Topic Modeling Yelp Reviews by Sentiment

Capstone Project 2
Miguel Montano

I. Background and Problem Statement

A customer's perspective of an individual business' strengths and weaknesses are of extreme importance in the service sector, and with the availability of a ranking system built by consumers themselves (Yelp star ratings), further insights can be drawn by analyzing the reviews that harbor polarizing opinions . Of the plethora of discoveries that can be inferred from such evidence, the ability to translate commentary into actionable results is of the utmost importance, particularly in identifying positive attributes to reinforce and negative attributes to mitigate. I propose an approach that utilizes the existing rating system in Yelp reviews to infer sentiment, attributing low and high ratings to negative and positive sentiment respectively, and creating a Latent Dirichlet Allocation topic model for each set, extracting topics that are being discussed when users write reviews with strong opinions. Latent Dirichlet Allocation (LDA) is a 'generative probabilistic model', seeking to allow sets of observations to be described by unobserved groups that explain why some parts of the data are similar. For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's presence is attributable to one of the document's topics. In this case the documents are the reviews and the parts are the words and/or phrases (n-grams), with LDA serving as way of soft clustering the documents and parts. The fuzzy memberships drawn provide a more nuanced way of inferring topics, as each review can be made up of a combination of them, versus simply attributing each review to one topic. This feature of LDA is key in this instance, as we are seeking to draw out terms indicative of constructive comments, with the assumption that customers might write about more than one topic in their reviews. A scikit-learn pipeline will be constructed to streamline the review extraction, text preprocessing, and LDA steps, creating a topic model customizable to the client's interests.

II. Potential Clients

The immediate beneficiaries of this model would be the businesses being reviewed themselves, seeking to better understand their customer's perspective without having to personally comb through all of the data Yelp makes available. Other potential uses include analysis based on any combination of location attributes, categorical tags, or particular ratings, as the pipeline approach allows for versatile experimentation. This use case would lend itself to any clients with an interest in discovering sentiment specific topics in a particular sector, such as the food industry, or special interest groups dedicated to industries in a local area.

III. Dataset

Data has been acquired from an online posting on Kaggle containing a subset of Yelp's business, review, and user data. It was originally assembled for the Yelp Dataset Challenge, and I will primarily be utilizing the reviews themselves grouped by categorical tags and star rating. Additional sets were used for reference, one with North American country province/state names and abbreviations, and another provided by Yelp housing the taxonomy of the categorical tags, utilized as a resource in the grouping process.

IV. Exploratory Data Analysis

Original review data was in a JSON format and imported into a pandas JsonReader object, as there are around six million entries. The JsonReader object houses the total dataset split into DataFrames with at most one hundred thousand rows each, which were then unpacked into a list to assist in the readability of future calculations. Business profile data, US & Canada state/province codes, and the taxonomy of Yelp categories were also JSON files opened into pandas DataFrame objects.

1. Business Profiles

```
BUSINESSES OVERVIEW

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 188593 entries, 0 to 188592
Data columns (total 16 columns):
address          188593 non-null object
attributes       162807 non-null object
business_id      188593 non-null object
categories       188052 non-null object
city             188593 non-null object
hours            143791 non-null object
is_open          188593 non-null int64
latitude         188587 non-null float64
longitude        188587 non-null float64
name             188593 non-null object
neighborhood     188593 non-null object
postal_code      188593 non-null object
review_count     188593 non-null int64
stars            188593 non-null float64
state            188593 non-null object
state_name       188003 non-null object
dtypes: float64(3), int64(2), object(11)
memory usage: 23.0+ MB
```

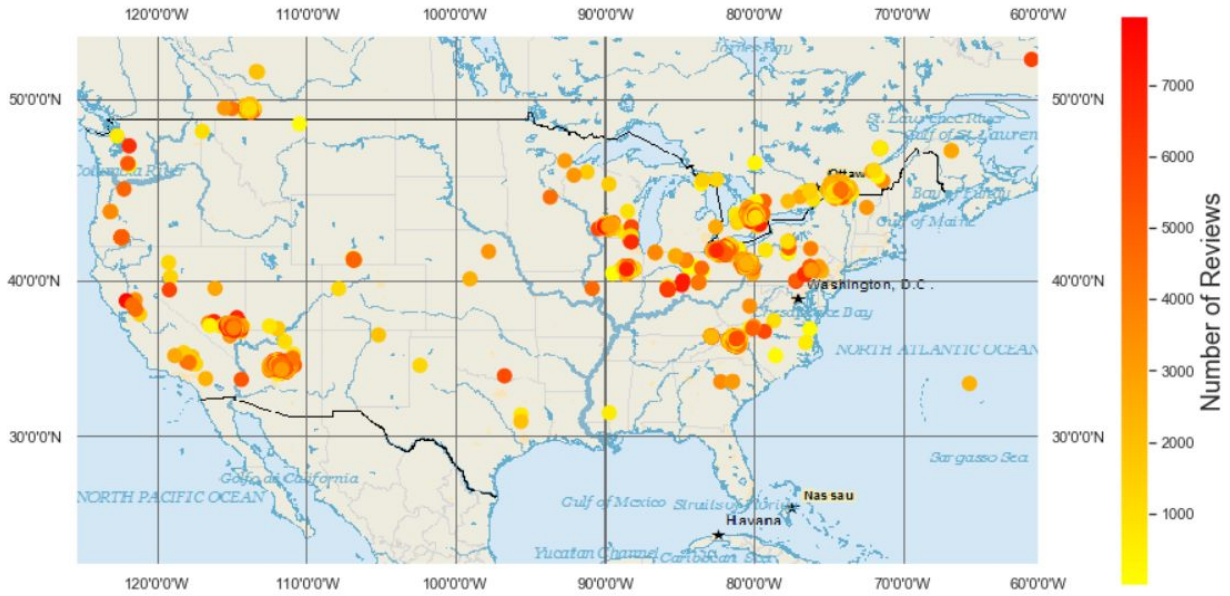
Descriptive Statistics:

	is_open	latitude	longitude
count	188593.000000	188587.000000	188587.000000
mean	0.830391	38.506793	-97.490873
std	0.375290	5.122684	17.693360
min	0.000000	-71.753941	-180.000000
25%	1.000000	33.630878	-112.279276
50%	1.000000	36.143595	-111.777460
75%	1.000000	43.593106	-79.982958
max	1.000000	85.051129	115.086769

	stars	review_count
count	188593.000000	188593.000000
mean	3.631550	31.797310
std	1.016783	104.124212
min	1.000000	3.000000
25%	3.000000	4.000000
50%	3.500000	9.000000
75%	4.500000	24.000000
max	5.000000	7968.000000

There are a total of 5,996,996 reviews in the Yelp Review dataset, with 99.996% of them correlating to a profile accounted for in the Yelp Businesses dataset. Profile information ranges from the name, location, and number of reviews, to the businesses' average star rating and categorical tags attributed to them. Eighty-three percent of the businesses with profiles are open, and on average they have a little less than 32 reviews each, with the most reviewed business garnering 7968 reviews. The vast majority of all businesses (99.14%) are in the same ten states/provinces, with 99.662% of all reviews coming from businesses in these locations.

2. Location of Businesses in the Top States

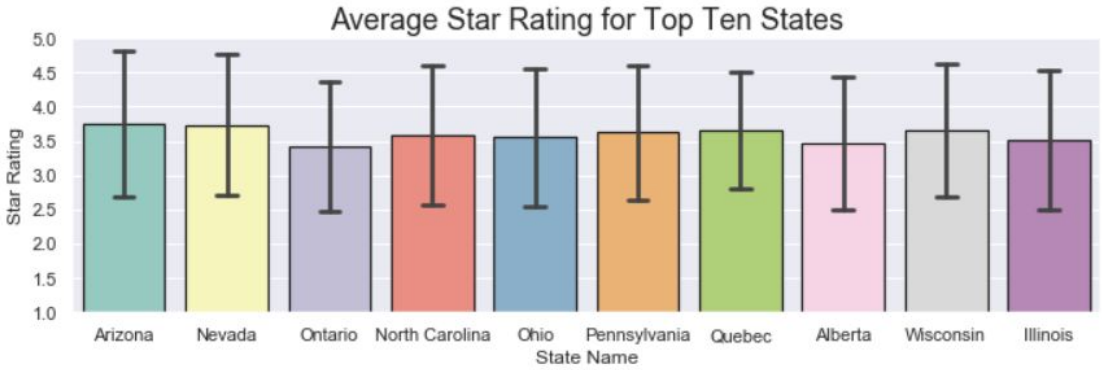


The top ten states are all in North America, with the majority of businesses being clustered around metropolitan areas, and the businesses with the largest number of reviews sharing a tendency of being located in large cities.

3. Top Ten States and Cities

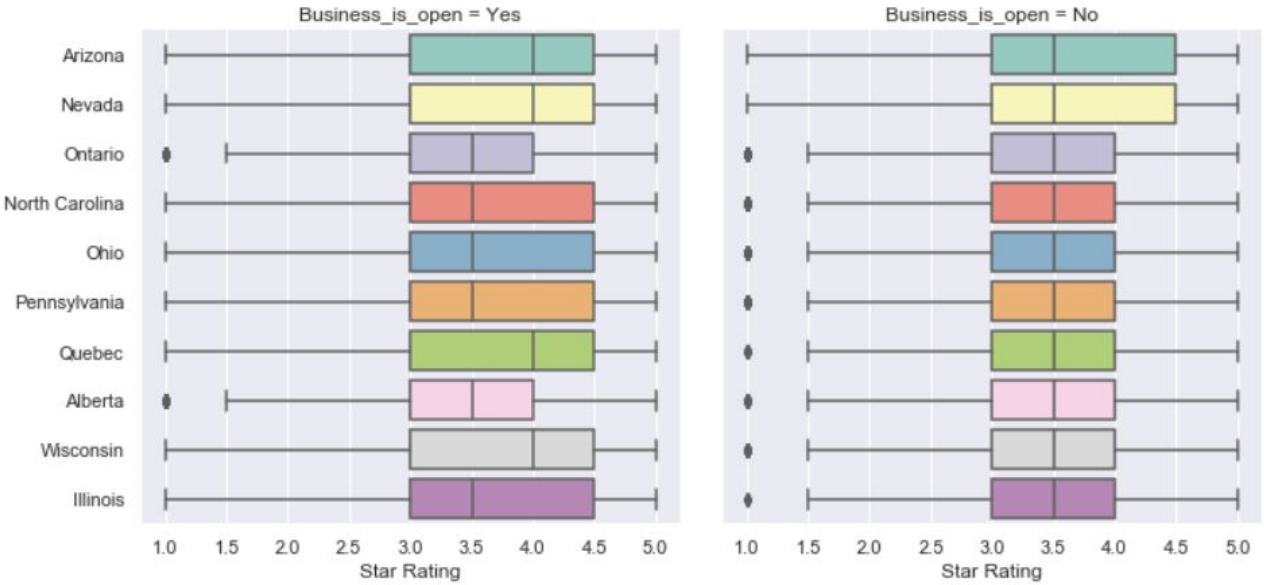
Of the top ten states, more than half of all businesses are located in the top three, with the top state having as many businesses as numbers five through ten combined. The top cities are similarly skewed, with Las Vegas alone containing more businesses than some of the other top states combined; the top cities are also all located in the same seven states.

4. Average Star Rating for the Top Ten States



Despite the notable variance in the number of businesses and total reviews in each state, the average star rating for all businesses in each state didn't vary much, hovering around 3.6 stars, with a standard deviation of around 1.5 stars.

5. Distribution of Star Ratings for Top Ten States



Unsurprisingly, the median rating and interquartile range for businesses that are still open are typically greater than in businesses that are closed, independent of what state they're in. The exception for having a smaller IQR being Arizona and Nevada, which had considerably larger sample sizes.

6. Average Number of Reviews Per Star Rating

A heatmap of the mean number of reviews per star rating for each of the top states shows that the more reviews a business receives, the more likely it is to achieve a moderate rating between 3-4.5 stars. Although low ratings appear more frequently when a business has a small number (<10) of reviews, the same can be said of a perfect five star rating, which seems more likely if a business has only been reviewed a few times.

7. Categories and Top Ten Food Tags

```
CATEGORIES OVERVIEW

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1539 entries, 0 to 1538
Data columns (total 5 columns):
alias                1539 non-null object
country_blacklist    349 non-null object
country_whitelist    518 non-null object
parents              1539 non-null object
title                1539 non-null object
dtypes: object(5)
memory usage: 60.2+ KB
```

```
Top 10 Food Tags:
1. Restaurants
2. Food
3. Nightlife
4. Bars
5. Coffee & Tea
6. Sandwiches
7. Fast Food
8. American (Traditional)
9. Pizza
10. Burgers
```

Yelp makes available to reviewers 1539 unique tags, which fall under the umbrella of 118 unique parent tags, encompassing virtually all types of businesses in the service sector. For the sake of narrowing the amount of reviews being utilized, parent tags pertaining to food related businesses were used to create a list of all tags that belong to the food industry; with the top ten being quite unspecific, number one being 'Restaurants' and number two simply 'Food'.

8. Food Industry Business Profiles

```
FOOD INDUSTRY OVERVIEW

<class 'pandas.core.frame.DataFrame'>
Int64Index: 73536 entries, 0 to 188590
Data columns (total 17 columns):
address         73536 non-null object
attributes      71567 non-null object
business_id     73536 non-null object
categories      73536 non-null object
city            73536 non-null object
hours           55299 non-null object
is_open         73536 non-null int64
name            73536 non-null object
neighborhood    73536 non-null object
postal_code     73536 non-null object
review_count    73536 non-null int64
stars           73536 non-null float64
state           73536 non-null object
state_name      73144 non-null object
latitude        73535 non-null float64
longitude       73536 non-null float64
color           73536 non-null object
dtypes: float64(3), int64(2), object(12)
memory usage: 10.1+ MB
```

```
Descriptive Statistics:
            is_open  review_count         stars
count  73536.000000  73536.000000  73536.000000
mean       0.738047     55.236796      3.494764
std        0.439700    145.715852      0.823946
min        0.000000      3.000000      1.000000
25%        0.000000      6.000000      3.000000
50%        1.000000     16.000000      3.500000
75%        1.000000     49.000000      4.000000
max        1.000000   7968.000000      5.000000

            latitude     longitude
count   73535.000000  73536.000000
mean       40.020325    -92.323266
std         5.360234     18.120518
min       -71.753941   -123.587426
25%        35.233611   -112.073403
50%        41.152726    -81.439480
75%        43.695149    -79.431609
max        59.438181    115.086769
```

Of the 188,593 business profiles, 73,536 belong to the food industry. Food businesses tend to have a higher review count on average than the population mean, with 55 reviews per business vs 32 for all businesses. Average star rating for food businesses didn't differ as much, still hovering around 3.5 stars, but with a smaller standard deviation of 0.8 stars. The food industry appears to also have a higher closing rate than the population, with only 73.8% of them still open, ten percent less than average.

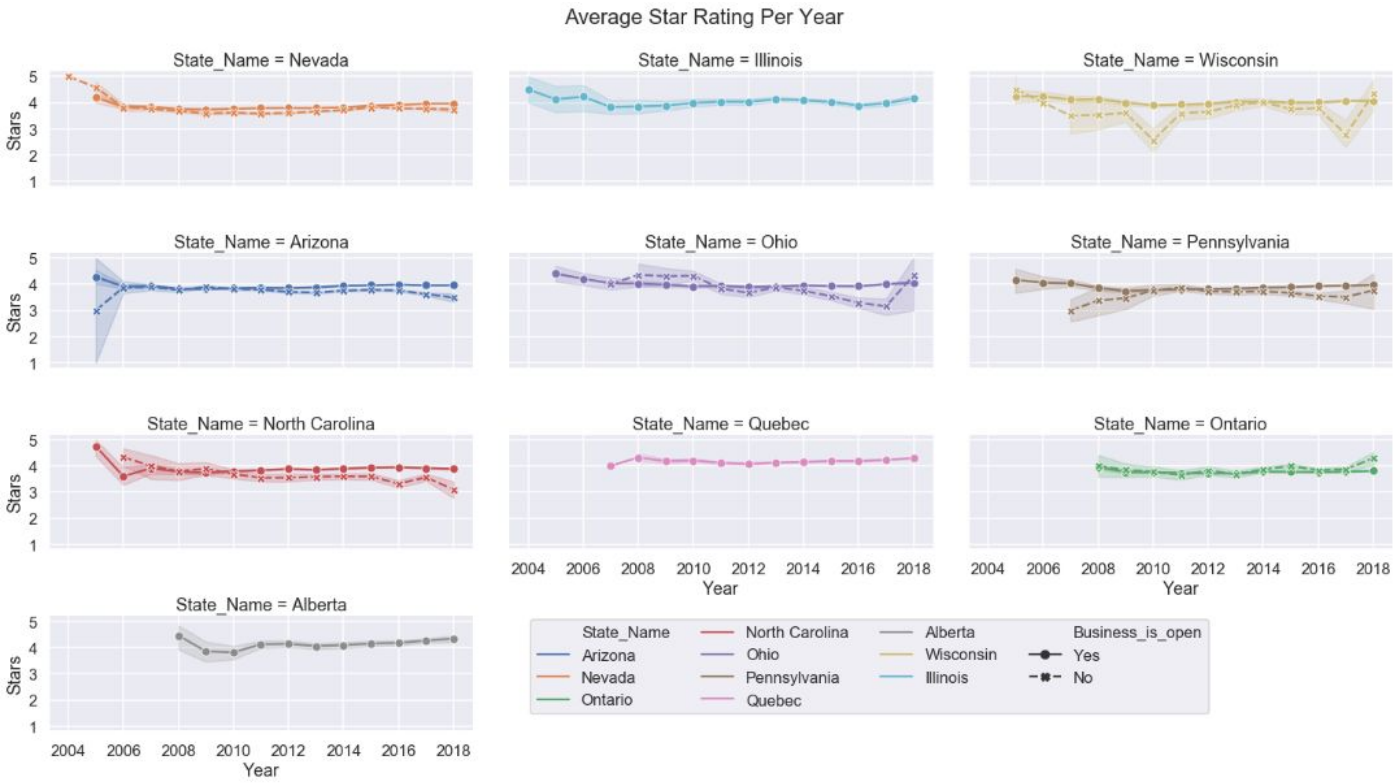
9. Food Industry Reviews

```
FOOD REVIEWS OVERVIEW

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1905325 entries, 0 to 1905324
Data columns (total 9 columns):
business_id     object
cool            int64
date            datetime64[ns]
funny           int64
review_id       object
stars           int64
text            object
useful          int64
user_id         object
dtypes: datetime64[ns](1), int64(4), object(4)
memory usage: 130.8+ MB
```

	cool					funny					useful				
	mean	std	min	max	count	mean	std	min	max	count	mean	std	min	max	count
stars															
1	0.307	2.144	0	505	162220	0.727	3.710	0	637	162220	1.610	8.598	0	1118	162220
2	0.428	1.872	0	172	157849	0.636	2.178	0	274	157849	1.398	5.250	0	1234	157849
3	0.634	2.603	0	245	241866	0.579	2.849	0	435	241866	1.201	3.741	0	805	241866
4	0.838	3.130	0	229	512665	0.556	3.094	0	566	512665	1.202	3.509	0	245	512665
5	0.608	2.322	-1	208	830725	0.371	3.240	0	991	830725	0.910	2.707	-1	215	830725

There are 1,905,325 Yelp reviews concerning Food Industry related businesses, nearly a third of all Yelp reviews available in the academic dataset. Another discernible trait is that the vast majority of reviews do not have an additional 'cool', 'funny', or 'useful' tag, but if a review does have such a tag, it is more than likely one giving a favorable rating of 4-5 stars.

10. Average Food Industry Star Rating Per Year For the Top States

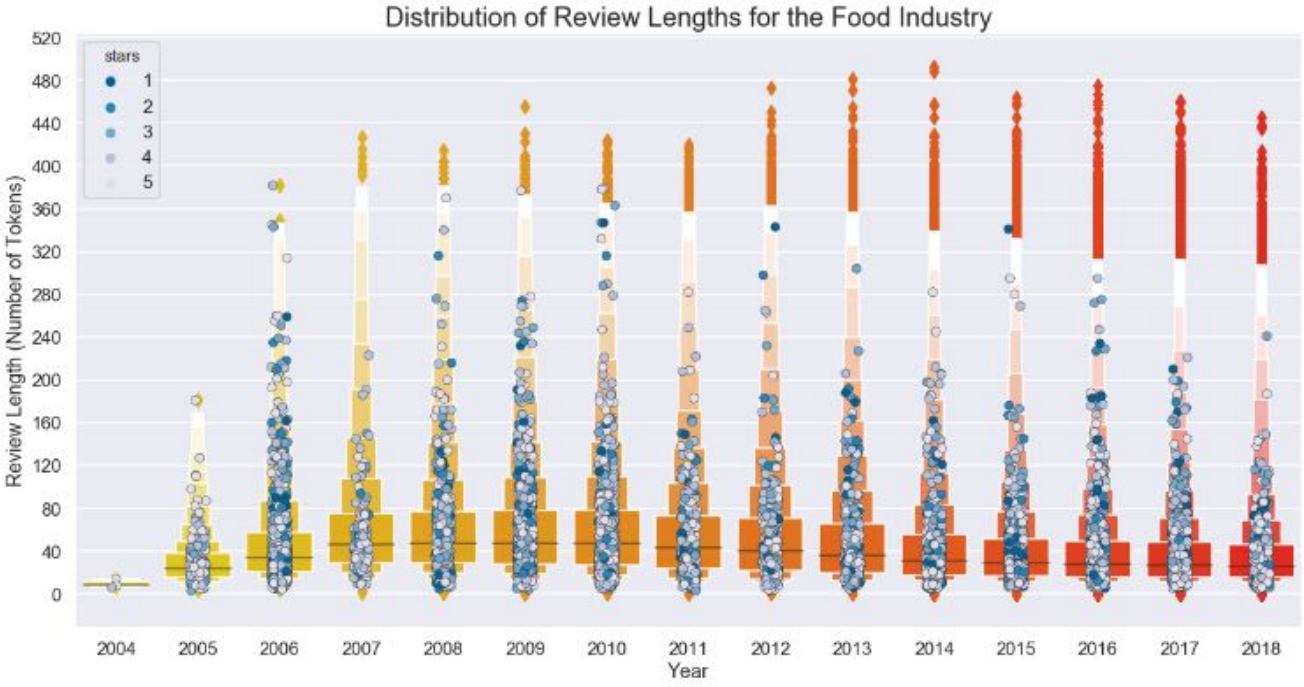


An overview of the change in the average star rating for each of the top ten states throughout the sample period shows that for the most part, reviewers have maintained the same levels of star ratings, around 3.5-4 stars per year. Notable exceptions may be attributed the deficiencies in the sample proportions available i.e. some states have few to no ratings in certain years, as well as the variance in the amount of reviews available per state. Some differences do appear to exist between businesses that are open and those that aren't, as open businesses appear to maintain higher average ratings, but further analysis is necessary to surmise the significance of such a finding.

11. Average Length of Food Industry Reviews Per Star Rating

Although the average length of most reviews lies between a small range of 35 -55 terms, excluding stopwords and special characters, there does appear to be a stark difference in the length of negative versus positive reviews, independent of what state the business being reviewed resides in. This might be attributed to the fact that a customer with a negative experience might feel more inclined to disclose their experience in greater detail as opposed to someone who had a positive experience. This is interesting as most reviews are positive (3-5 stars), but the longest reviews are overwhelmingly negative.

12. Distribution of Review Lengths By Year for the Food Industry



A closer look at the distribution of review lengths by year showed that after Yelp's founding in 2004, the quantity and length of reviews made increased tremendously over the following two years.

Although eventually the length of reviews appears to have leveled out, with the 75th percentile being less than eighty terms for all the years in the sample, and outlier reviews hovering at around 320-500 words.

V. Data Wrangling

Although the transformative steps will be included in future pipeline construction, for the purpose of training a baseline model, the food industry reviews selected earlier during exploratory data analysis were treated as the initial subjects. The total reviews were split into groups based on the star rating they attribute, with those giving less than three stars being deemed negative and those giving more than three stars considered positive . In this case data wrangling encompasses taking reviews from raw text strings and manipulating them into vectors that can be run through the LDA algorithm. The two major steps undertaken were text pre-processing and feature extraction.

1. Text Preprocessing

This portion of the procedure encompasses the majority of the work involved, with the four major steps of preprocessing being tokenization, normalization, stopwords removal, and lemmatization. Tokenization involves removing punctuation, whitespace, and special characters from the entire review, and transforming the passages from one long string into a list of word strings. Normalization consists of converting the remaining word strings into lowercase, expanding contractions by removing hyphens and apostrophes, and doing away with numerals and accent marks, such that only individual words remain. Stopwords, or terms which occur frequently and contribute little to overall meaning ('which', 'the', etc..), as well as words less than three characters are removed, so as to not clutter the final result. Finally, Lemmatization is undertaken, which is the process of eliminating affixes from a word by capturing the canonical forms based on a word's lemma, or chosen representative, in this case assigning a Verb category tag to the tokens i.e. 'running', 'runs', 'ran' all become 'run'; this simplifies the total vocabulary being analyzed without diminishing semantic value.

```
Negative Food Industry Reviews

<class 'pandas.core.frame.DataFrame'>
Int64Index: 319968 entries, 13 to 1904179
Data columns (total 14 columns):
business_id         319968 non-null object
cool                319968 non-null int64
date                319968 non-null datetime64[ns]
funny               319968 non-null int64
review_id           319968 non-null object
stars               319968 non-null int64
text                319968 non-null object
useful              319968 non-null int64
user_id             319968 non-null object
year                319968 non-null int64
State_Name          319968 non-null object
Business_is_open    319968 non-null object
tokens              319968 non-null object
review_length       319968 non-null int64
dtypes: datetime64[ns](1), int64(6), object(7)
memory usage: 36.6+ MB
```

```
Positive Food Industry Reviews

<class 'pandas.core.frame.DataFrame'>
Int64Index: 1342441 entries, 0 to 1904177
Data columns (total 14 columns):
business_id         1342441 non-null object
cool                1342441 non-null int64
date                1342441 non-null datetime64[ns]
funny               1342441 non-null int64
review_id           1342441 non-null object
stars               1342441 non-null int64
text                1342441 non-null object
useful              1342441 non-null int64
user_id             1342441 non-null object
year                1342441 non-null int64
State_Name          1342441 non-null object
Business_is_open    1342441 non-null object
tokens              1342441 non-null object
review_length       1342441 non-null int64
dtypes: datetime64[ns](1), int64(6), object(7)
memory usage: 153.6+ MB
```

```
Negative Food Industry Reviews

1904080    Decor is nice but slow service, pastries were not good at all, tiramisu just ok but too thick. C...
1904091    Came in and order at 3.16 pm waited for 30 mins. Still haven't gotten our order. Hopefully next ...
1904147                  Good atmosphere and location, but the taste of the coffee and deserts are horrendous.
1904178    We traveled 30 minutes to this spot since we been here once before and wanted to come back to tr...
1904179    OVERHYPED and OVERRATED.\n\nYes it's aesthetically pleasing to the eyes. Nice greenhouse in the ...
Name: text, dtype: object

1904080    [decor, nice, slow, service, pastries, good, tiramisu, litchi, rise, pastrie, flavor, come, choo...
1904091             [come, order, wait, mins, haven, get, order, hopefully, time, come, bavk, better, service]
1904147                                    [good, atmosphere, location, taste, coffee, desert, horrendous]
1904178    [travel, minutes, spot, want, come, different, things, yelp, google, show, close, arrive, exactl...
1904179    [overhyped, overrate, aesthetically, please, eye, nice, greenhouse, middle, beautiful, victorian...
Name: tokens, dtype: object


Positive Food Industry Reviews

1904173    Friends night out and I chose Gabi Coffee and Bakery and everyone loved this place.  5 stars for...
1904174    This is a hidden gem in Las Vegas! The aesthetic and history surely creates an experience to the...
1904175    This is my favorite hang in Las Vegas!!! Because of the indoor atrium and korean decor it's like...
1904176    Little confusing to find cause of just having a big brown door and no sign. But when you walk in...
1904177    Their desserts are super cute and you can tell that the staff puts hard work into them, consider...
Name: text, dtype: object

1904173    [friends, night, choose, gabi, coffee, bakery, love, place, star, vibe, decor, ambiance, wish, h...
1904174    [hide, vegas, aesthetic, history, surely, create, experience, personal, favorite, latte, perfect...
1904175                          [favorite, hang, vegas, indoor, atrium, korean, decor, like, vegas, love]
1904176    [little, confuse, cause, have, brown, door, sign, walk, inside, like, different, little, world, ...
1904177    [desserts, super, cute, tell, staff, put, hard, work, consider, small, detail, cake, enjoy, expe...
Name: tokens, dtype: object
```

2. Feature Extraction

Transforming the preprocessed tokens garnered from each review into features is done with the help of Gensim's Dictionary module. Creating a Dictionary object results in a mapping between words and their given integer id, drawn from the preprocessed tokens in each group (negative and positive reviews). It does so by sweeping across the entire body of work (all the reviews in each group), assigning a unique integer id to each word, and then collecting word counts and other relevant statistics. For the baseline model a bag-of-words approach was used, such that the tokenized documents were converted into vectors by counting the number of occurrences of each distinct word, converting the word to its integer id, and returning the result as a sparse vector; thereby disregarding grammar and even word order but keeping multiplicity. Thus a sentence such as 'apples are great, I love apples' after preprocessing and vectorization would become (0,2), (1,1),(2,1). This is done with the doc2bow method of the Dictionary object created earlier, and would be different for each set of reviews, resulting in two final corpora (one for negative and one for positive reviews) .

VI. Baseline Model

A baseline was built for each corpus, the negative and positive food industry reviews. It was done so using Gensim's LDA Model, which is streamed and runs in constant memory with respect to the number of documents; meaning training documents may come in sequentially, requiring no random access, and the size of the training corpus does not affect memory footprint. In this case they are considered unigram models as a bag-of-words approach was taken, where each vector houses words individually, ignoring context and phrasing. Ten topics were selected to be drawn from the LDA process, with all other parameters left at their default settings. Both models showed promising initial results. With the ten topics derived from each seeming to harbor associations feasibly inline with human logic, i.e. 'breakfast', 'coffee', 'brunch' all appear together, and 'beer', 'wine', 'glass' also appearing together. An apparent downside to selecting such a broad range of reviews was the lack of specificity when it came to the topics themselves, as many were simply food categories,

breakfast/pizzeria/alcohol items were all grouped together for both the negative and positive topic models, instead of drawing different sentiment specific topics.
Below are the top five words for each topic, and their respective weights:

```
NEGATIVE LDA TOPICS
Topic: 0
Words: 0.035*"breakfast" + 0.030*"coffee" + 0.024*"egg" + 0.014*"brunch" + 0.014*"toast"
Topic: 1
Words: 0.027*"place" + 0.025*"like" + 0.015*"look" + 0.012*"drink" + 0.009*"people"
Topic: 2
Words: 0.076*"beer" + 0.049*"wine" + 0.037*"glass" + 0.025*"bottle" + 0.021*"beers"
Topic: 3
Words: 0.019*"order" + 0.016*"fry" + 0.016*"good" + 0.014*"like" + 0.013*"taste"
Topic: 4
Words: 0.019*"room" + 0.012*"tell" + 0.011*"stay" + 0.011*"say" + 0.011*"go"
Topic: 5
Words: 0.030*"order" + 0.026*"pizza" + 0.019*"like" + 0.018*"rice" + 0.018*"chicken"
Topic: 6
Words: 0.039*"food" + 0.025*"place" + 0.024*"price" + 0.019*"restaurant" + 0.014*"good"
Topic: 7
Words: 0.062*"food" + 0.048*"service" + 0.035*"place" + 0.029*"time" + 0.027*"good"
Topic: 8
Words: 0.037*"buffet" + 0.020*"line" + 0.017*"crab" + 0.014*"cake" + 0.013*"vegas"
Topic: 9
Words: 0.030*"order" + 0.025*"come" + 0.024*"wait" + 0.022*"table" + 0.019*"food"

POSITIVE LDA TOPICS
Topic: 0
Words: 0.025*"breakfast" + 0.019*"coffee" + 0.016*"cream" + 0.015*"buffet" + 0.013*"brunch"
Topic: 1
Words: 0.020*"place" + 0.016*"like" + 0.014*"drink" + 0.012*"good" + 0.008*"pretty"
Topic: 2
Words: 0.032*"fry" + 0.027*"good" + 0.025*"chicken" + 0.022*"burger" + 0.019*"order"
Topic: 3
Words: 0.026*"order" + 0.020*"food" + 0.020*"time" + 0.020*"come" + 0.015*"wait"
Topic: 4
Words: 0.045*"place" + 0.029*"food" + 0.028*"love" + 0.027*"best" + 0.020*"time"
Topic: 5
Words: 0.046*"tacos" + 0.027*"chip" + 0.027*"mexican" + 0.021*"salsa" + 0.020*"taco"
Topic: 6
Words: 0.070*"great" + 0.061*"food" + 0.042*"place" + 0.042*"service" + 0.040*"good"
Topic: 7
Words: 0.092*"pizza" + 0.024*"italian" + 0.021*"pasta" + 0.019*"crust" + 0.018*"sauce"
Topic: 8
Words: 0.018*"order" + 0.018*"good" + 0.018*"roll" + 0.016*"rice" + 0.016*"dish"
Topic: 9
Words: 0.013*"steak" + 0.012*"dish" + 0.011*"dinner" + 0.011*"dessert" + 0.009*"restaurant"
```

VII. LDA Pipeline

A scikit-learn pipeline will be constructed for the purpose of streamlining the various procedural steps necessary for generating a desired topic model. Pipelines operate by sequentially applying a list of transformers, leading to the implementation of an estimator. Each of the steps undertaken prior to training the LDA model, from selecting the desired reviews and splitting them by sentiment, preprocessing the texts, and finally extracting features from them, will occur in custom transformers; the LDA model itself will be resting in the final custom estimator object. Each of the custom objects will be created by wrapping Gensim modules and Pandas functions, and most importantly, inheriting Scikit-Learn base classes, which allow the objects to be utilized in a pipeline by giving them fit/transform and get/set params methods.

Transformers:

1. Yelp Review Selector

Selects Yelp reviews based on specified criteria, with the option of returning all review data in a pandas DataFrame, or simply the review text filtered by sentiment as a pandas Series.

- **Input** (not case-sensitive, in order of superseding importance):
  - Unique business_id
  - Name of a business
  - List of categorical tags
  - All reviews by using 'select_all_reviews' as input str
- **Parameters:**
  - sentiment (str): 'positive', 'negative', or None (not case-sensitive)
  - star_threshold (int): rating used to infer sentiment (includes number of stars specified)
  - atleast_nreviews (int): only comments from businesses with at least the specified number of reviews will be included
  - given_id (bool): True if a specific business_id is given, False otherwise
  - given_name (bool): True if the name of a business is given, False otherwise
  - given_tags (bool): True if a list of categorical tags is given, False otherwise
- **Methods:**
  - fit (X, y=None): returns self
  - select_reviews (X): returns pandas DataFrame of selected reviews, unfiltered
  - transform (X): calls select_reviews method, filters by sentiment, and returns only review text as pandas Series

2. Preprocess

Preprocess text data, in this case the Yelp reviews themselves, per specifications; with the option to expand final vector space with n-grams.

- **Input:**
  - Container of text documents, wherein each is assumed to be in the form of a single string
- **Parameters:**
  - lemmatize (bool): True to lemmatize tokens using NLTK's WordNetLemmatizer, False otherwise
  - pos_tag (str): Part-of-Speech tag to use in lemmatization, 'v' for Verb, 'n' for Noun, and 'a' for Adjective
  - bigrams (bool): True to create new bigram tokens from multi-word expressions in corpus with score above threshold parameter, False otherwise
  - trigrams (bool): True to create new trigram tokens from multi-word expressions in corpus with score above threshold parameter, False otherwise
  - min_count (int): minimum number of occurrences for token to be considered in n-gram formation
  - threshold (float): score threshold phrase must meet for forming n-grams, higher meaning fewer n-grams are formed
- **Methods:**
  - fit (X, y=None): returns self
  - transform (X): returns list of lists, wherein each is a preprocessed document that has been tokenized, normalized, lemmatized, with stopwords removed

3. Vectorize

Create a vector space of features, where each feature is a sparse vector extracted from each document, in this case preprocessed Yelp reviews, consisting of terms and term weights

- **Input:**
  - Container of tokenized text documents
- **Parameters:**
  - filter_vocab (bool): True to filter the vocabulary of the corpus, False to use all terms
  - filter_no_below (int): if filter_vocab, keep tokens which are contained in at least this many documents
  - filter_n_freq (int): if filter_vocab, filter out this number of the most frequent tokens that appear in the documents
  - tfidf (bool): True to utilize Term Frequency-Inverse Document Frequency term weights, False to utilize term frequency weighting
- **Methods:**
  - dictionary (X): returns an instance of Gensim's Dictionary module called on X
  - corpus (X): calls dictionary method, and returns a bag-of-words(or n-grams) vector representation of X, with the option to filter the vocabulary inferred from the corpus
  - fit (X, y=None): returns self
  - transform (X): calls dictionary & corpus methods, and returns a tuple containing the results of each, with the option to use TF-IDF term weighting

Estimator:

LDA Estimator

Trains an instance of Gensim's Latent Dirichlet Allocation Model, with the option to create a pyLDAvis figure from the result.

- **Input:**
  - Container with a corpus (in the form of a vector space), and a trained Gensim Dictionary instance
- **Parameters:**
  - n_topics (int): number of latent topics to be extracted from the training corpus
  - passes (int): number of passes through the corpus during training
  - alpha (np.ndarray, 'auto'): 'auto' learns an asymmetric prior from the corpus, or a 1D array of length 'n_topics' that expresses a-priori belief for each topics' probability
  - eta ('auto', float, np.ndarray): A-priori belief on word probability, 'auto' to learn the asymmetric prior from the corpus; float for a symmetric prior over topic/word probability; 1D array of length = number of words in the corpus, to denote an asymmetric probability for each word; matrix of shape 'n_topics' and number of words in corpus , to assign a probability for each word-topic distribution
  - visualize (bool): True to create a pyLDAvis figure from the trained LDA model, False otherwise
- **Methods:**
  - fit (X, y=None): returns self
  - transform (X): returns trained instance of Gensim's LDA model; if visualize, then list with trained model and pyLDAvis figure of it

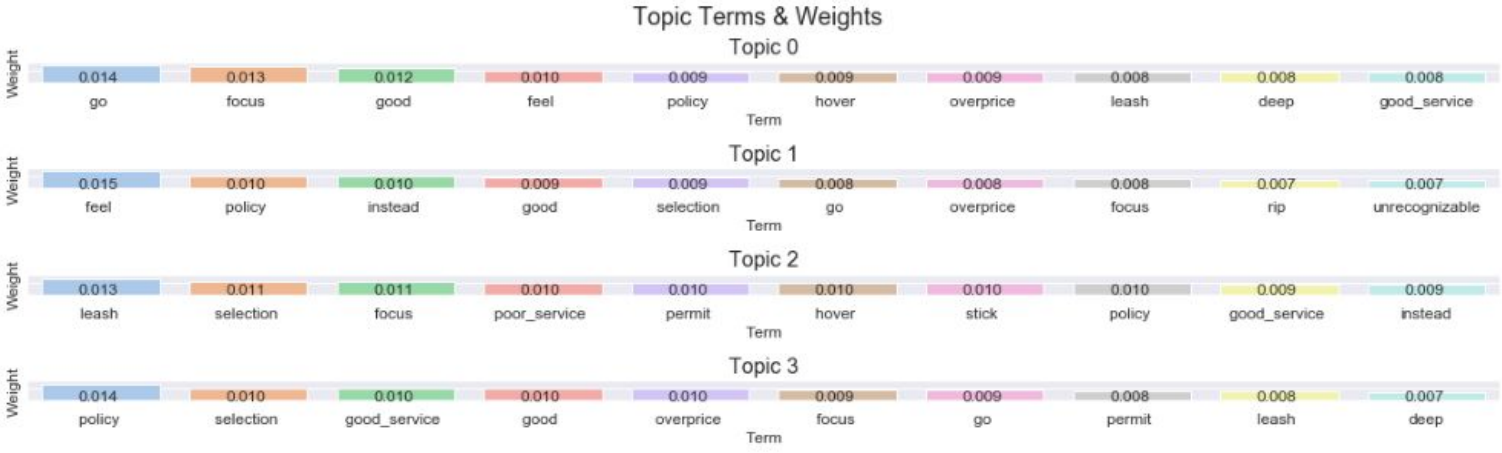
LDA Pipeline Implementation - Individual Business Example

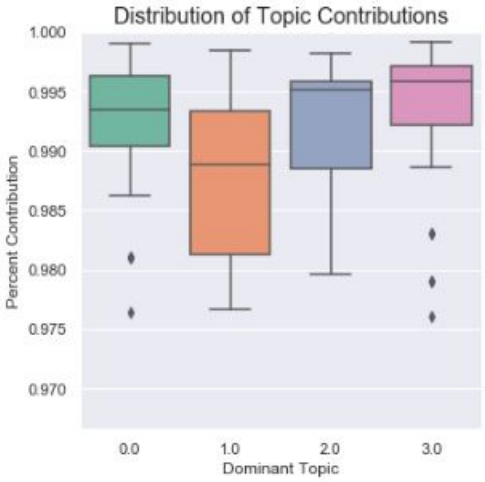
The primary use case apparent for the LDA pipeline constructed will be to better understand a consumer's perspective on an individual business without having to personally comb through all data Yelp makes available. For the purpose of ensuring a large enough sample size is available, a random business with at least 500 reviews was selected, in this case 'Postino Arcadia' restaurant from Phoenix, Arizona. Two separate pipelines were constructed, one for negative reviews and one for positive reviews, both of which sought to create vectors with term frequency weighting, possessing trigrams and removing the two most frequent tokens, as well as any tokens occurring in less than two reviews. The negative LDA model was trained on reviews having three or less stars, and the positive LDA model was trained on reviews giving ratings of four to five stars. The pipeline parameters were set such that the Gensim LDA model instances being trained ran ten passes over their respective corpora, and drew four distinct topics from the data. The metrics used to evaluate each result were a combination of statistical approximations for how well the probability models predicted topics based on their given samples in the case of Variational Bound and Log Perplexity, and a numeric representations of the perceived human interpretability of the garnered topics in the form of Coherence, drawn from a separate pipeline implementation through Gensim's Coherence Model. These metrics were coupled with graphical representations of per topic performance, the terms and weights for each topic, descriptive statistics per topic, and the most representative review for each one, calculated as the review in which the topic in question had the highest percentage contribution.

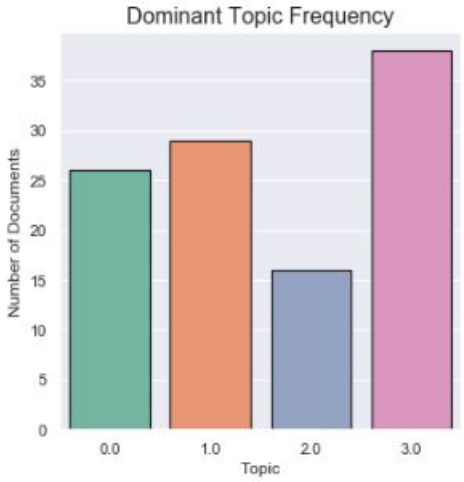
```
Individual Business - Negative Reviews (Trigrams)

Variational Bound: -29457
Log Perplexity: -6.383
Coherence Score: 0.544
Coherence Per Topic
Topic 3: 0.587
Topic 2: 0.562
Topic 0: 0.536
Topic 1: 0.492
```

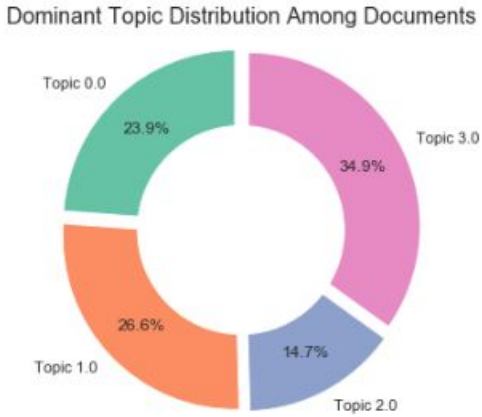
```
Individual Business - Positive Reviews (Trigrams)

Variational Bound: -41914
Log Perplexity: -9.082
Coherence Score: 0.536
Coherence Per Topic
Topic 2: 0.594
Topic 0: 0.543
Topic 1: 0.540
Topic 3: 0.466
```

Individual Business - Negative Reviews (Trigrams)

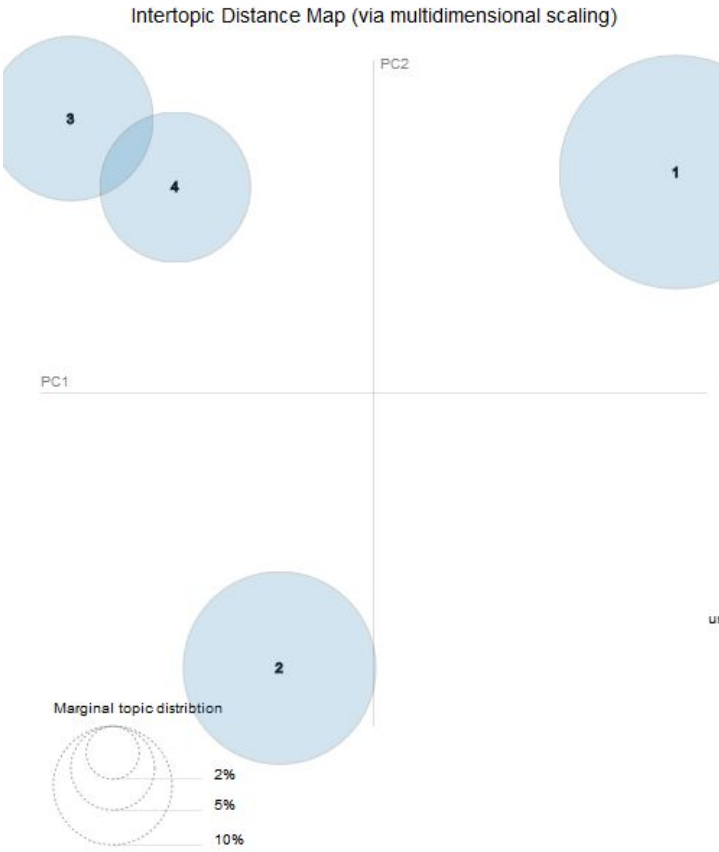


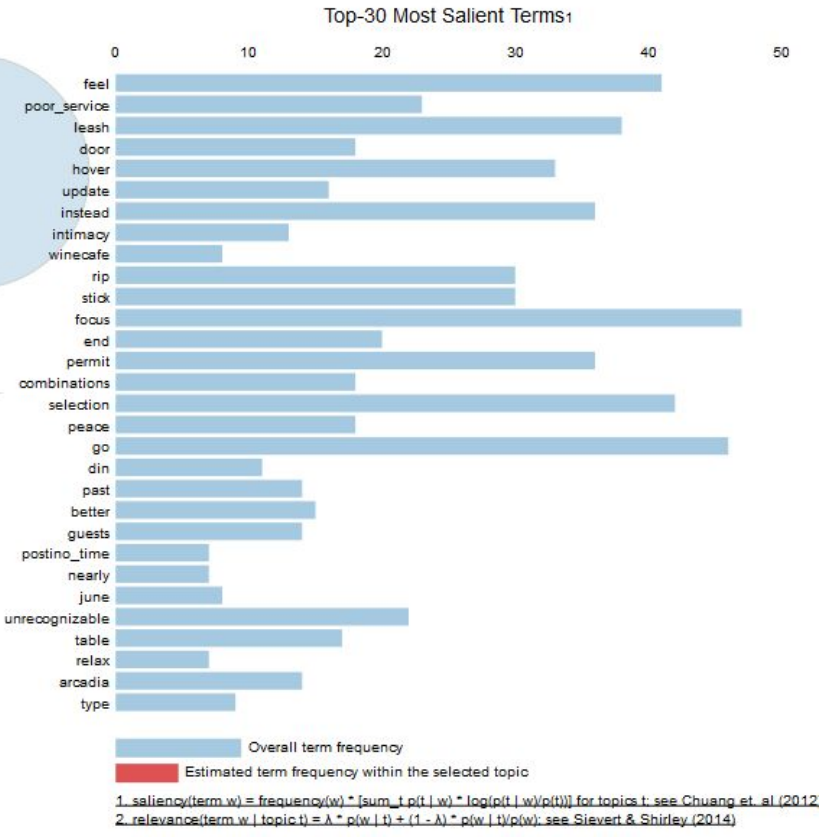




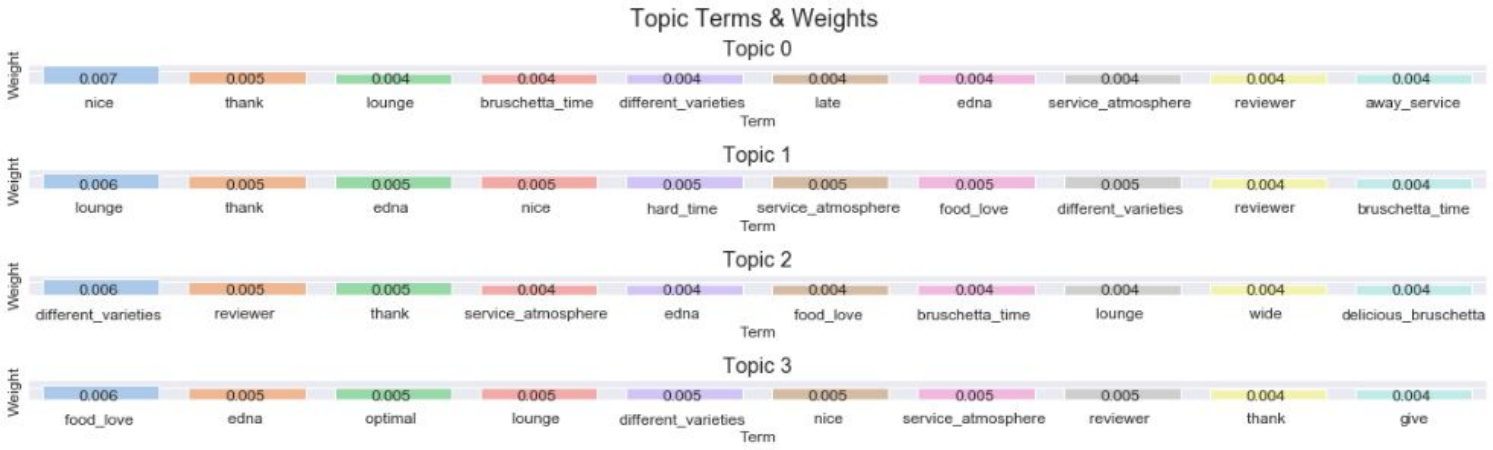


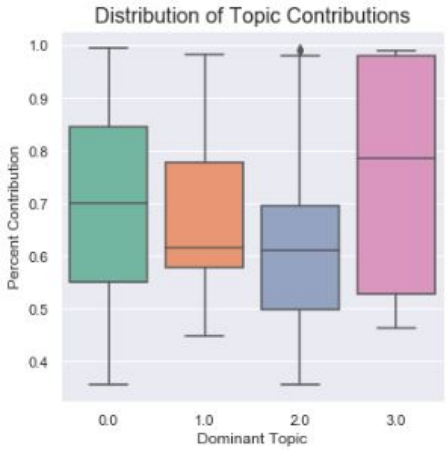
Topic	Topic_%_Contribution	Topic_Keywords	Most_Representative_Text
0.0	0.9990	go, focus, good, feel, policy, hover, overprice, leash, deep, good_service	This was my favorite restaurant. But not anymore. Today is my Birthday & thanks to them my day i...
1.0	0.9984	feel, policy, instead, good, selection, go, overprice, focus, rip, unrecognizable	We decided to visit this place while in the area based on all of the strong reviews from our fel...
2.0	0.9982	leash, selection, focus, poor_service, permit, hover, stick, policy, good_service, instead	Postino is normally one of our favorite local spots to grab a drink and some dinner. Last night ...
3.0	0.9991	policy, selection, good_service, good, overprice, focus, go, permit, leash, deep	I won't even venture to guess why so many people like this place. But a few notes: no, it is not...

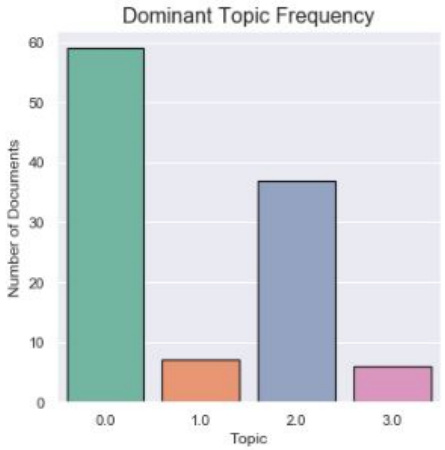


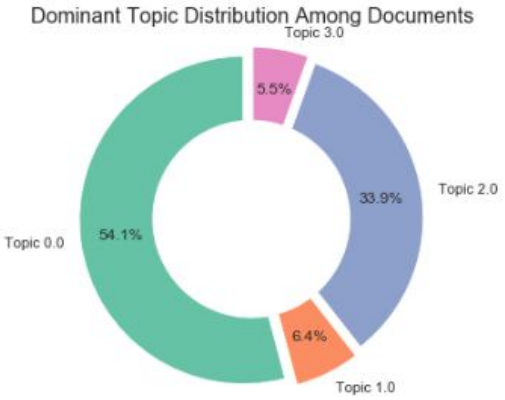


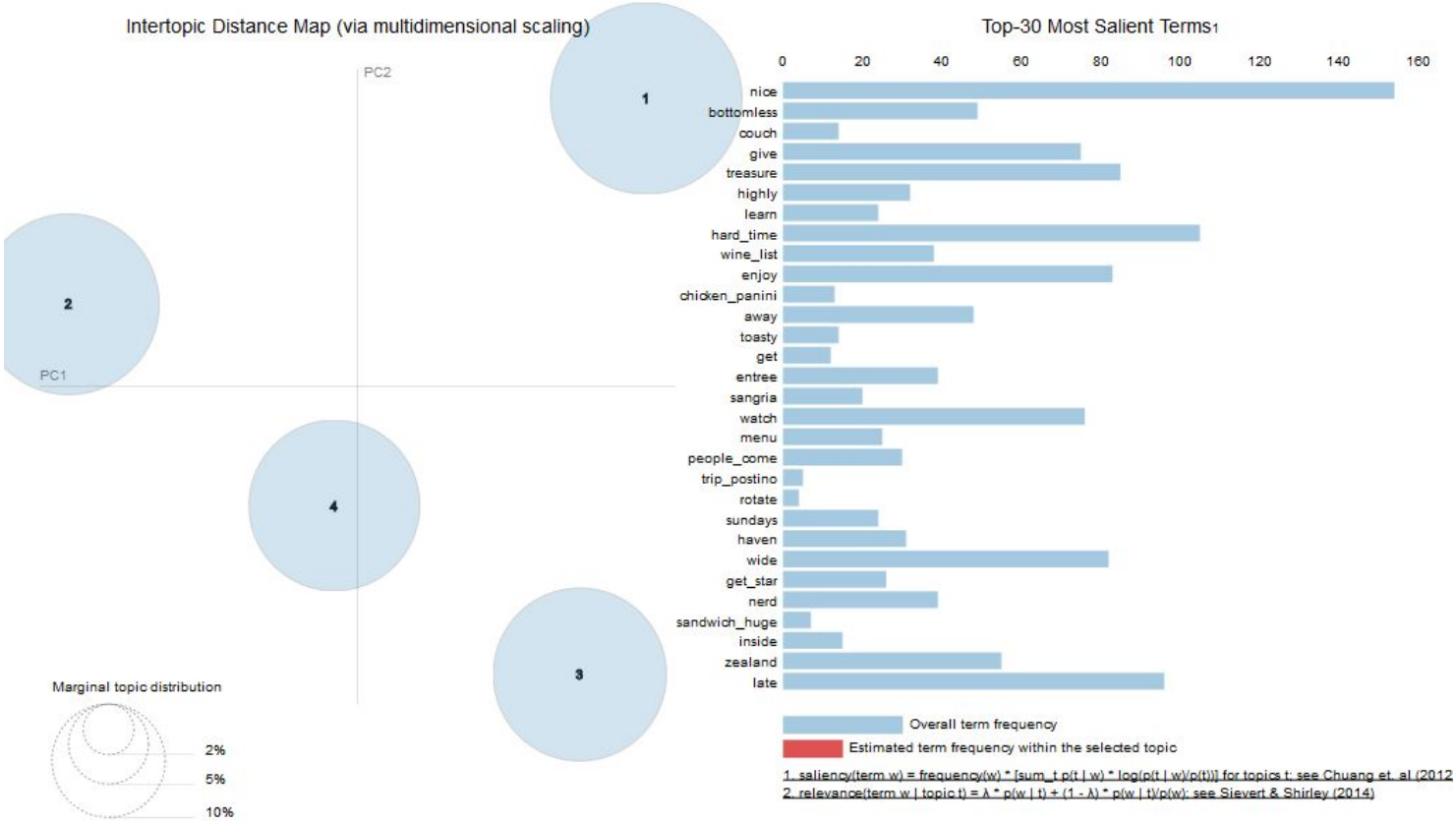
Individual Business - Positive Reviews (Trigrams)











Topic	Topic_%_Contribution	Topic_Keywords	Most_Representative_Text
0.0	0.9959	nice, thank, lounge, bruschetta_time, different_varieties, late, edna, service_atmosphere, revie...	Visiting Postino Wine Cafe makes me feel like I am in another city... an urban, metro, pedestria...
1.0	0.9831	lounge, thank, edna, nice, hard_time, service_atmosphere, food_love, different_varieties, review...	Nothing worth writing about except the bruschetta. Most of the wine tastes like cheap box wine a...
2.0	0.9924	different_varieties, reviewer, thank, service_atmosphere, edna, food_love, bruschetta_time, loun...	I came here to meet a friend who I hadn't seen in ages. Some how the waitress assumed we just ne...
3.0	0.9911	food_love, edna, optimal, lounge, different_varieties, nice, service_atmosphere, reviewer, thank...	Food was ok bread was stale on my sandwich. But service was not good at all! God for bid you ask...

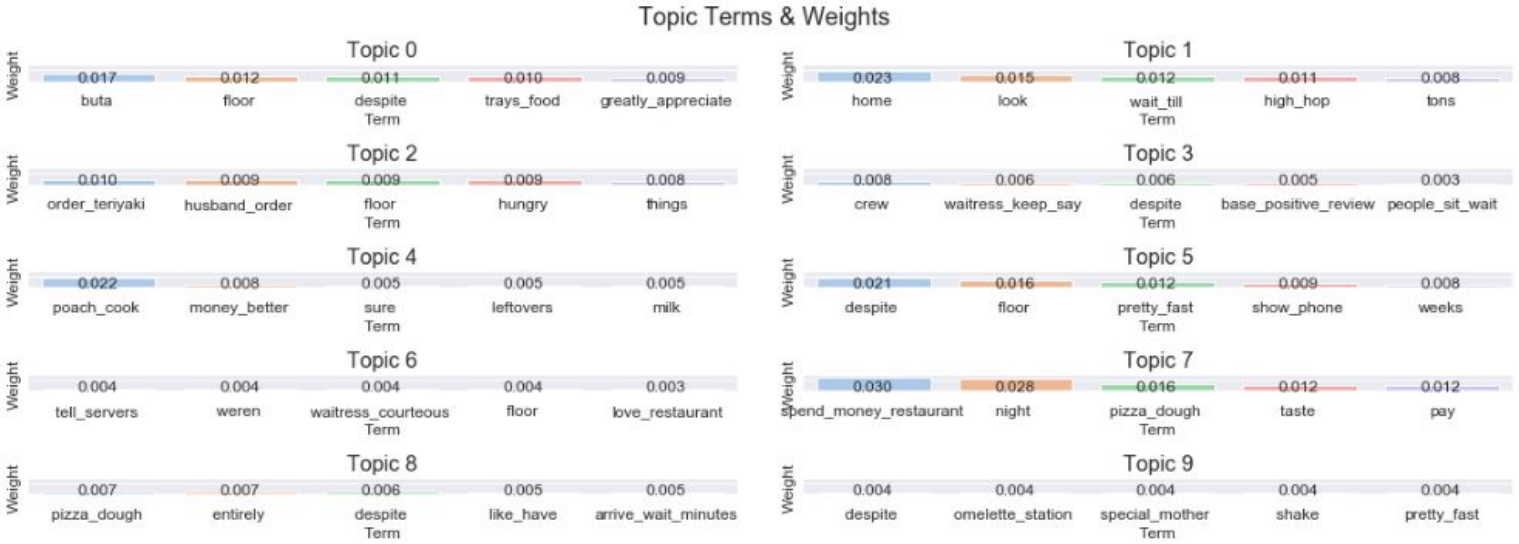
LDA Pipeline Implementation - Food Industry Example

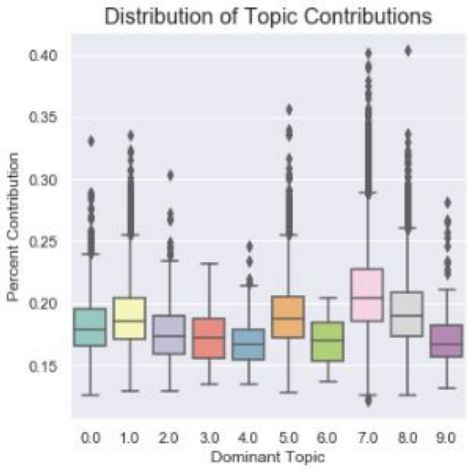
The LDA Pipeline can also be used with a more global perspective, analyzing large datasets with a wide variety of constituents. In this example we will be taking another look at the food industry reviews utilized in the Baseline model. Any pre-selected subset of reviews can be addressed by building a custom pipeline that does not contain the Yelp Review Selector transformer, and only the Preprocess, Vectorize, and LDA_Estimator objects. This time during preprocessing trigrams were created from the corpus, and the vocabulary drawn from it was filtered to remove the three most frequent tokens and any tokens that occur in less than ten instances. The large volume of documents allows the data to be split into training and test sets without jeopardizing the efficacy of the LDA model, and doing so also provides a use case for the pipeline when one seeks to draw topics from a particular vertical or a large geographical area, which may need to be trained and then later tested against new emerging data.

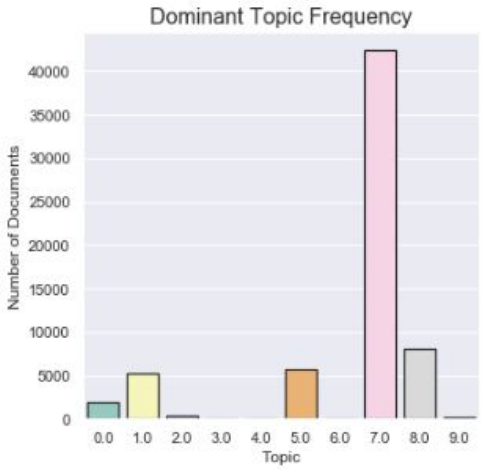
```
Food Industry - Negative Reviews (Trigrams)

Variational Bound: -5452816
Log Perplexity: -11.207
Coherence Score: 0.542
Coherence Per Topic
Topic 6: 0.668
Topic 3: 0.645
Topic 9: 0.578
Topic 2: 0.563
Topic 4: 0.553
Topic 0: 0.544
Topic 7: 0.474
Topic 1: 0.471
Topic 8: 0.470
Topic 5: 0.454
```

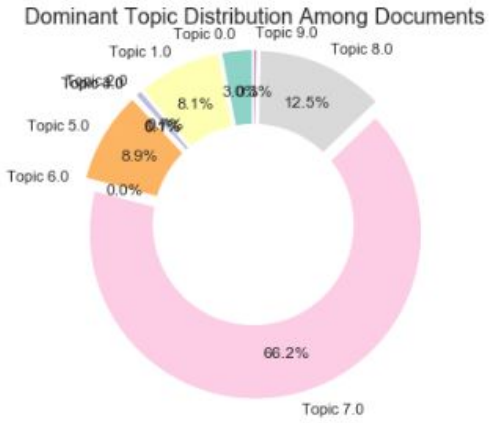
```
Food Industry - Positive Reviews (Trigrams)

Variational Bound: -16619774
Log Perplexity: -11.544
Coherence Score: 0.525
Coherence Per Topic
Topic 5: 0.598
Topic 4: 0.582
Topic 0: 0.550
Topic 1: 0.549
Topic 2: 0.544
Topic 7: 0.521
Topic 9: 0.503
Topic 3: 0.476
Topic 6: 0.476
Topic 8: 0.456
```

Food Industry - Negative Reviews (Trigrams)

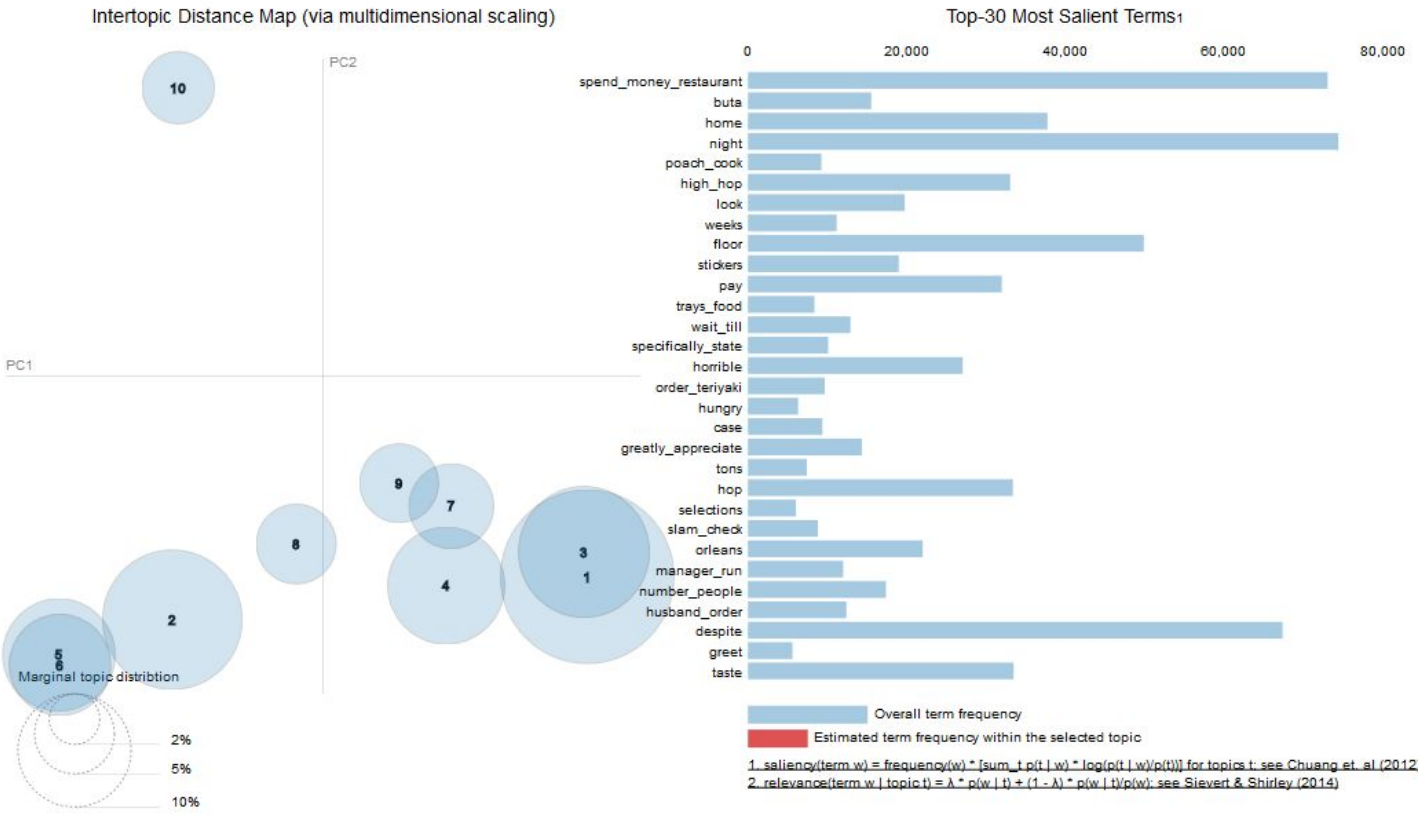




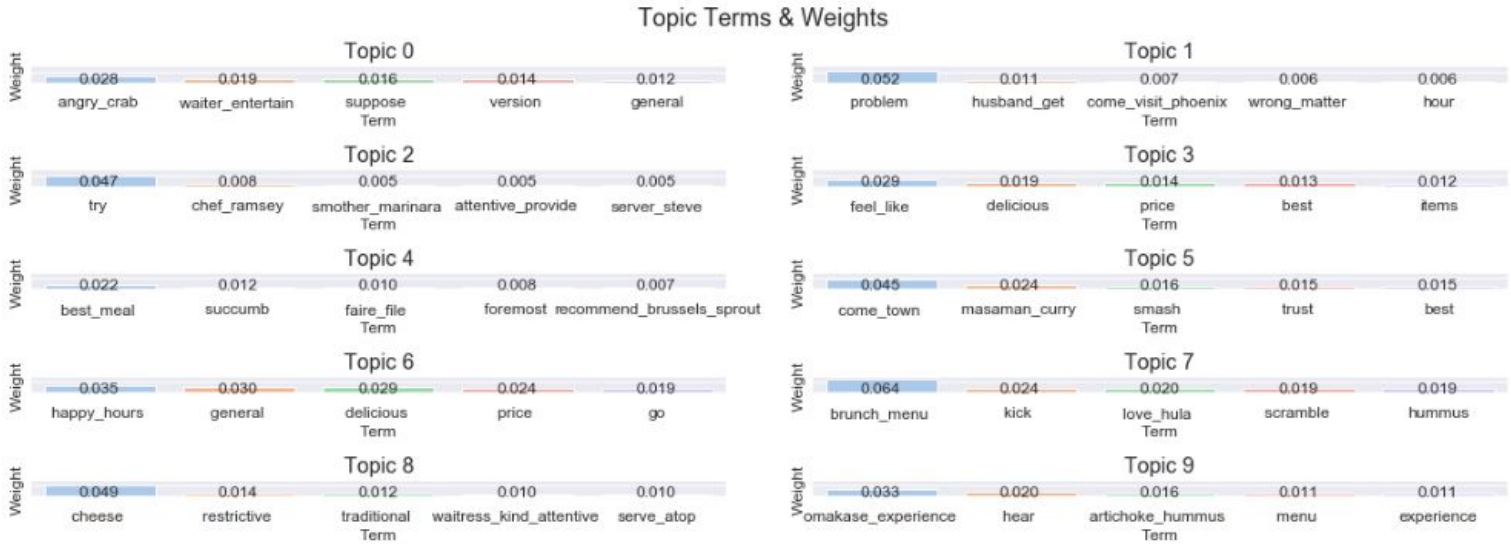


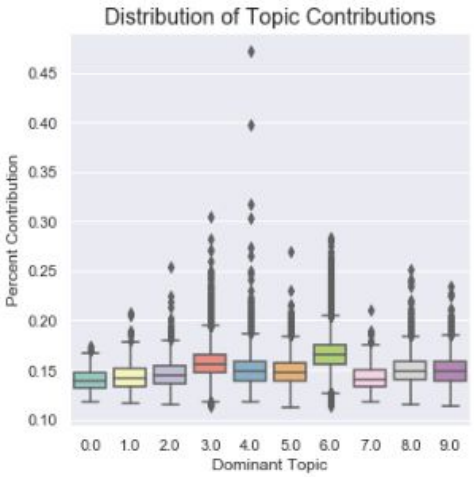


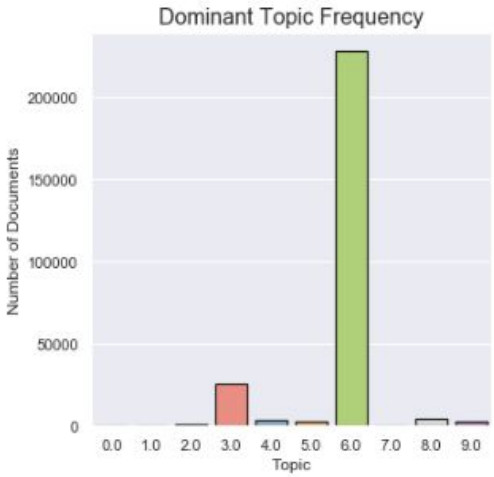
Topic	Topic_%_Contribution	Topic_Keywords	Most_Representative_Text
0.0	0.3308	buta, floor, despite, trays_food, greatly_appreciate, selections, manager_run, greet, decently_f...	Always on the top of the Brunch list, I had to try it, to my dismay, it did not live up to it's ...
1.0	0.3360	home, look, wait_till, high_hop, tons, slam_check, place_try_hard, despite, waitress_come_table,...	Server was a straight bitch , didn't add my blazin rewards after I personally gave her my number...
2.0	0.3039	order_teriyaki, husband_order, floor, hungry, things, night, shift, service_food_good, despite, ...	Originally, I would have rated them five stars because I thought their food was awesome. At the...
3.0	0.2315	crew, waitress_keep_say, despite, base_positive_review, people_sit_wait, potato, price_match, pr...	Ahhh, Grimaldis, I love you so...BUT you really disappointed me today. We were at the mall with...
4.0	0.2456	poach_cook, money_better, sure, leftovers, milk, home, chinese_mexican_food, waitress_come_table...	So this Italian Restaurant was on my to do wish list so I finally gave it a try and I was dissap...
5.0	0.3568	despite, floor, pretty_fast, show_phone, weeks, poorly, specifically_state, couple_time, case, t...	I used to love coming here, the smell of someone else's Gandhi would trigger my own urge to get ...
6.0	0.2046	tell_servers, weren, waitress_courteous, floor, love_restaurant, lack_authenticity, especially, ...	Oyster specials were good and meaty for $2.50 each.\n\nI came here to try the Live Lobster pho t...
7.0	0.4020	spend_money_restaurant, night, pizza_dough, taste, pay, hop, horrible, food_inconsistent, high_h...	There are a lot of high reviews and 5 stars for this place and I am confused as to why. I in no ...
8.0	0.4044	pizza_dough, entirely, despite, like_have, arrive_wait_minutes, serve, stone_cold, pay, waitress...	Avant de commencer, je dois dire que je ne connais rien Ã la cuisine polonaise et que ce restau...
9.0	0.2819	despite, omelette_station, special_mother, shake, pretty_fast, taste, binge, close_kitchen, yest...	After hearing so many people talk about shake shack we all had to go check it out... After waiti...

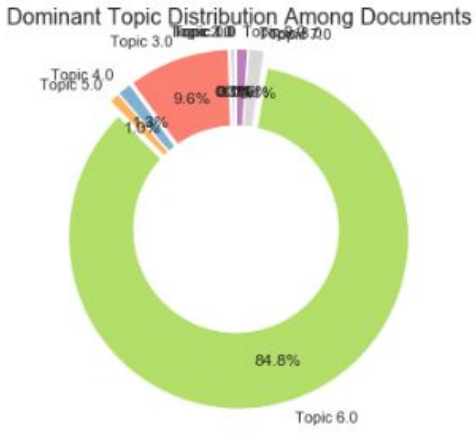


Food Industry - Positive Reviews (Trigrams)

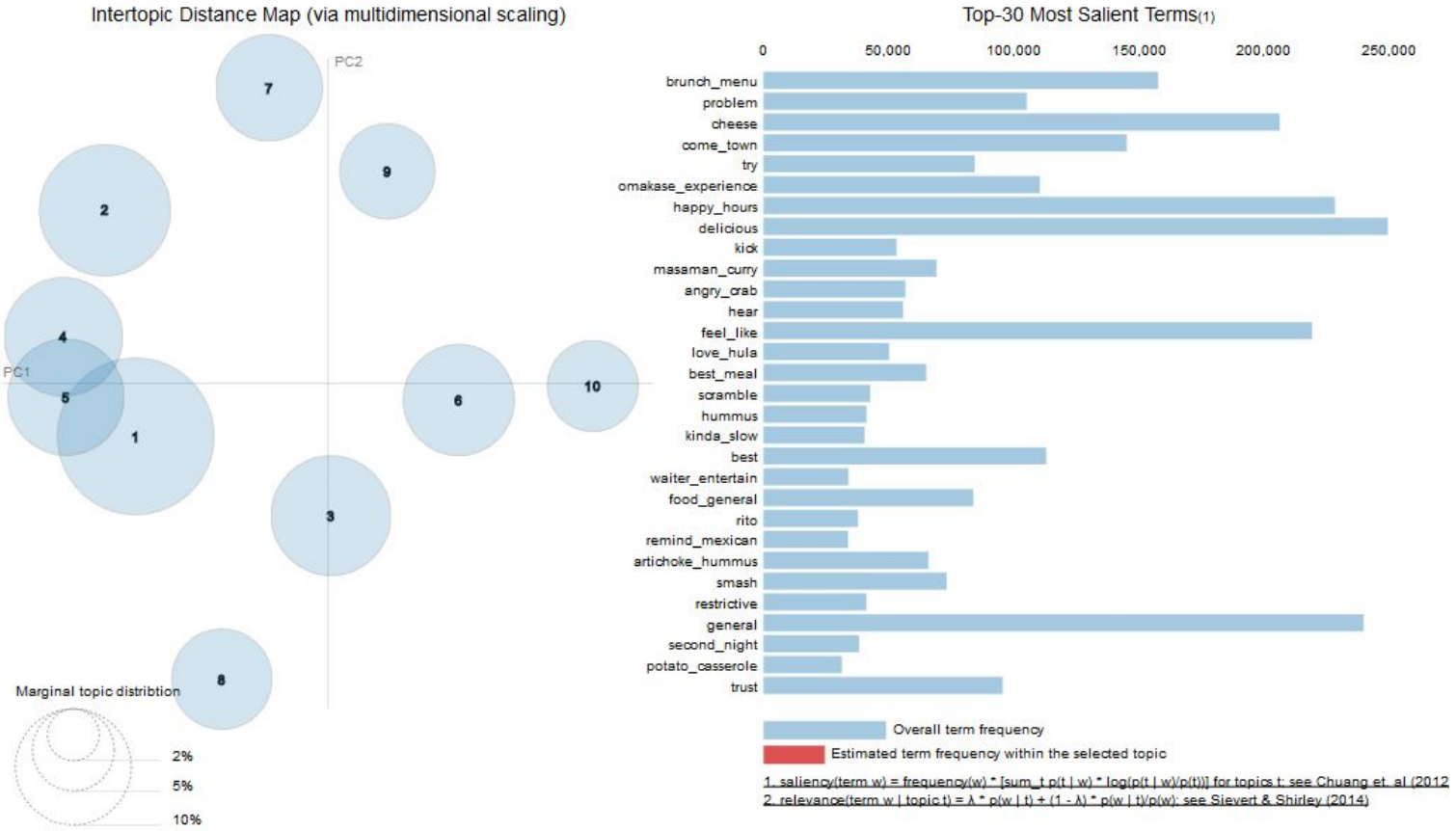








Topic	Topic_%_Contribution	Topic_Keywords	Most_Representative_Text
0.0	0.1744	angry_crab, waiter_entertain, suppose, version, general, chef_ramsey, surely, original, price, c...	Its a bit difficult to say if the food here is any good or not. I'm writing this review while go...
1.0	0.2078	problem, husband_get, come_visit_phoenix, wrong_matter, hour, menu, quality_quantity, nutella_ba...	I came in on thursday to order a 50th anniversary cake. The lady helping me was friendly and he...
2.0	0.2538	try, chef_ramsey, smother_marinara, attentive_provide, server_steve, north_phoenix, boyfriend_ta...	The atmosphere is really elegant at this place especially at the top floor. Our server did a rea...
3.0	0.3052	feel_like, delicious, price, best, items, run, meat_combo_platter, cheese, second_night, toast	Famous Daves... Famous Daves....\nYes, I know a few.\n\nDavid Lynch \nDavid Cassidy \nDavid Bore...
4.0	0.4726	best_meal, succumb, faire_file, foremost, recommend_brussels_sprout, dress, artichoke_hummus, fe...	Buffet-Restaurants haben fast alle Hotels in Las Vegas. Im teuren Vegas eine MÃ¶glichkeit, gut u...
5.0	0.2690	come_town, masaman_curry, smash, trust, best, rito, flavorful, ahead_time, delicious, hand	I am no expert on soul food, but the fried chicken is finger lickin' good. Please don't sue me, ...
6.0	0.2841	happy_hours, general, delicious, price, go, plastic_bag, say_hour_wait, food_general, feel_like,...	This is definitely the best bar in the area. It has a ton of British charm, European and domesti...
7.0	0.2109	brunch_menu, kick, love_hula, scramble, hummus, kinda_slow, remind_mexican, potato_casserole, fe...	I totally agree with the other Yelpers! It is the best authentic Chinese food in Vegas. Prices...
8.0	0.2511	cheese, restrictive, traditional, waitress_kind_attentive, serve_atop, taste, crisp_fresh, best,...	Sunday at Noon and we were able to get a table immediately even though it was a packed house. F...
9.0	0.2350	omakase_experience, hear, artichoke_hummus, menu, experience, terrace_cafe, pineapple_fry, mexic...	Hints: Bone-In Rib Steaks\nBone-In Rib Steaks\nBone-In Rib Steaks\nBone-In Rib Steaks\nBone-In R...



VIII. Client Recommendations

Based on the possible implementations of the product constructed and the results that may obtained from it, the following are some insightful uses that I would promote to any future clients:

- To determine the direction of favorability for a particular product or initiative
    - Success of a loss-leader in creating profitable upsell scenarios
    - Response to rebranding or new marketing strategy
    - Reception of an aesthetic change such as a remodel or relocation
- To assess global versus local analysis
    - Alongside a market penetration analysis to boost insight
    - Consistency in customer service across localities for a chain of businesses
    - Distinguishing between circumstantial and pervasive issues for a particular industry

IX. Conclusions and Future Work

In constructing this text analysis pipeline I encountered several problems consistent with common parables in reference materials, a notable one being that the quality of the topic model output rests tremendously on the quality of the preprocessing steps taken prior to training. The initial extraction of the text data from the overall Yelp academic dataset, and the subsequent transformation into a serviceable vector that can be used to train an LDA model, were both some of the most computationally expensive and consequential aspects of the overall procedure. Minor changes in these steps, such as filtering the vocabulary or adding n-grams to the corpus, often led to drastically differing results, which themselves proved uniquely challenging to interpret. Moving forward I would add more visualizations such as word clouds to assist in readability, as the success of each topic model created hinges on a combination of the user's domain knowledge and the ease in which it can be interpreted.