# NLP Driven Approach for Improving Businesses


**S Gaopande**
M.S Computer Science
sulagaop@iu.edu

**A Saikia**
M.S Data Science
arsaikia@iu.edu

**N Thakurdesai**
M.S Computer Science
nthakurd@iu.edu

**F Thomas**
M.S Data Science
frathom@iu.edu



May 2, 2020


## 1 Introduction

Yelp reviews are one of the most useful resources people rely on in their hunt for cuisines, dishes and restaurants. Yu et al. [1] mentions independent restaurants can expect almost 59% bump in revenue with a 1-star increase in rating. Natural language processing and machine learning can be applied to extract actionable insights from the customer reviews about their experiences. If restaurant owners can gauge user sentiment about their restaurant, identify areas where their business is lacking, which dishes are doing well and what customers are saying about their competitors - they can take appropriate steps to improve their business. This project aims to analyze restaurant reviews on Yelp via topic modeling to determine aspects of restaurant that users dislike. We will also try to capture segments of restaurants in a city based on cuisines, restaurant attributes, popularity and customer sentiment using clustering techniques. Finally, we use sentiment analysis, topic modeling and named entity recognition(NER) to identify popular dishes in restaurant.

## 2 Related Works

In [1] the authors tried to find negative and positive aspects by associating polarity score to each word in a review after building a sentiment analysis model. The author in [2] tries to use topic modeling on positive comments to find similar restaurants. Although we couldn't find examples of segmentation for restaurants, in [3] the authors perform clustering of schools and universities based on different attributes.

## 3 Dataset Description

The data was sourced from the online Yelp Dataset Challenge[4], consisting of six parts which provides us with attributes like hours, parking, availability, and ambience for over 190k businesses, 6.6 million customer reviews as well as 1.2 million tips by 1.6 million users across 10 metropolitan cities in US. The total size of data is about 8 GB. For this project, we mainly focused on three files - yelp_business.csv which contains different restaurant attributes like business id, name, postal code, rating, category etc, yelp_review.csv which contains reviewer id, business id corresponding to the review, the text, rating, date etc and yelp_tip.csv which contains reviewer id, business id, tip (text like - 'Kung pao chicken is a must try'), date etc. We also use checkin.json from [5] which has check-in information for various businesses.

## 4 Problem Statement 1

Find areas of improvement for restaurants using customer reviews (topic modeling) e.g - service, ambience, parking

### 4.1 Discussion

#### 4.1.1 Pre-processing and Feature Engineering

The idea was to help restaurant owners understand sentiment of customers towards their restaurants and identify potential areas of improvements.

Table 1: Top 5 cities with most restaurant reviews

| City | # of restaurant reviews | # of restaurants |
|---|---|---|
| Las Vegas | **795,581** | 3,990 |
| Pheonix | 284,627 | 2,571 |
| Toronto | 254,479 | 4,968 |
| Scottsdale | 149,430 | 897 |
| Charlotte | 138,974 | 1,860 |

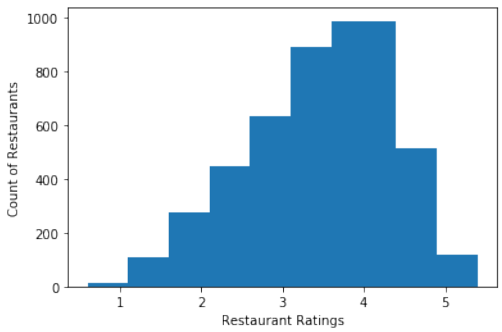


Figure 1: Distribution of ratings for different restaurants in Las Vegas

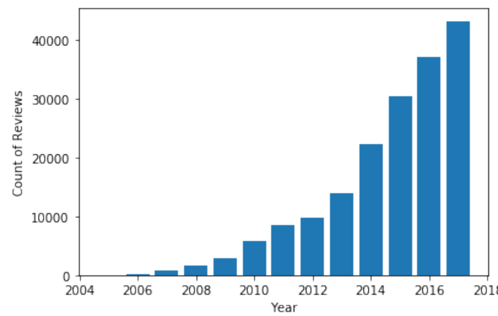


Figure 2: Distribution of reviews for 1000 restaurants in Las Vegas across all years

First, we filtered restaurants which are open and ignored restaurants which are closed. Since, our data consists of restaurants across different cities, we thought of continuing our analysis for the city with most restaurant reviews. A quick analysis showed that restaurants in Las Vegas has the highest number of reviews (Table 1).

We observed that total number of open restaurants in Las Vegas was 3990. The distribution of ratings for these restaurants can be seen in Figure 1. We also observed on an average each restaurant has 200 reviews. To reduce the number of reviews from 800k we sampled at random 1000 out of these 3990 restaurants. We also looked at the how the reviews for these restaurants are distributed across years. Based on Figure 2. we decided to continue our analysis with reviews written in 2016-2017 and discard old reviews.

Next we wanted to look at the distribution of ratings for these reviews (Figure 3). We observed that although most of the reviews were positive, for restaurants to really understand the potential areas of improvement, it was more prudent to analyze the negative reviews. We then extracted the negative reviews having rating 1 or 2.

As a last step before we tried to extract topics we went ahead and pre-processed the text. As natural language can contain different combinations of characters, punctuations and symbols it is imperative that we ensure that the text is clean for the model to make sense of it. We removed punctuation marks, converted all text to lower case, removed non-english words and lemmatized each word to verb form. Word cloud of the processed text is shown in Figure 4.

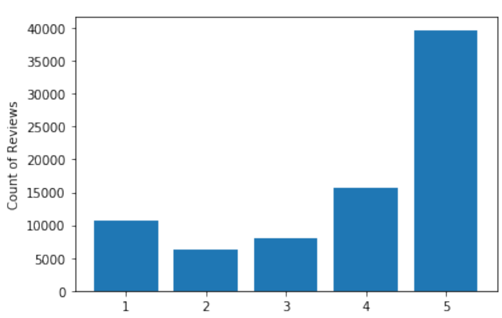


Figure 3: Distribution of reviews across different ratings


Figure 4: Word cloud of the processed text

Table 2: Top 10 words for different topics given by LDA

| Topic Index | # Top 10 Words |
|---|---|
| 0 | place, service, time, say, food, come, just, tell, card, like |
| 1 | order, pizza, time, food, say, just, come, location, tell, service |
| 2 | food, good, like, place, taste, chicken, order, eat, just, come |
| 3 | say, just, like, make, come, tell, know, don, order, place |
| 4 | food, come, service, table, time, order, wait, place, server, just |

Table 3: Top 10 words for different topics given by NMF

| Topic Index | # Top 10 Words |
|---|---|
| 0 | come, table, say, wait, just, time, tell, sit, server, leave |
| 1 | pizza, cheese, delivery, crust, pepperoni, slice, driver, sauce, cold, burn |
| 2 | chicken, taste, like, good, eat, sauce, place, just, rice, meat |
| 3 | food, service, good, horrible, place, worst, terrible, bad, great, slow |
| 4 | order, time, location, wrong, drive, say, tell, delivery, wait, phone |

## 4.2 Results

### 4.2.1 Latent Dirchlet Allocation

To convert our text into feature vector we tried CountVectorizer and TfidfVectorizer from sklearn[6]. After the pre-processing steps we had around 16k reviews. We first tried Latent Dirchlet Allocation (LDA) to get topics with different values of number of topics (k). We observed that large values of k gave less distinct topics subjectively. With k = 5 and CountVectorizer as feature generator having 1000 features, top 10 words for each topic can be seen in Table 2. We also observed there is a lot of overlap between the words across different topics.

### 4.2.2 Non-Negative Matrix Factorization

Since LDA was not giving satisfactory results, we then tried NMF to extract the topics. We represented each review as a TF-IDF vector using TfidfVectorizer with max_df = 0.95, min_df = 10, n_features = 1000. We tried different values of k = 5, 6, 7. Since we objectively cannot judge the best value of k, we were looking for a value of k which gave diverse, coherent topics and made sense. With k = 5, the top 10 words for each topic can be seen in Table 3.

By comparing Table 2 and Table 3 we can immediately observe that NMF is performing better with less overlapping words. Since one review can belong to multiple topics we calculated dominant topic for each review and looked at samples of reviews belonging to each topic. After randomly scanning through couple of samples across different topics we labeled the topics as seen in Table 4.

Since Topic 0 has words like wait, time, table, server we deduced it was probably referring to table wait times. Similarly, words like pizza, delivery, pepperoni, cheese probably refers to pizza joints and delivery. For Topic 2 with words like chicken, taste, rice, meat we thought it refers to food items. Topic 3 was really distinct having words like service, horrible, slow so we tagged this as something related to customer service. For the last topic we deduced it refers to wrong orders, deliveries gone wrong or there is some issue with the location of the restaurant.

Table 4: Identified topics after examining topic vectors and samples

| Topic Index | # Topic Label |
|---|---|
| 0 | Wait-time |
| 1 | Pizza/Delivery |
| 2 | Chicken/Rice/Sauce/Taste/Meat |
| 3 | Customer Service/Slow |
| 4 | Orders/Location/Delivery |

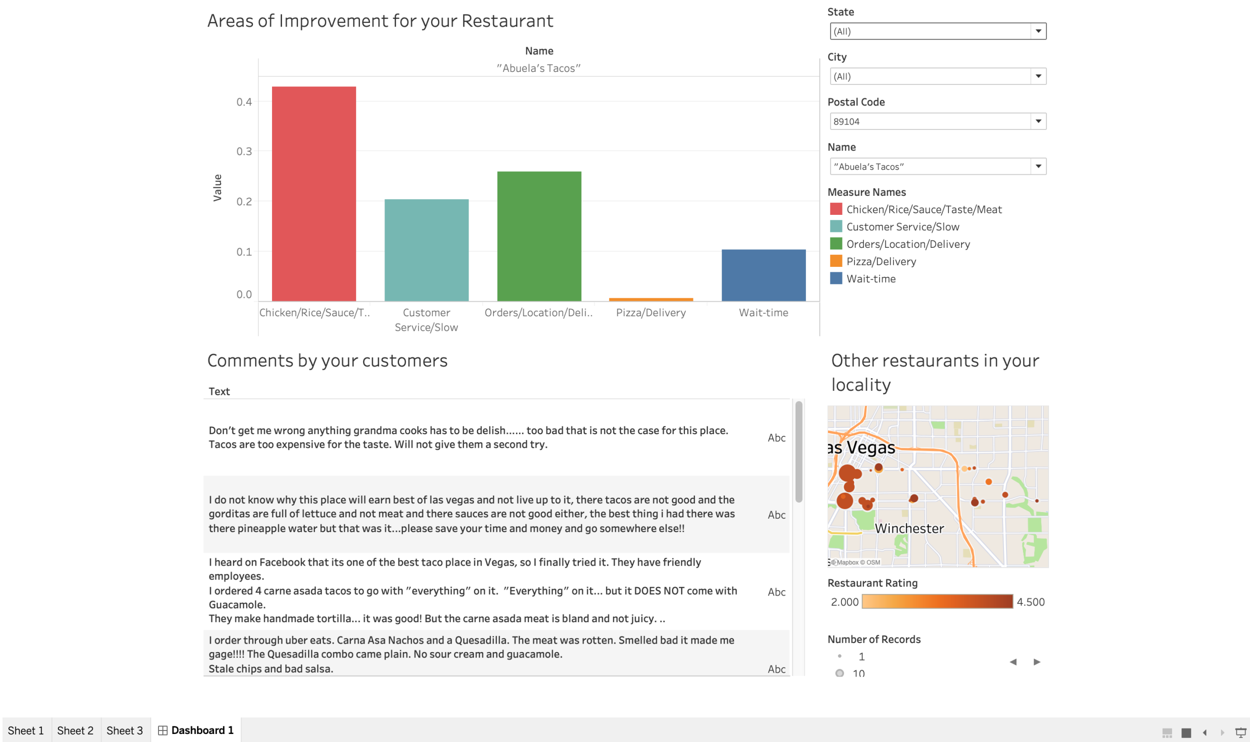


Figure 5: Dashboard for restaurant owners to understand their business

### 4.3 Conclusion and Next Steps

Now that we have a way to associate a topic to each negative review, we needed to figure out a way to put it all together so that restaurant owners can gain value out of it. We decided to build a dashboard where restaurant owners will be able to select their restaurant, gain insights and develop strategies to improve their business. A snapshot of the dashboard can be found in Figure 5. The dashboard gives the restaurant owner information on which areas are the customers complaining about and what the specific complaints are. We also have a map view of other restaurant in the locality with their respective ratings.

There were several bottle necks we faced while working on this problem. From a data mining standpoint, since we had huge number of reviews for different restaurants across different cities we had to restrict ourselves to one city and only 1000 restaurants. Also since topic modeling is an unsupervised technique, and since there was no objective way to access the correct value of k - there was a lot of manual interpretation required. The topics we got were not individual distinct topics like we initially expected but a mixture of topics.

As next steps, we could try using our approach for restaurants in different cities and analyze the results. Also, we could try tuning hyper-parameters of the algorithms(LDA and NMF) and see if we can get better topics.

## 5 Problem Statement 2

Identify segments of restaurants in a city/neighborhood based on cuisine, customer sentiment, user check-in patterns and other restaurant attributes. (clustering) e.g - trending, budget-friendly

### 5.1 Discussion

#### 5.1.1 Pre-processing and Feature Engineering

The idea was to create clusters of restaurants based on different restaurant attributes like – rating, count of reviews, expensiveness, cuisine availability and user check-in patterns. If we can segment restaurants then each restaurant

Table 5: Top 5 cities with most open restaurants

| City | # of restaurants |
|---|---|
| Toronto | **4,968** |
| Las Vegas | 3,990 |
| Montreal | 2,596 |
| Pheonix | 2,571 |
| Charlotte | 1,860 |

within a segment is a competitor. Since we have already observed from Table 1. there can be thousands of restaurants in a city, it can be very easy for restaurants to be unaware of their competitors. Knowledge about competitors and understanding how competitors are faring can help the owners develop better and informed strategies to improve their business. To create segments we needed different attributes of the restaurants. Although some of these features were available directly from the dataset, some of them had to be created after parsing data(e.g. – rating, count of reviews, expensiveness factor) from the Yelp API or after considerable feature engineering steps.

First, we filtered restaurants which are open and ignored restaurants which are closed. Since, our data consists of restaurants across different cities, we thought of continuing our analysis for the city with most restaurants. A quick analysis showed us in our data Toronto, Canada has the greatest number of restaurants (Table 5)

Next, we wanted to add check-in information at a restaurant level to capture user check-in behavior. In the raw data [5] the check-in information was present as a series of time stamps for each restaurant.


```
{
    business_id: tnhfDv5Il8EaGSXZGiuQGg
    date: 2016-04-26 19:49:16, 2016-08-30 18:36:57, 2016-10-15 02:45:18
}
```


We identified and created following features which might help us capture the user check-in patterns from this data –

- Avg. checkins in the Mornings during weekends
- Avg. checkins in the Afternoons during weekends
- Avg. checkins in the Evenings during weekends
- Avg. checkins in the Night during weekends
- Avg. checkins in the Late Night during weekends
- Avg. checkins in the Mornings during weekdays
- Avg. checkins in the Afternoons during weekdays
- Avg. checkins in the Evenings during weekdays
- Avg. checkins in the Night during weekdays
- Avg. checkins in the Late Night during weekdays

We also observed that most of the restaurants did not have a steady flow of customers and 70% of restaurants had less than 50 reviews. This led us to believe out of the 4.9k restaurants in Toronto, most of them do not have an active customer base. To only keep the restaurants with an active customer base we continued our analysis with restaurants having atleast 50 reviews. Finally we had around 900 restaurants to segment.

Next, we wanted to have cuisine information for every restaurant, i.e. – the cuisines that are available for each restaurant. We performed a quick EDA to identify the top 30 popular cuisines in terms of restaurants serving them. We then created a one hot encoded representation of the cuisines. The following cuisines were considered – Nightlife, Bars, Canadian, Chinese, Italian, Japanese, American, Indian, Mexican, Thai. Since one restaurant can serve multiple cuisines, we created an identifier for total number of cuisines as well.

The final set of features we had at a restaurant level were as follows

- Rating
- Count of reviews
- Expensiveness
- Number of cuisines served
- Check-in information (as listed above)
- Cuisine Flags

#### 5.1.2 Distance Metric

Now, that we were done with the necessary pre-processing and feature engineering steps – we were ready to perform the clustering analysis. Since our data is a mix of continuous, nominal and ordinal data, clustering algorithms that work

Table 6: Feature subsets for clustering

| Subset Index | Features | # of Features |
|---|---|---|
| 0 | Rating, Count of reviews, Expensiveness, Number of cuisines served | 4 |
| 1 | Features in Index 0 + Check-in information | 14 |
| 2 | Features in Index 0 + Cuisine Flags | 14 |
| 3 | All Features | 24 |

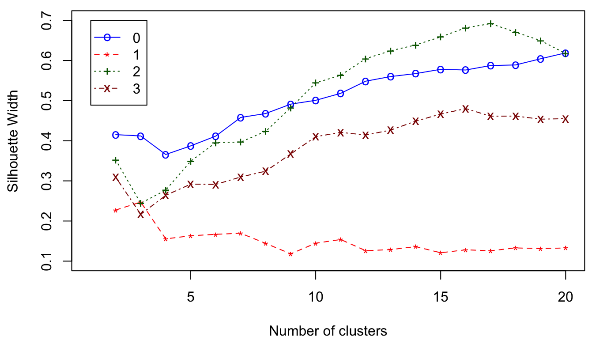


Figure 6: Silhouette analysis for different feature subsets after PAM

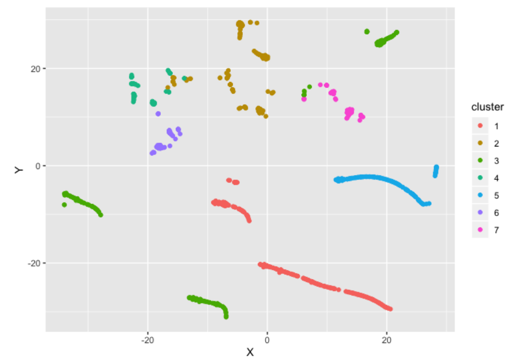


Figure 7: Visualization of the clusters with feature subset 0 and 7 clusters after PAM

with Euclidean distance like k-means will not work. Preliminary research[3][7] showed that for such data with mixed data types Gower distance is a reliable metric of (dis)similarity that can yield sensible results.

### 5.1.3 Feature Selection

Since clustering is an unsupervised approach, therefore to select the best subset of features we planned to try different combinations of features. The different subset of features we tried can be seen in Table 6.

## 5.2 Results

### 5.2.1 Partitioning Around Mediods (PAM)

The first algorithm we tried was Partitioning Around Medoids (PAM) a.k.a k-mediods. Although identical, both k-means and k-medoids differ on the fact k-means has cluster centers defined by Euclidean distance (i.e. centroids), while cluster centers for PAM are restricted to be the observations themselves (i.e. medoids) [3]. After deciding a distance/(dis)similarity metric and a clustering algorithm, next step was to decide the number of clusters (k). We then looked at the silhouette width – a validation metric which is an aggregated measure of how similar an observation is to its own cluster compared its closest neighboring cluster. The metric can range from -1 to 1, where higher values are better [3].

We visualized the silhouette width for the different feature subsets by varying the number of clusters (Figure 6). Increasing number of clusters will almost always lead to better silhouette score, but that does not make sense or satisfy the objective. Long story short, we should pick a meaningful number that is simple and equally as good. Feature subset 0 with 4 features gave highest silhouette width for smaller values of k. We decided to go with k = 7.

2D visualization of the clusters with k = 7 and feature subset 0 can be seen in Figure 7. Although not perfect, colors are mostly located in similar areas, confirming the relevancy of the segmentation. Each restaurant within a cluster are competitors in some sense.

### 5.2.2 Hierarchical Clustering

We also tried hierarchical agglomerative and divisive clustering. The advantage of these techniques is they don't require the number of clusters to be specified beforehand. The dendrograms for both these algorithms for different types of linkages can be seen in Figure 8 and Figure 9. Ideally we want a clustering technique which creates uniform clusters. We can see that agglomerative clustering with ward linkages is splitting the data more uniformly as compared to other

Table 7: Agglomerative coefficient for different types of linkages

| Linkage Type | average | single | complete | ward |
|---|---|---|---|---|
| Agglomerative coefficient | 0.9860011 | 0.978832 | 0.9936436 | **0.9984664** |

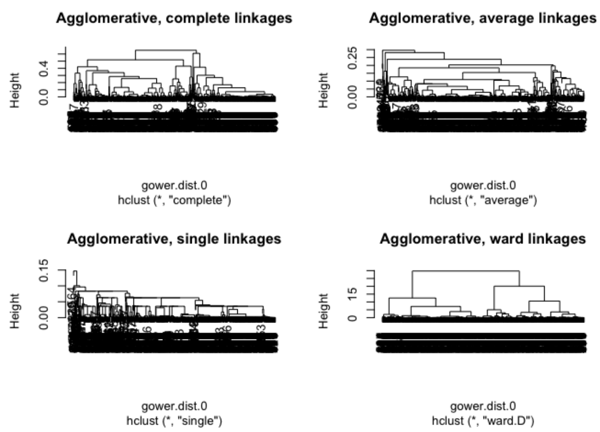


Figure 8: Dendrograms for Agglomerative Clustering

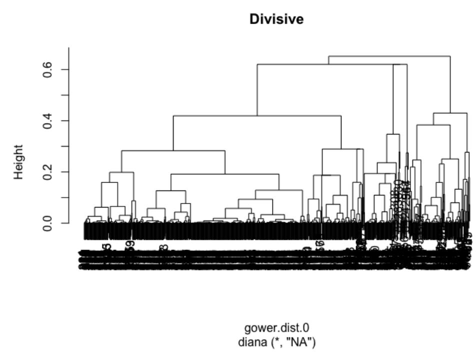


Figure 9: Dendrogram for Divisive Hierarchical Clustering

types of linkages. We also looked at the agglomerative coefficient, which measures the amount of clustering structure found (values closer to 1 suggest strong clustering structure) [7]. From Table 7, we can see ward linkage has the highest agglomerative coefficient. Figure 11 gives the 2D representation of the clusters with feature subset 0 and 6 clusters after agglomerative clustering with ward linkage.

### 5.3 Conclusion and Next Steps

In conclusion we can see that feature engineering combined with clustering can be used to find segments of restaurants. Although all clusters are not distinct and there is overlap, certain clusters were distinctly identified like cluster 4, 5, 6 (Figure 11). As next steps we can try selecting better features with an exhaustive search approach. Also, to find better clusters we can try spectral clustering methods like DBSCAN or HDBSCAN. We can also create a dashboard like Figure 5 for the restaurant owners.

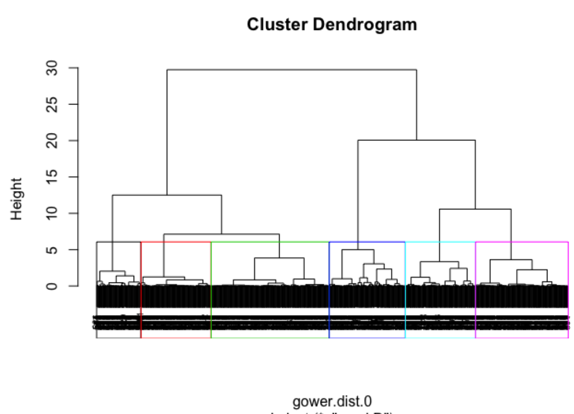


Figure 10: Dendrogram for Agglomerative clustering with ward linkage

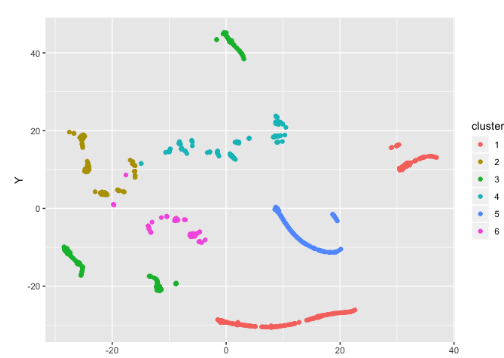


Figure 11: Visualization of the clusters with feature subset 0 and 6 clusters after Agglomerative clustering with ward linkage

Table 8: Distribution of restaurants across different states

| State | No of Restaurants | Percentage of Restaurants |
|---|---|---|
| Ontario | 13,501 | 25% |
| Arizona | 10,598 | 19% |
| Nevada | 7,135 | 13% |
| Quebec | 4,981 | 9% |
| Ohio | 4,814 | 9% |
| North Carolina | 3,850 | 7% |
| Pennsylvania | 3,647 | 7% |
| Others | 6,092 | 11% |

# 6 Problem Statement 3

What are some popular and recommended items/dishes in the restaurant that customers like? e.g - Popular items for Taste of India are - Butter Chicken and Palak Paneer.

To answer this question there are two major components. First we had to extract the positive comments that customers have left for each restaurant. Then we had to extract the references to food items in these comments.

## 6.1 Sentiment analysis

### 6.1.1 Pre-processing and Feature Engineering

To build the sentiment analysis model we needed text with some sort of sentiment flag. The tips data did not have such a flag. We decided to work with the reviews data to build the model, and then pass the tips through this model to filter the positive tips. We had different businesses and some basic attributes for every business in the Yelp dataset. The information available pertaining to a business were as follows:

- Business_id
- Name
- Neighborhood
- Address
- City
- State
- Postal_Code
- Latitude
- Longitude
- Stars
- Review_count
- Is_open
- Categories

There are a total of 174,567 businesses from which we needed to filter only the restaurants. We filtered the 'Restaurants' category and we were left with 54,618 restaurants. We looked at the distribution of restaurants across different states in the Table 8.

We then move on to investigate the yelp_reviews.csv dataset Columns in the dataset are –

- Review_id
- User_id
- Business_id
- Stars
- Date
- Text
- Useful
- Funny
- Cool

The dataset contains 5,261,668 reviews across all businesses. We first filtered the reviews for businesses which are only restaurants. We were left with 3,221,418 reviews. Since this was a huge number, to start with our initial Natural Language Processing (NLP) exercise we randomly sampled 100,000 reviews.

We extracted the year from the date and filtered out reviews older than 2010. Next we needed to create a 1-0 polarity for each review, which would serve as the dependent variable for the sentiment analysis model. We looked at the distribution of reviews belonging to different star categories (Table 9) and decided to encode a review with a star rating of 4 or greater as having positive sentiment and reviews having star rating less than 4 as negative. This encoding gave us a dependent variable with event rate of 65%

### 6.1.2 Results

Now that we had the reviews for restaurants and associated sentiment flag we started by building a baseline sentiment analysis model. We split our reviews to train (70%), validation (15%) and test (15%) set.

Table 9: Distribution of reviews across different star ratings

| Stars | No of Reviews |
|---|---|
| 1 | 11,395 |
| 2 | 9,447 |
| 3 | 14,090 |
| 4 | 27,817 |
| 5 | 37,251 |

We tried unigram as well as bigram, Bag of Words features and TF-IDF features. We built a baseline Logistic Regression model and SVM model. We also regularized the parameters using L2 penalty since the data was very sparse with close to 62,000 features for unigram and 1,166,319 features for unigram + bigram. The regularization parameters were chosen by looking at the results on the validation set. F1 score on the test set for the different iterations are as follows:

| Features/Models | Logistic Regression | | SVM | |
|---|---|---|---|---|
| | Regularization Parameter | F1 Score | Regularization Parameter | F1 Score |
| Bag of Words : Boolean (uni) | 0.05 | 0.9166 | 0.01 | 0.9175 |
| Bag of Words : Boolean (uni+bi) | 0.25 | 0.9265 | 0.01 | 0.9265 |
| Bag of Words : Count (uni) | 0.05 | 0.9176 | 0.01 | 0.9174 |
| Bag of Words : Count (uni+bi) | 0.25 | 0.9254 | 0.01 | 0.9251 |
| Tf-idf (uni) | 1 | 0.9192 | 0.25 | 0.9201 |
| Tf-idf (uni+bi) | 1 | 0.9255 | 0.25 | **0.9292** |

SVM with unigram and bigram tfidf features gaves the highest F1 score on the validation set so we finalized this model.

### 6.2 Food entity recognition

#### 6.2.1 Pre-processing and Feature Engineering

The idea was to look at the positive comments about a restaurant present in the tips left by customers using sentiment analysis and then identify the popular food items for a given restaurant. As mentioned in section 5.1.1 as Toronto has the highest number of restaurants in our dataset we're considering tips of restaurants in Toronto for our analysis. Being aware of the products or services that are popular among the customers is a great insight for businesses to decide where to invest money in. In our case a restaurant knowing what dishes are really popular and appreciated would be a great attribute to decide how to optimise deals and discounts.

We had a total of 1,098,324 rows in the 'yelp_tip.csv'. We merged our data from 'restaurants_in_Toronto.csv' that we generated while working on Research Question 2 with the tips dataset and got the tips for restaurants in Toronto which were 37,250. Now we had a dataset where each tip was a string from which we had to identify the food items.

#### 6.2.2 Topic Modeling

This is a technique used to obtain topic vectors for any given corpus of text. There are various methods to do this, but the most popular of all is Latent Dirichlet Allocation (LDA), which is the one we have used. We only considered the first 5000 tips since doing LDA on all the tips was taking a lot of time. Then we performed some pre-processing where we removed all the unnecessary characters, numbers and symbols since they don't contribute towards our goal. We then used NLTK's POS tagger to remove the words which weren't "nouns" since we are only interested in nouns (food items). Next we removed any stop-words, in case the previous step missed anything. Then we converted our entire text to lowercase, since the case of the characters is irrelevant to the problem at hand. The next step performed was lemmatization to bring all the words to their base form to further reduce the word count which would help our algorithm in generating better topic vectors. The result of our pre-processing is shown in Figure 12.


Figure 12: Result of pre-processing tips

We then created a "bag of words" model to feed it to our LDA. We tried many different hyperparameters for the number of topic vectors. Through experimentation, we found that keeping number of topic vectors to 7 was giving us the most relevant results. The 7 topic vectors found are shown in Figure 13.

```
[(0,
  '0.026*"ice" + 0.025*"cream" + 0.016*"tea" + 0.015*"line" + 0.014*"room" + 0.012*"table" + 0.012*"business" + 0.012*"place" + 0.011*"event" + 0.011*"dog"'),
 (1,
  '0.030*"beer" + 0.026*"pizza" + 0.024*"bar" + 0.016*"lot" + 0.015*"selection" + 0.015*"pittsburgh" + 0.014*"great" + 0.014*"nice" + 0.012*"park" + 0.012*"awesome"'),
 (2,
  '0.069*"time" + 0.046*"day" + 0.028*"burger" + 0.025*"staff" + 0.019*"good" + 0.018*"year" + 0.016*"card" + 0.014*"work" + 0.014*"thing" + 0.013*"special"'),
 (3,
  '0.090*"food" + 0.058*"service" + 0.050*"great" + 0.020*"game" + 0.014*"shop" + 0.013*"cash" + 0.012*"everything" + 0.010*"customer" + 0.010*"brunch" + 0.010*"bar"'),
 (4,
  '0.081*"place" + 0.036*"hour" + 0.031*"night" + 0.026*"lunch" + 0.021*"dinner" + 0.021*"wing" + 0.015*"friday" + 0.014*"home" + 0.014*"roll" + 0.012*"fry"'),
 (5,
  '0.040*"order" + 0.025*"menu" + 0.020*"price" + 0.018*"location" + 0.017*"today" + 0.017*"breakfast" + 0.015*"sandwich" + 0.014*"meal" + 0.013*"side" + 0.012*"day"'),
 (6,
  '0.028*"coffee" + 0.027*"way" + 0.027*"store" + 0.014*"item" + 0.014*"love" + 0.013*"flavor" + 0.011*"cake" + 0.011*"area" + 0.009*"ipa" + 0.009*"family"')]
```

Figure 13: Topic vectors obtained from LDA

As we can see, topic vector number 1 contains food related items such as "beer", "pizza", "bar", etc. So this topic seems to be of relevance to our problem. But the problem with this approach is that the topic vectors are not as easily interpretable as we would like for the problem at hand. Another point to note is that they also contain irrelevant information such as "selection", "pittsburgh", "card", etc. Thus, even though we were able to obtain some relevant topics, this is not much use to us for the problem at hand. This brings us to the next approach we tried - Named Entity Recognition.

#### 6.2.3 Named Entity Recognition

In Named Entity Recognition (NER), we try to assign an entity such as "LOCATION" to a name such as "Pacific Ocean", "TIME" to "minutes" and so on. We tried to use pretrained models on NER such as Flair [10], Spacy [11] to tag the food items mentioned in the tips in our dataset. Although the results were not very accurate since the entity "PRODUCT" represents items such as "vehicles", "weapons", etc. along with "food", which returns a lot of noise. This could be solved by training the NER ourselves on food items, but we did not have any labeled dataset for this task. So we could not continue with this approach and thus had to use the Naive approach, which surprisingly (or not surprisingly), gave decent results.

#### 6.2.4 Naive approach

In this Naive approach, we used a list of all the food items [12], which was made by the creator by scraping wikipedia for food items. Again, we performed some pre-processing on the tips in our dataset. First, we converted everything to lowercase. Then, we removed the punctuations. Next, we used NLTK's POS tagger to just extract the nouns from the tips since we are only interested in the food items. Now, from the remaining words we can check if the word is present in the extensive list of food items. But a problem may arise if the names of food items consist of 2 or 3 words, eg- "Garlic Bread". To tackle this problem, we have used monograms, bigrams and trigrams to search for food items in our list. Using this simple approach, we were able to successfully identify the food items in our tips to a great extent. The results can be seen in the following image:

| | | |
|---|---|---|
| 29 | Can't go wrong in this place. Love the Tawse Riesling. | [riesling, , ] |
| 30 | Thursday night bring some ketchup and mustard. It's a sausage night after work. :) | [sausage,ketchup, , ] |
| 31 | good food. Had a great tuna sushi starter. Very flavourful. | [sushi,tuna, , ] |
| 32 | I've always enjoyed the consistency of service and food offered by Earls it nary falters. Today was no different. | [, , ] |
| 33 | Everything was great. Had an awesome setup. Really enjoyed myself | [, , ] |
| 34 | Great spot to hangout. Staff not so friendly but decor is fabulous. | [, , ] |
| 35 | Thursday night's wine special ... $1 per ounce :) ... if you're into wine, this is awesome! Added atmosphere, appys, food & friends, you're good to go! | [wine, , ] |
| 36 | Beautiful people and a killer kale salad. | [salad,kale, , ] |
| 37 | Good food and awesome vibe. One of my go to spots in TO's financial district. | [, , ] |
| 38 | Perfect Tuscan Cuisine. Just like going. Take a vacation there. | [, , ] |
| 39 | Any Italian restaurant that serves bottarga gets my business. This one does it even better by having it on its regular pasta menu, and serving it with wonderfully textured fresh made pasta and pieces of anchovies. Anchovy heaven! | [pasta, , ] |
| 40 | The portions are smallish so dinner gets pricy: $80 EACH for a SHARED app, two pasta entrees, glass of wine and dessert. | [pasta,dessert,wine, , ] |
| 41 | Great patio! Servers are attentive and pleasant. Food is full of flavour! | [flavour, , ] |
| 42 | They renovated , the wines are cellar temp & the clam linguini is amazing! | [clam, , ] |
| 43 | Best SUMMER cocktail: The Affair-totally fresh! | [cocktail, , ] |
| 44 | The lamb ragu is phenomenal | [, , ] |
| 45 | The tortellini filled with mushrooms and the flour less chocolate cake with cayenne brittle are Absolutely Fabulous! | [flour,tortellini,cake, chocolate cake, ] |
| 46 | Tasty wines on the Enomatic + good food = yay! Cauliflower, potato side, carbonara, & amatriciana - all tasty orders. | [carbonara, , ] |
| 47 | pending 170125\nYelp best of... | [, , ] |
| 48 | Great date place ;-)) lighting is perfect and the food is yummy. Ripasso red wine from Veneto is like Nectar | [wine,veneto, red wine, ] |
| 49 | Try the Grilled Octopus and Mezzaluna Pasta. Both were to die for. | [pasta,octopus, , ] |

Figure 14: Naive Approach

### 6.3 Conclusion and Next Steps

From our experiments we can see that topic modeling is giving us indistinct overlapping topics which are hard to interpret. The naive approach is hugely dependent on the accuracy of the list of food items. If the list is inaccurate and has noisy words, we might miss out on identifying certain food items accurately. The pretrained NER models we tried are not specifically trained on identifying food items so the results were not accurate. If we have a labelled dataset of food items and text we can try building an NER model and then use the trained model to identify references to food items on tips left by the customers.

## References


[1] Yu, Boya, Jiaxu Zhou, Yi Zhang, and Yunong Cao.Identifying Restaurant Features via Sentiment Analysis on Yelp Reviews. In arXiv preprint arXiv:1709.08698, 2017

[2] https://medium.com/@Vishwacorp/nlp-analysis-of-yelp-restaurant-reviews-30b3d0e424a6

[3] https://www.r-bloggers.com/clustering-mixed-data-types-in-r/

[4] More information about the Yelp Dataset Challenge can be found at. https://www.yelp.com/dataset/challenge

[5] https://www.yelp.com/dataset/documentation/main

[6] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[7] https://towardsdatascience.com/hierarchical-clustering-on-categorical-data-in-r-a27e578f2995

[8] Hassan, Abdalraouf, and Ausif Mahmood. "Deep learning approach for sentiment analysis of short texts." In 2017 3rd international conference on control, automation and robotics (ICCAR), pp. 705-710. IEEE, 2017.

[9] https://medium.com/@pmin91/aspect-based-opinion-mining-nlp-with-python-a53eb4752800

[10] https://github.com/zalandoresearch/flair

[11] https://spacy.io/api/entityrecognizer

[12] https://github.com/CurtisGrayeBabin/List-of-all-Foods