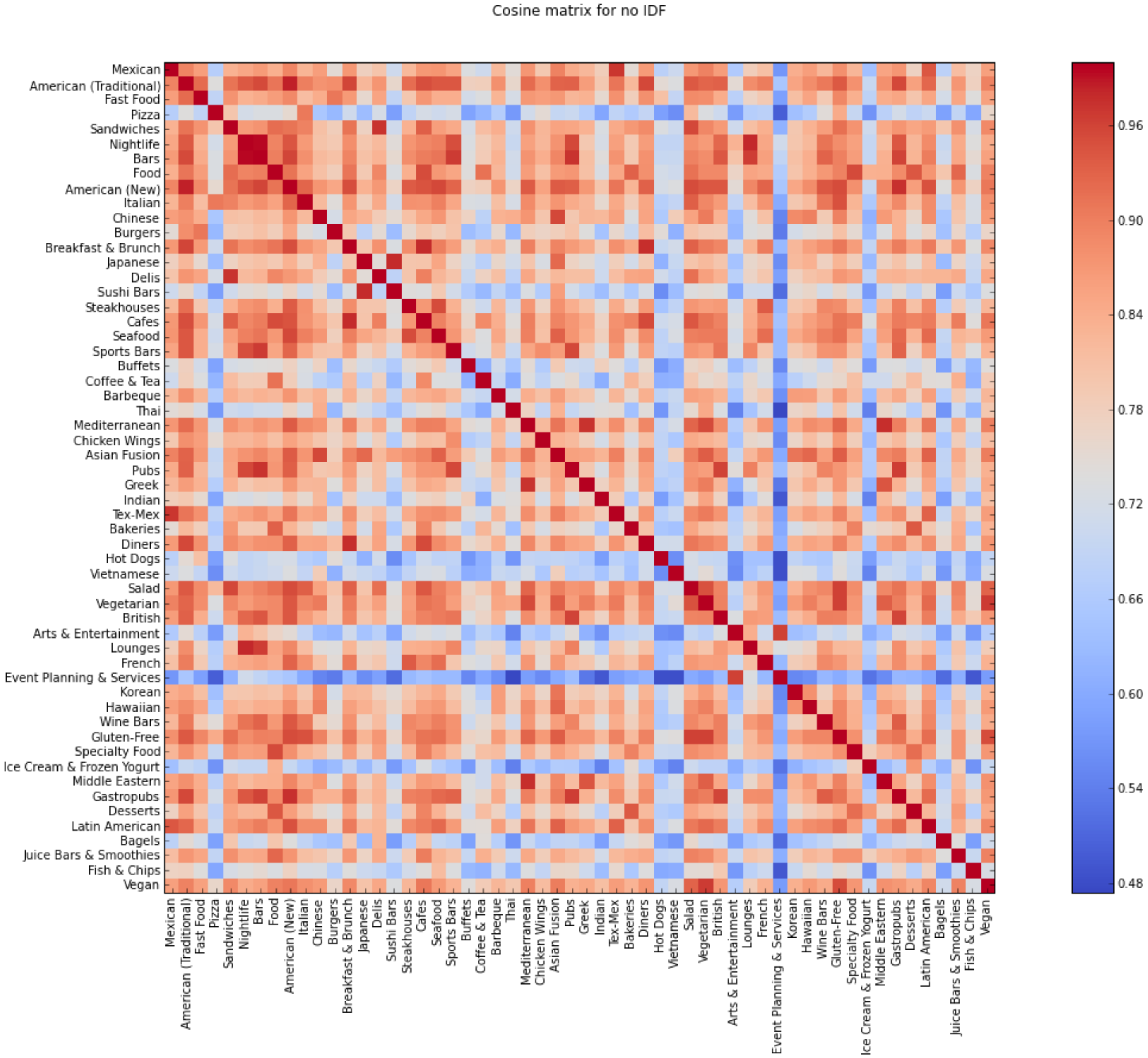
For each category I performed concatenation of each reviews into one text. The number of reviews for each category is presented below:



Feature extraction and clustering

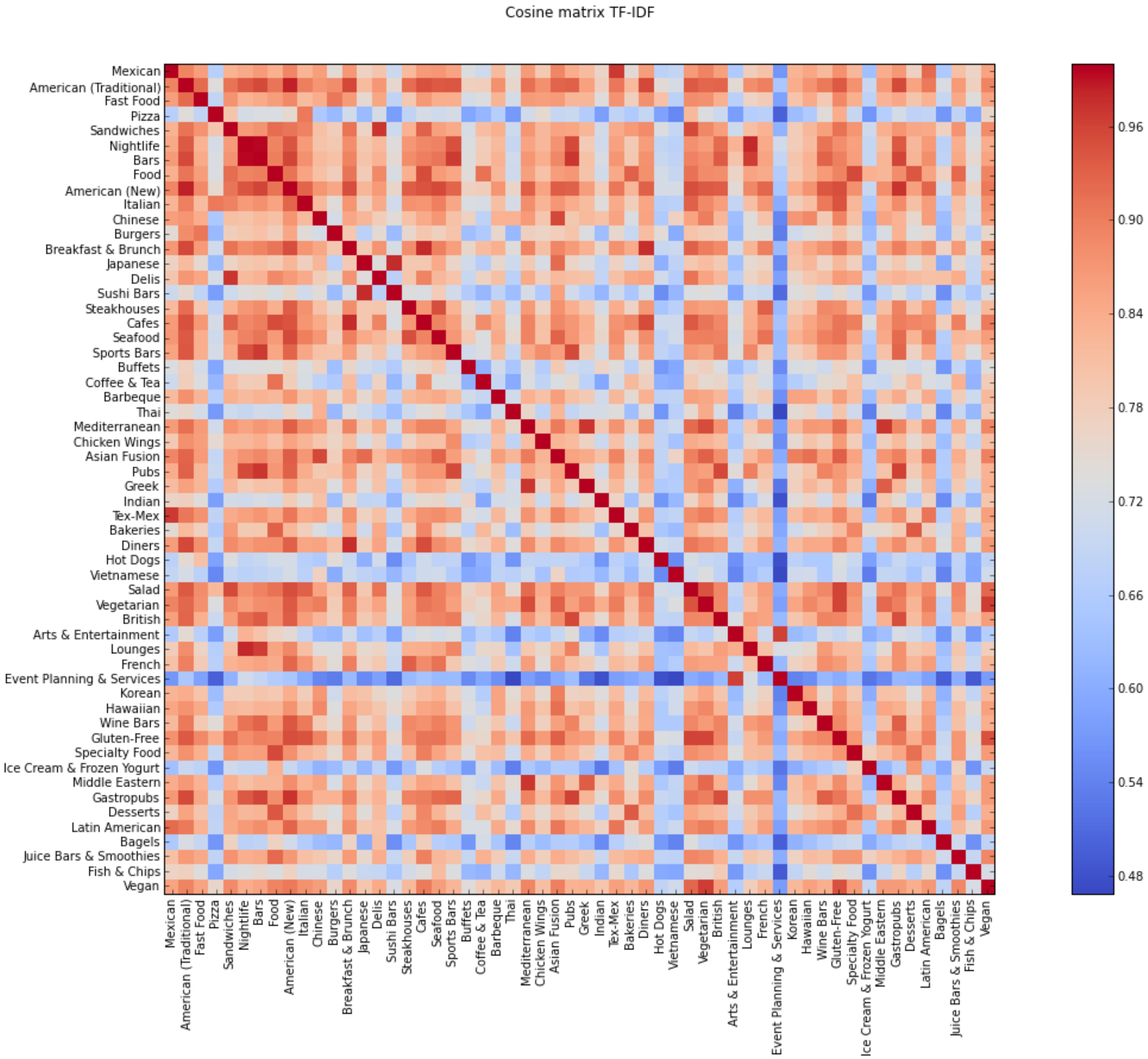
TF model

After combination set of reviews for each category and the next step is to build feature vector based on TF-IFD model. But for first step we will not use IDF. The similarity matrix is based on



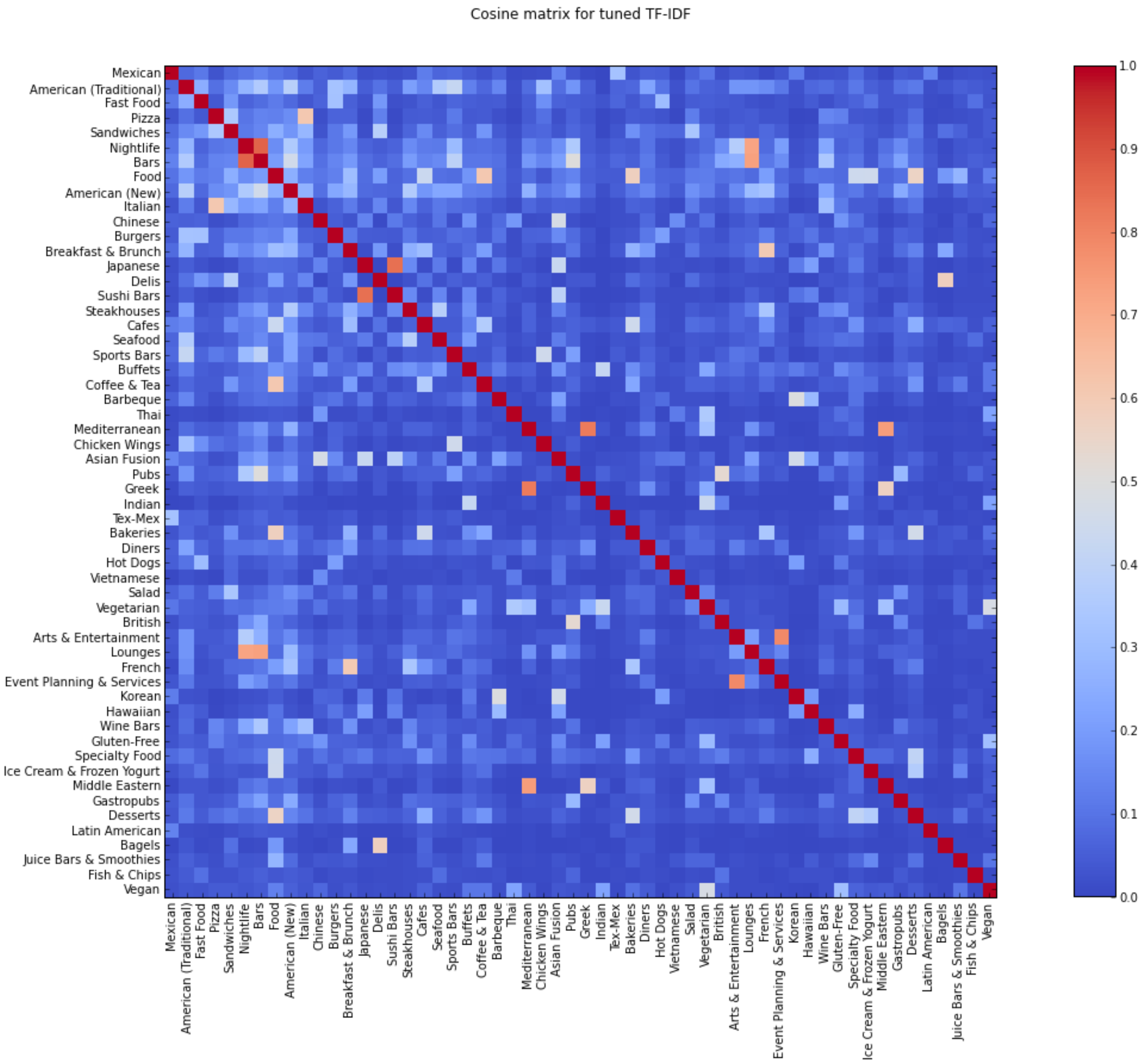
The result we've got is not so good, many categories appear to be quite similar based on the current similarity matrix. Further we're going to make steps to improve the result.

TF-IDF model



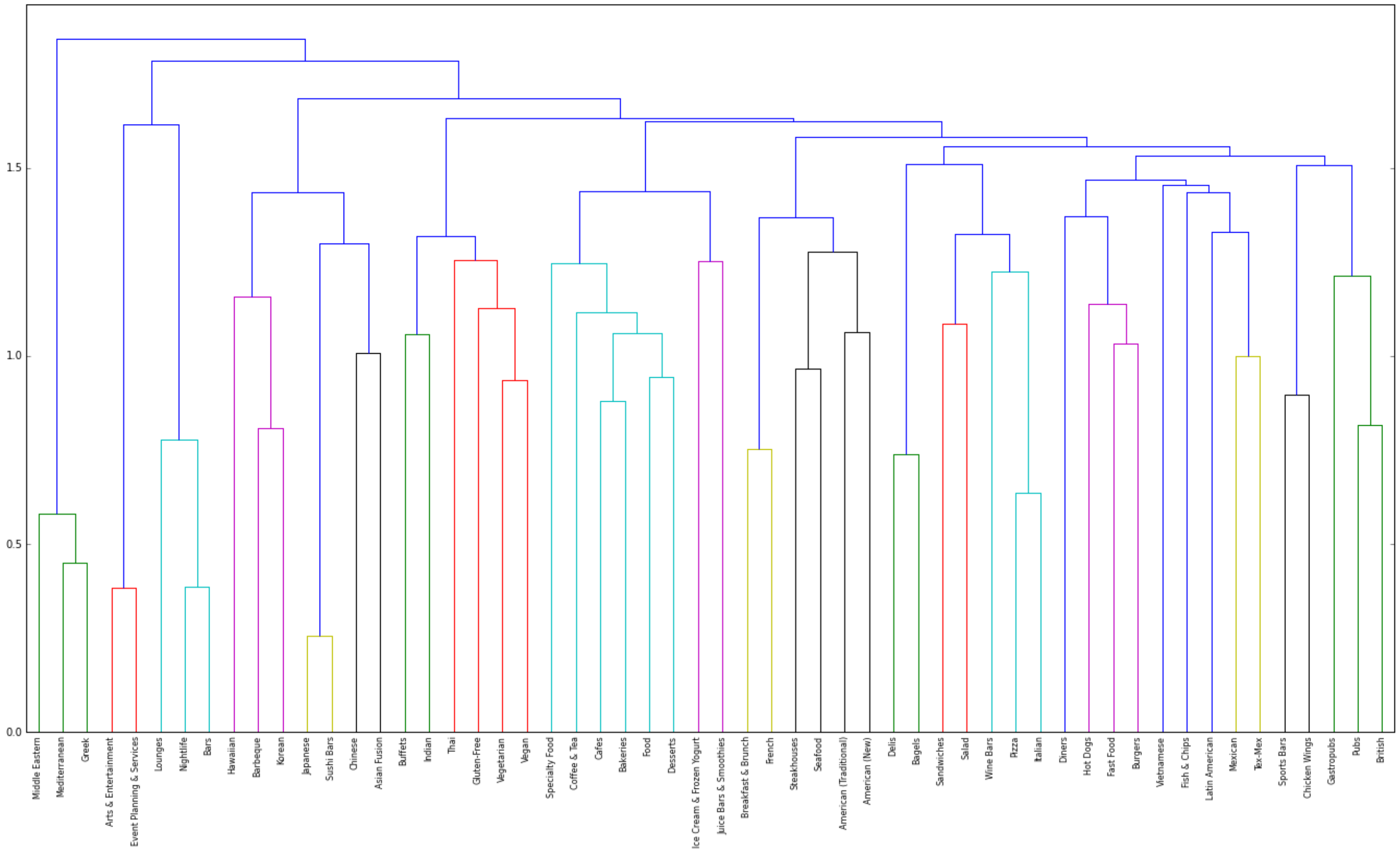
Tuned TF-IDF model

Based on previous similarity matrices we can see that many categories have quite high similarity measure. The reason is because built-in stop words from python implementation is quite general and work well only for general texts but we bump with many general words specific for restaurant reviews. Examples could be 'good', 'just' or 'food' that we observed in the previous task of the capstone project. From other side we need bear in mind that reviews could have words with misspellings but the IDF part will reward such words like this is a unique and very specific word that characterizes a text. So the next step was to build TF-IDF model with filtering of terms with TF less than 2 and higher than 0.8 proportion of documents.



The similarity matrix looks now significant better, we can observe long distance between quite different cuisines while categories that actually relate to a same restaurant type have still high similarity, for example 'Pizza' and 'Italian' have quite high similarity, or another example is 'Japanese' and 'Sushi bars' are extremely similar (the distance is about 0.9).

To confirm this statement, we can make the next step and perform clustering. I chose the hierarchy clustering and visualization as dendrogram since it provides the easiest to read view and emphasize what cuisines are quite similar and how close are a whole groups of cuisines.



We can see the result here is quite sensible. Let's follow when different cuisines are combined together during clustering. The first categories that has been combined were Japanese and Sushi Bars and indeed they offer almost identical dishes. The next candidate to combine were Nighlife and Bars, also very similiar type of places. The same think with Art & Entertainment and Event Planning. The next sensible combination is a combination of Mediaterian and Greek. From perspective of not so good combination can be Thai and a group of (Gluten-Free, Vegetarian, Vegan).

Adding pre-processing step before build TF-IDF model

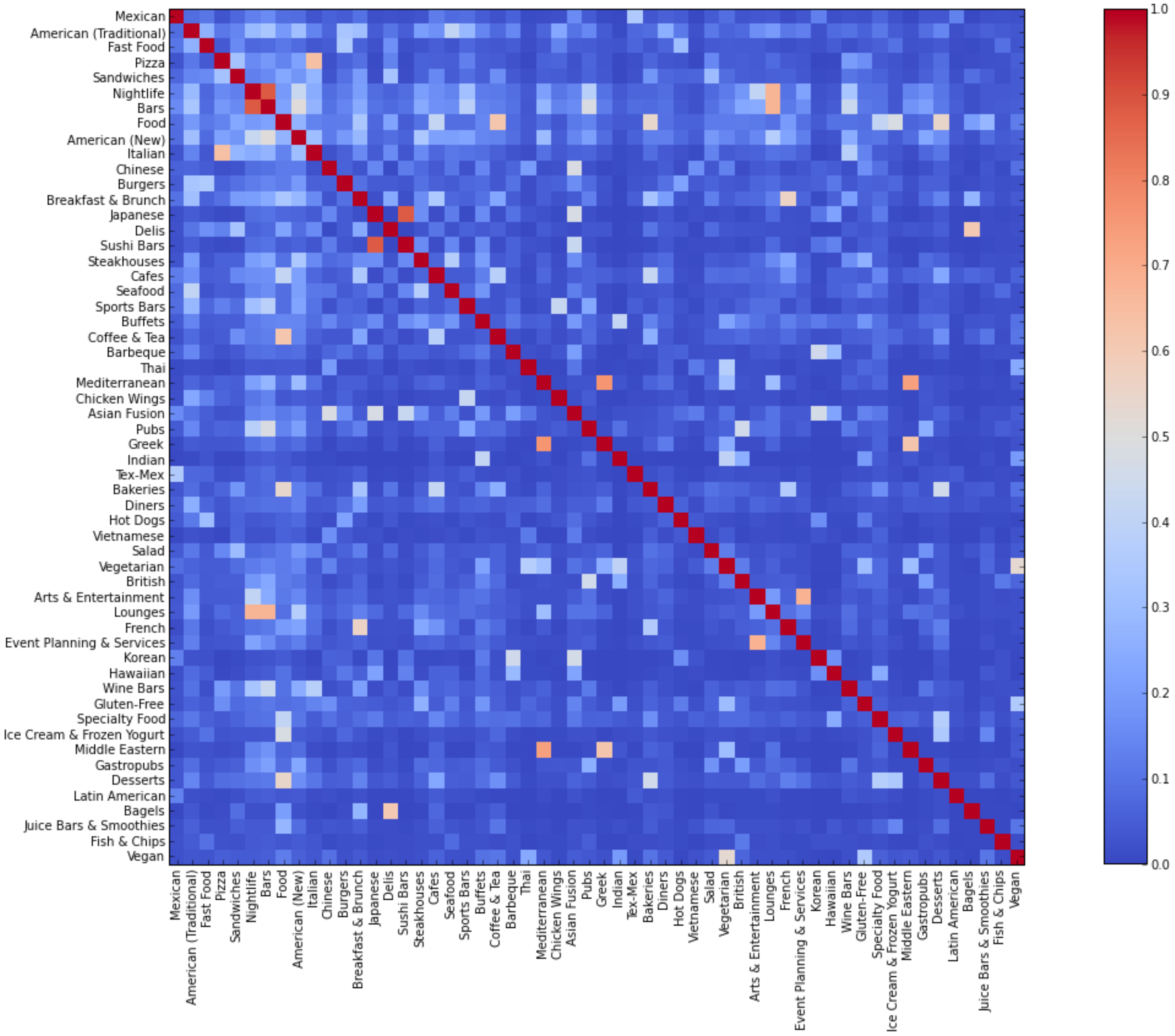
The idea of the next improvement is to add a pre-processing step in scope of which we'll delete punctuation and perform stemming. That should reduce the dimension of feature space and increase accuracy of comparison.

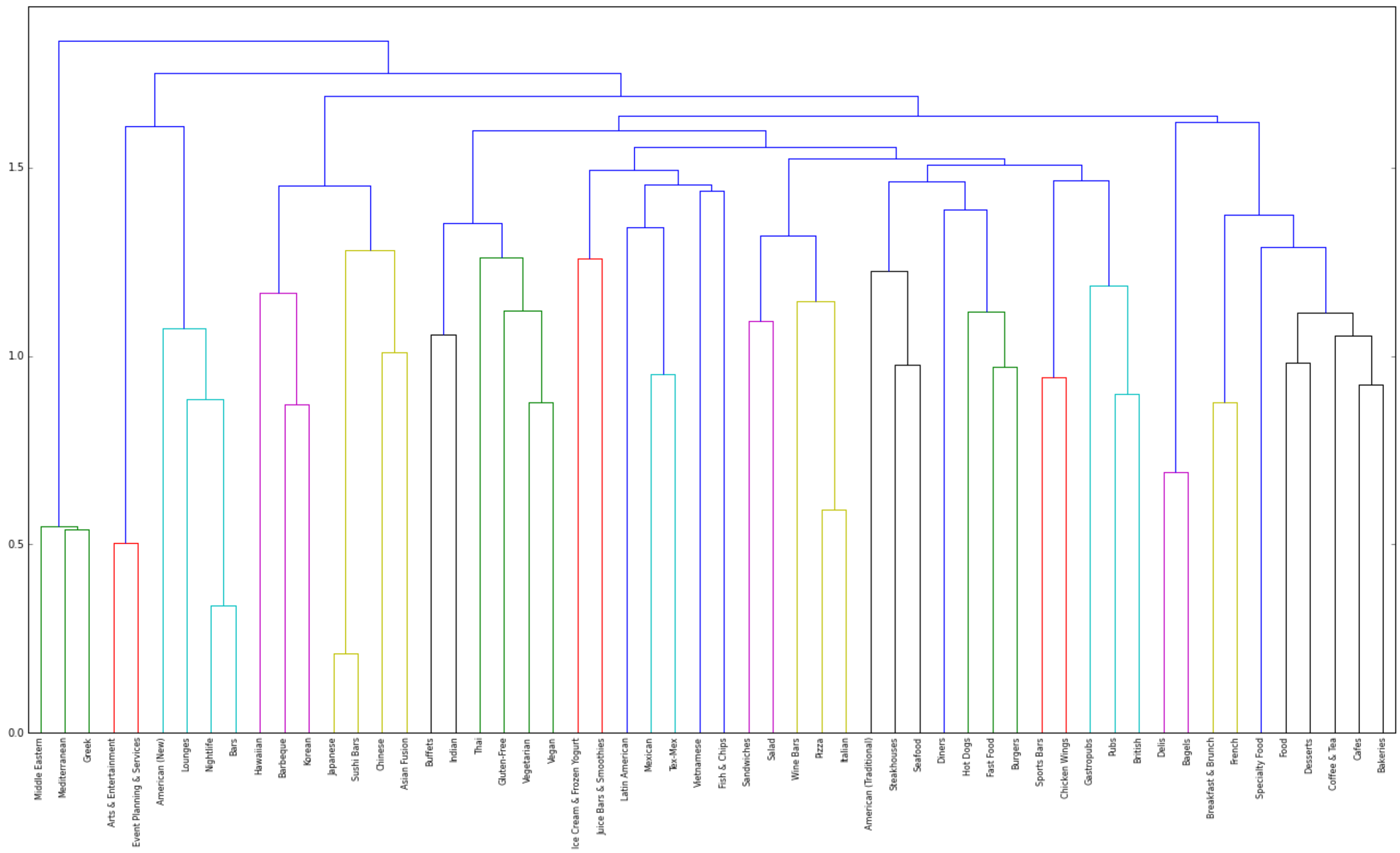
We used nltk library (in particularly stemming package http://www.nltk.org/api/nltk.stem.html). ntlk provides a series of Stemmers:

- The Porter Stemming Algorithm. Classical stemming algorithm published in 1980.
- The Lancaster Stemming Algorithm. This algorithm is much newer, published in 1990, and can be more aggressive than the Porter stemming algorithm.
- The WordNet Lemmatizer. it uses the WordNet Database to lookup lemmas. Lemmas differ from stems in that a lemma is a canonical form of the word, while a stem may not be a real word.

This report contains only Porter algorithm because Lancaster shows slightly worse hierarchy clustering, most likely due of ‘too aggressive’ stemming. The result for WordNet Lemmatizer is very close to Porter. You can find details and additional attempt to build TF-IDF without filtering but with Lancaster stemming in the iPython notebook (https://www.dropbox.com/s/oc1nnkm9s91ks9c/Capstone%20project%202.html?dl=0).

Cosine matrix for TF-IDF with Lancaster stemmer

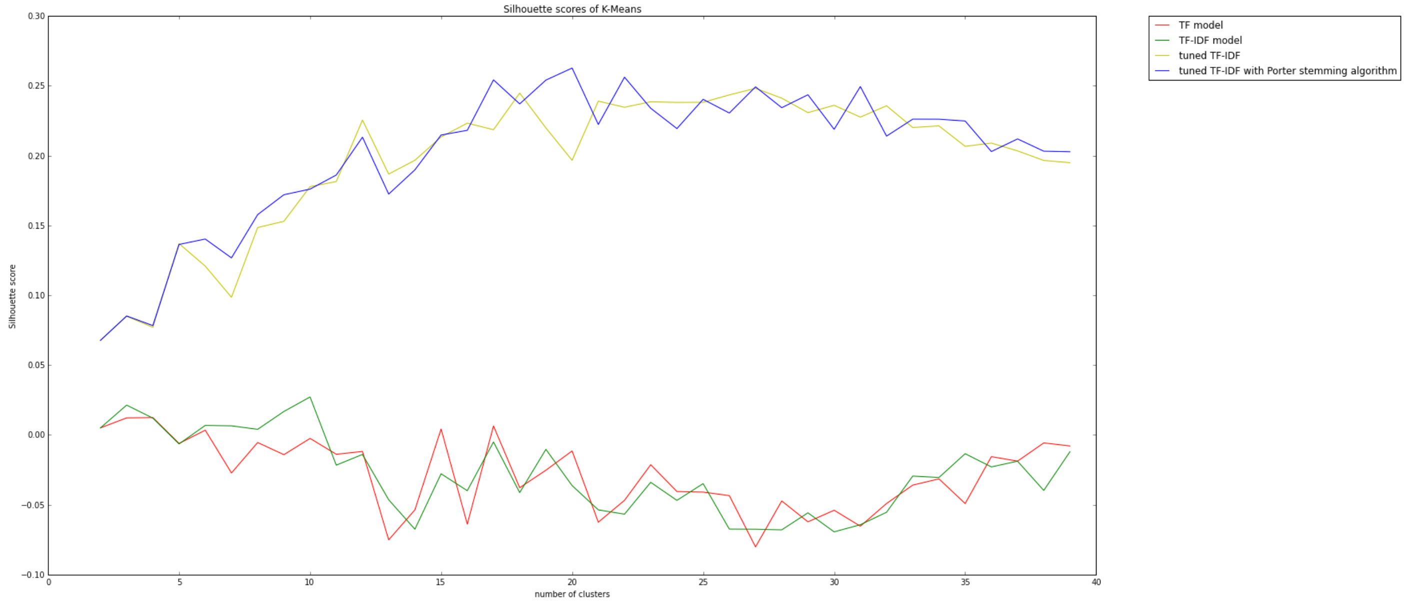




Measuring clustering's quality

We can check a quality of built clusters. Since there are no known ground truth labels, so the evaluation must be performed using the model itself. *Silhouette Index* is one of the most popular ways of measuring a particular clustering's quality. It measures how similar each pattern is to the patterns in it's own cluster as compared to patterns in other clusters. Silhouette score close to one means that the datum is appropriately clustered. If the score is close to negative one, then by the same logic we see that some items would be more appropriate if it was clustered in its neighboring cluster. A score near zero means that the datum is on the border of two natural clusters.

For clustering we're going to use an enhanced version of K-means algorithm named Mini Batch K-Means (http://scikit-learn.org/stable/modules/clustering.html#mini-batch-k-means). Silhouette score will be depicted as a plot score/number of clusters.



From the graph above we can see that TF and TF-IDF model are significant worse than tuned TF-IDF and TF-IDF after stemming. The maximum value of Silhouette Index reaches for option TF-IDF with stemming and for 19 clusters. Let’s construct these clusters:

```
cluster 1: Buffets, Indian
cluster 2: Mexican, Tex-Mex, Latin American, Fish & Chips
cluster 3: Sandwiches, Salad
cluster 4: Pubs, British, Gastropubs
cluster 5: Food, Cafes, Coffee & Tea, Bakeries, Specialty Food, Desserts
cluster 6: American (Traditional), Steakhouses, Seafood
cluster 7: Delis, Bagels
cluster 8: Mediterranean, Greek, Middle Eastern
cluster 9: Barbeque, Korean, Hawaiian
cluster 10: Sports Bars, Chicken Wings
cluster 11: Ice Cream & Frozen Yogurt, Juice Bars & Smoothies
cluster 12: Arts & Entertainment, Event Planning & Services
cluster 13: Japanese, Sushi Bars, Asian Fusion
cluster 14: Nightlife, Bars, American (New), Lounges
cluster 15: Fast Food, Burgers, Hot Dogs
cluster 16: Chinese, Thai, Vietnamese, Vegetarian, Gluten-Free, Vegan
cluster 17: Breakfast & Brunch, French
cluster 18: Pizza, Italian, Wine Bars
cluster 19: Diners
```