Dish Recognition

For recognizing different dishes in a cuisine, Indian cuisine was selected as more dishes in this cuisine were known. Since more dishes were known, it was easier to label some additional manual annotations for Task 3.1. It also helped in analyzing the results of running different tools on yelp reviews from restaurants serving Indian food for Task 3.2 .

Overall 12/10 score was achieved towards the end after tweaking different parameters for ToPMine and using refined labels in SegPhrase tool. The description below will specify what data and tools (along with the parameters) were used and corresponding score obtained for Task 3.2

Following approaches were tried:

- Word Association Mining (MeTA)
- ToPMine
- SegPhrase

Initial attempts with MeTA and SegPhrase [score obtained: 5/10]

The Indian.txt file was used from the Yelp sample data provided by staff.

Word Association Mining (MeTA)
For word association mining, the code from Text Mining and Analytics course was used (provided in programming assignment Task 2 section to identify syntagmatic words using word association mining]

Words with document frequency at least 2 were selected, filter was written to find syntagmatic words only for:
“chicken”,”veg”,"garlic","lamb","fish","puri","curri","spicy","grill","fry","baked","naan,"paneer","aloo","curr", "goat", "beef","roti", "fri", "chaat", "methi", "palak", "butter", "onion","pav". The intuition was that output will contain most likely all chicken dishes or vegetarian or fish dishes. It should also give different type of breads and other fast food dishes.

The output contained some good dishes like chicken tikka, butter chicken or garlic naan but also contained lot of other phrases like chicken delicious, naan order (which makes sense but was not useful for this task). The entire output list was submitted to test the scoring mechanism and the score came low (5/10) as expected as it did not have enough quality dish phrases. It was probably also because of low document frequency threshold (low value here was selected to obtain as much related phrases as possible - with high threshold not a lot of phrases were getting generated)

SegPhrase:

The manually annotated file form Task 3.1 was used in labels file to run SegPhrase tool (provided by the course staff for this task)

Following parameters were changed to run **train.sh** file from SegPhrase tool:
RAW_TEXT='data/Indian.txt', DATA_LABEL='data/Indian.label' and AUTO_LABEL=0 ….**[i]**
All other parameters were kept as default. The phrases obtained in ranking.csv file under results folder were not of good quality. The total number of phrases returned were also not very high (~ 5,200 phrases). Some

of the popular Indian dishes were present like chicken tikka masala and lamb rogan gosh but there was large amount of unrelated phrases like vegas, great service and delicious food that was not about dish names.
The output from ranking.csv file was merged with output dishes from association mining and still the score remained low (5/10) because of large number of phrases completely unrelated to dishes.

Final attempts with TopMine and SegPhrase [Score obtained: 12/10]

First change was done in Indian.txt reviews file. It was observed that each review was not on single line as expected. There were some formatting issues with the sample data provided. Changes were made to python script **py27_processYelprestaurants.py** such that the reviews of particualr cuisine were formatted correctly (i.e. ‘\n’ character in a review were replaced by empty character so that entire review will be on one line) before outputting to the cuisine specific file. Also categories only associated with Indian were parsed which made it easier to extract all of the reviews from Indian restaurants without having to sample and hope that Indian cuisine gets selected.The revised **Indian.txt** file was used for dish recognition using ToPMine and SegPhrase.

TopMine:

Following parameters were changed in **run.sh** file:

```
inputFile='../rawFiles/Indian.txt'
# minimum phrase frequency
minsup=8
#maximum size of phrase (number of words)
maxPattern=3
#Two variations of phrase lda (1 and 2). Default topic model is 2
topicModel=2
numTopics=8
#set to 0 for no topic modeling and > 0 for topic modeling (around 1000)
gibbsSamplingIterations=1000
```

Various number of topics and number of gibbs sampling iterations were tried. It was observed that with very low number of gibbs sampling iterations the dish phrases were getting mixed up restaurant service. If this happened, it would have been tedious to extract just the dishes phrases from other phrases for using as labels in SegPhrase tool. Hence increasing number of iterations helped separate the phrases for dishes in different topic than restaurant service phrases. Topics were much more distinct (easily identifiable - e.g. dishes, service, ambience) with increased number of gibbs sampling iterations.

The topic 5th and 7th were observed to have all the phrases that contained a lot of the Indian dishes. Most of the phrases from these topics were added to the **Indian.label** file. Indian.label was edited multiple times to make sure only correct Indian dishes are labelled as 1. For using the dish phrases directly from TopMine topic output, **topTopics.py** file was edited to output phrase and corresponding true label 1 (instead of frequency). For submitting the output to the auto grader, topTopics.py and topPhrases.py was edited to output just the phrases **(cand[0])** and no frequency/label was outputted with it.

SegPhrase:

Indian.label edits:

The label file was edited so that some words like **and/or** and dishes that don't make sense or are too generic were removed. A lot of effort was made to review this file as it was observed that erroneous labels can lead to erroneous output phrases.

The new Indian.label and Indian.txt files were now used to run **train.sh in SegPhrase tool.** The parameters were same as described in [i] (Page 1).
RAW_TEXT='data/Indian.txt', DATA_LABEL='data/Indian.label' and AUTO_LABEL=0 .
No other parameters were changed.

The ranking.csv file in results folder now had a lot of good quality phrases. The extra probability column from this file was removed and the words were merged with topPhrases.txt obtained from ToPMine. The extra phrases from ranking.csv list was removed to satisfy the 10,000 word limit for Task 3.2 auto grader.
The merged file was then submitted and 12/10 score was obtained.

Recognized Dishes (Observations):

A lot of **popular** Indian dishes were recognized correctly from ToPMine and SegPhrase:
palak paneer, butter chicken, basmati rice, garlic naan, saag paneer, chicken korma, chicken vindaloo, malai kofta, gulab jamun, mango lassi, paratha, aloo gobi,curry chicken,lamb chops,goat curry.

Some **interesting** dishes (unexpected or tough to mine) were:- seekh kabob, Halwa carrot, tamarind and mint chutneys, paneer tikka, biryani chicken, pappadams, samosa chaat, kheer, pakoras, chickpea, masala dosa, lentil soup, makhani chicken

As expected, apart from Indian dishes there were lot of normal phrases which were not even remotely related to the dishes. The recall percentage (percent of dish phrases compared to normal phrases) did not look that good. But the fact that phrases could be outputted and some of them are perfect Indian dishes seem like a **satisfactory** result. Once the higher frequency phrases based on the results are extracted, most of them can be considered as dishes. This can then be used to explore which dishes were associated with good or bad reviews and then can be used in scoring for Task 4 - Dish and Restaurant recommendations for a cuisine.

This would have been very difficult task with algorithms that use simple bag of words (specifically unigram) model. TopMine and SegPhrase tools definitely are good starting point to start mining **good phrases** from text documents.

References:

- [1] El-Kishky, Ahmed, et al. "Scalable topical phrase mining from text corpora." *Proceedings of the VLDB Endowment*, 8.3 (2014): 305-316.
- [2] Jialu Liu*, Jingbo Shang*, Chi Wang, Xiang Ren and Jiawei Han, "Mining Quality Phrases from Massive Text Corpora," *Proc. of 2015 ACM SIGMOD Int. Conf. on*

Management of Data (SIGMOD'15), Melbourne, Australia, May 2015. (* equally contributed)

- ToPMine and SegPhrase tools used from Task 3 Resources section.