**ASPECT BASED SENTIMENT ANALYSIS**

**Introduction:**

**Sentiment analysis of reviews can determine the opinion (positive/negative/neutral) of a user about an organisation/entity (restaurant) on the whole but not the subject/attributes (food, service, ambience etc.) specifically. Determining the aspects related to the user sentiment is way more actionable from the business point of view rather than just overall sentiment as the users might have good things to say about few things and complaints/bad comments about some other things. It will also help customers in deciding better which place/entity to choose for specific needs.**

**Data Set Used:**

The dataset has been collected from SEMEVAL 2016 Task 5 Subtask 1(Task description mentioned below).

**Task Description:**

**Subtask 1: Sentence-level ABSA**. Given an opinionated document about a target entity (e.g. a laptop, a restaurant or a hotel), the goal is to identify all the opinion tuples with the following types of information:

* **Slot 1**: **Aspect Category Detection**. The task is to identify every entity E and attribute A pair towards which an opinion is expressed in the given text. E and A should be chosen from predefined inventories of entity types (e.g. LAPTOP, MOUSE, RESTAURANT, FOOD) and attribute labels (e.g. DESIGN, PRICE, QUALITY). The E, A inventories for each domain are described in the respective annotation guidelines documents.
* **Slot 2**: **Opinion Target Expression (OTE)**. The task is to extract the OTE, i.e., the linguistic expression used in the given text to refer to the reviewed entity E of each E#A pair. The OTE is defined by its starting and ending offsets. When there is no explicit mention of the entity, the slot takes the value “NULL”. This slot will be required only for particular datasets/domains.
* **Slot 3**: **Sentiment Polarity**. Each identified E#A, OTE tuple has to be assigned one of the following polarity labels: positive, negative, or neutral (mildly positive or mildly negative sentiment).

**Data Description:**

The dataset was provided in XML format with 350 reviews and a total of 2000 sentences. The dataset has been converted to CSV format for further analysis by dropping the sentences from reviews for which the corresponding opinion categories are not provided. Every sentence that has multiple opinion categories (category) have been repeated with the corresponding polarity.

Number of records/rows: 2507

Number of attributes/columns: 6

Names of attributes/columns:

RID : Review ID

SID : Sentence ID for each review

text : Review text

polarity : Sentiment associated with the opinion category (to be predicted)

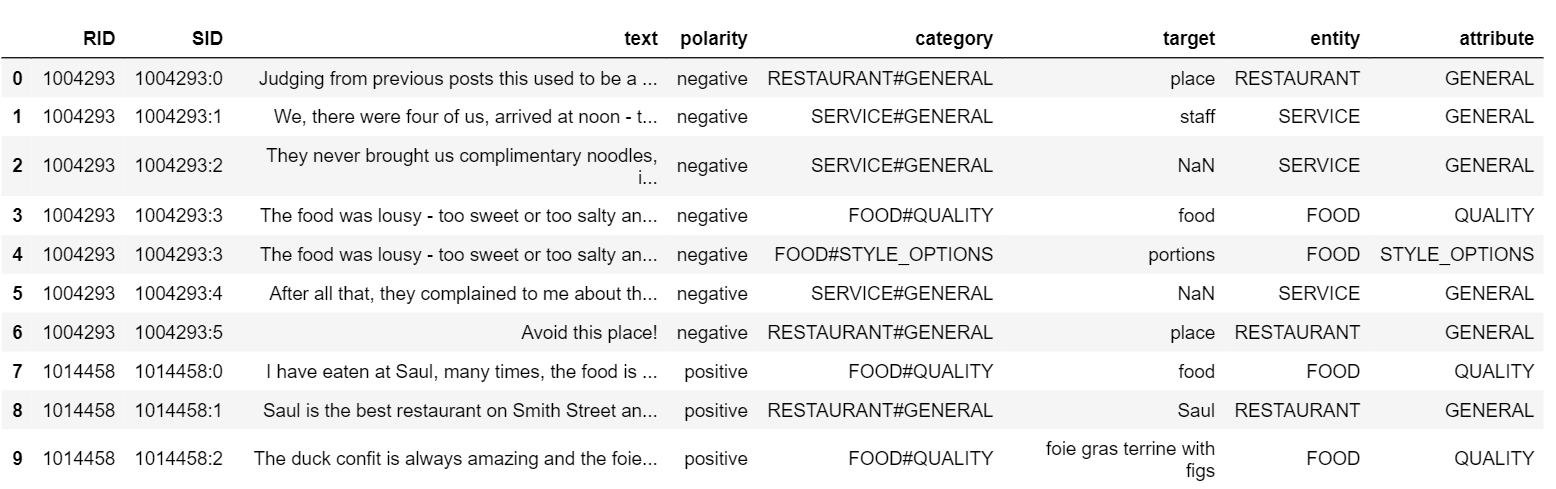
category : opinion category/aspect (to be predicted)

target : the linguistic expression used in the given text to refer to the reviewed entity

entity : aspect category (extracted from category)

attribute : aspect term (extracted from category)

Preview of data:



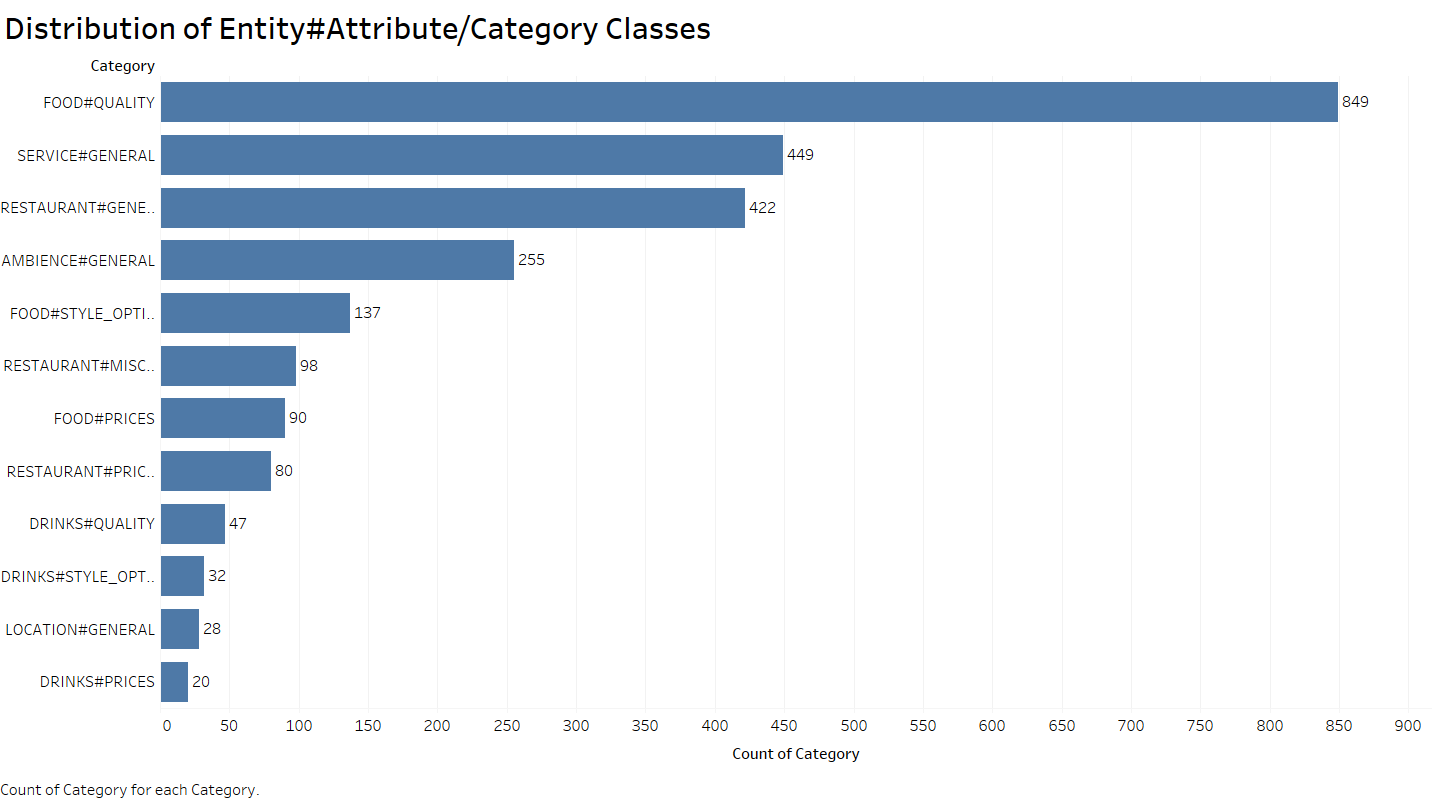
**Data Analysis:**

Number of missing values present if any:

target : 627

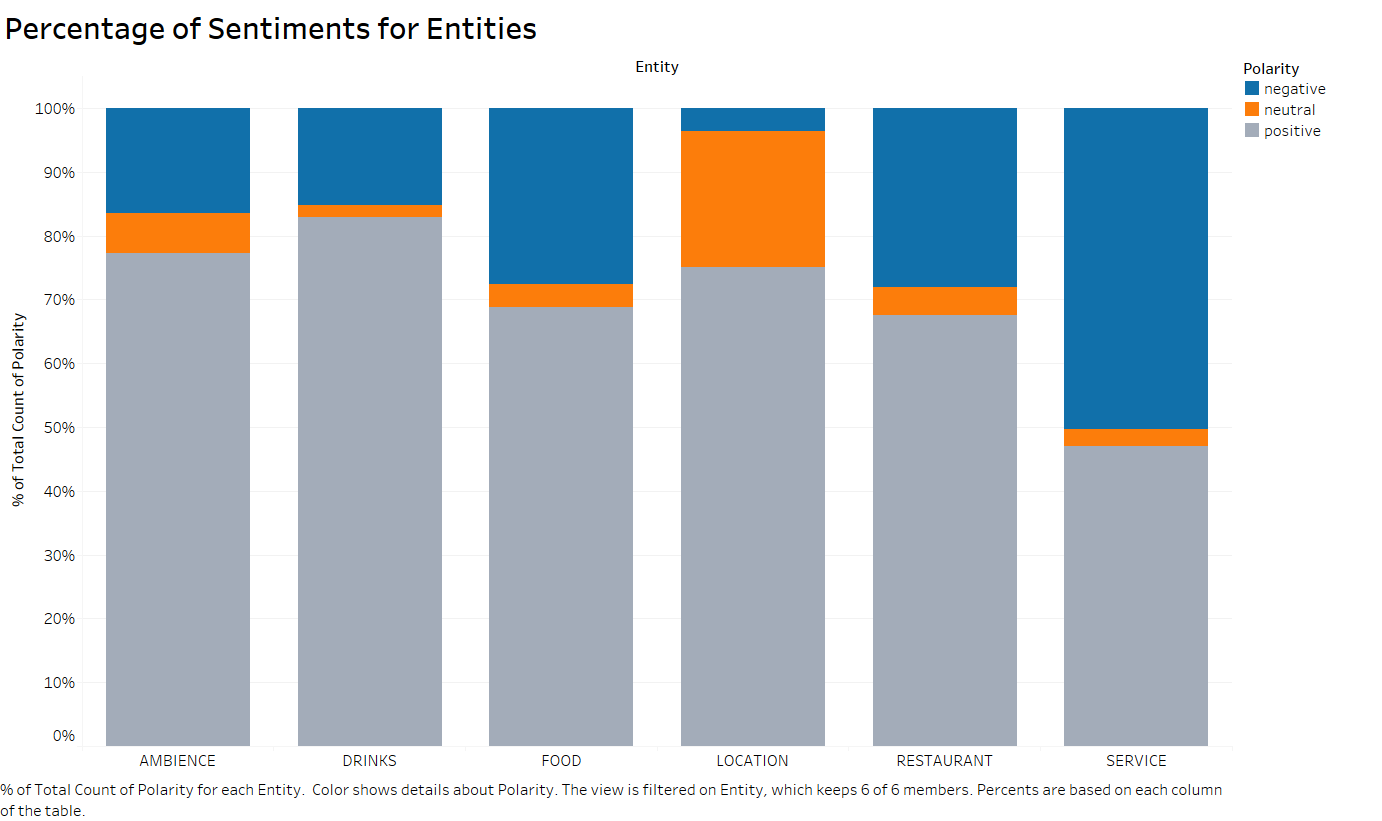
As it is required to classify the sentiments of given reviews with respect to every opinion category present in the review, let us first understand how the data is distributed among these sentiments and opinion categories.

Following is the distribution of Opinion Category Classes (12):

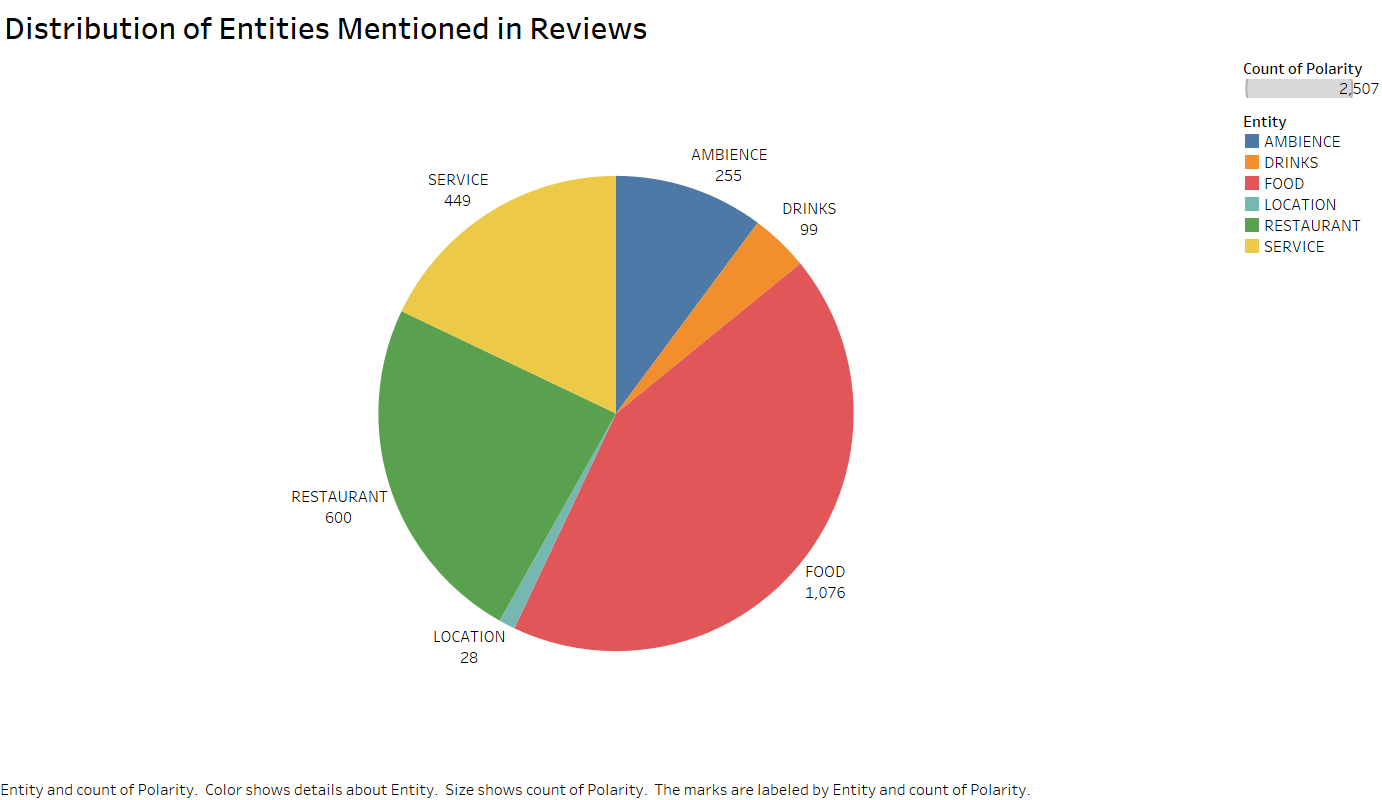


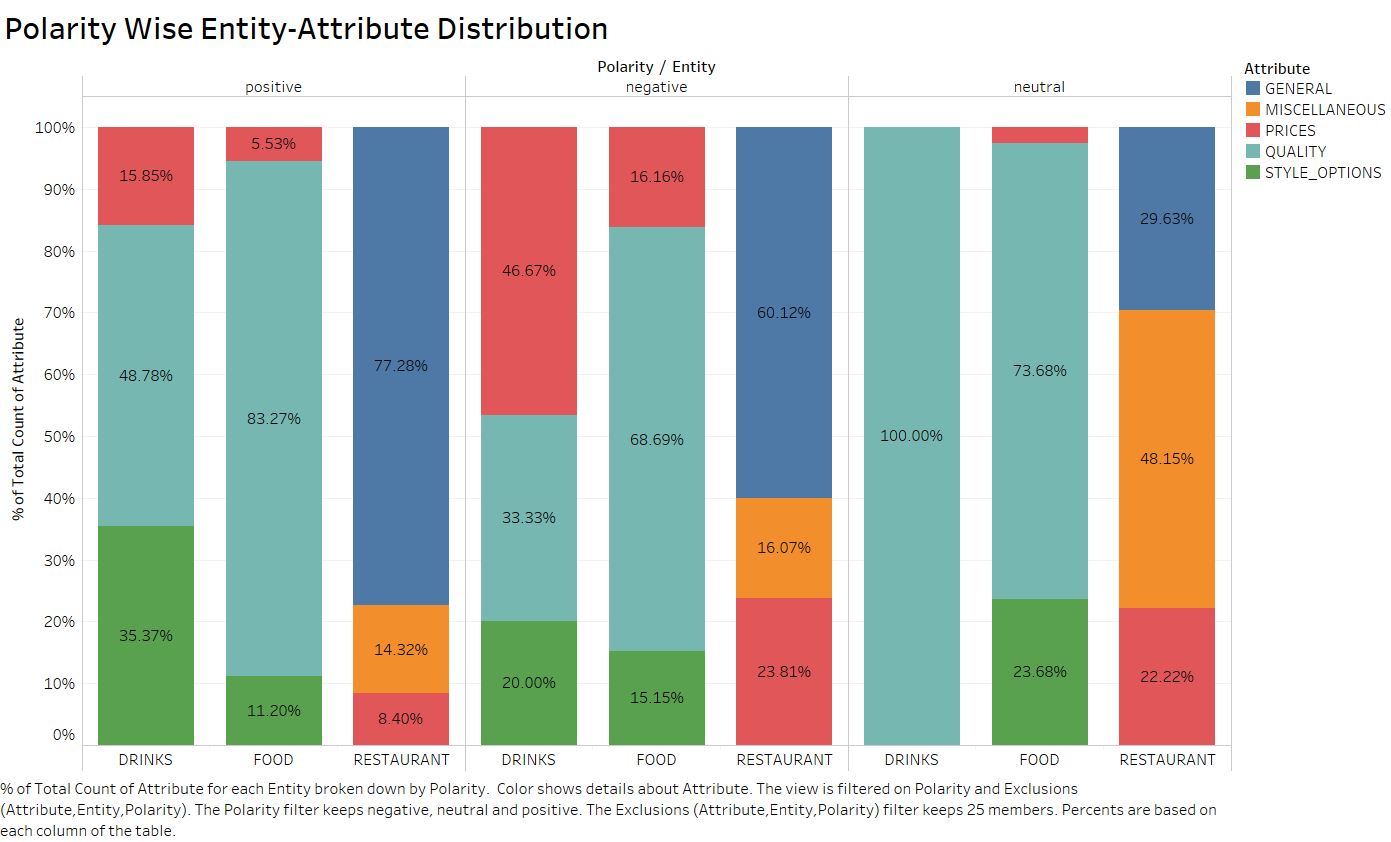
It is observed from the above graph that the opinion categories are highly imbalanced with majority of data belonging to food quality, general reviews on service and restaurant. They constitute to about 69% of the entire data and the other 6 classes constitute to only 31%.

Let us now try to analyse the sentiment distribution with respect to each which helps the business owner to get a sense of which aspects the customers are satisfied about their business, and which ones they need to improve on. Below plot depicts the percentage of polarity for each of the aspect categories at large.



Of course, this only tells you the simple ratio and not the volume, so you probably also want to find out which aspects people are most opinionated about. As expected for a restaurant, people seem to be mostly concerned about food.





The polarity wise entity-attribute distribution provides a clear comparison of entities along with their corresponding attributes with respect to the polarity of users. For example, if we compare the entity drinks with respect to positive and negative polarity, we observe that users are more positive about the style options and quality of drinks whereas comparatively more negative about their prices.

Now let us try understanding the hidden patterns in the data to understand the aspects of user opinions and their corresponding polarity value.

**Topic Modelling Using Latent Dirichlet Allocation (LDA) Algorithm:**

To extract/identify multiple aspects from a given sentence (statement in a review), we need to understand the distribution of words corresponding to each aspect (topic) and then find the top most aspects for each statement. This can be achieved by Latent Dirichlet Algorithm (LDA) which works on a similar principle that helps in identifying topics (here, aspects) present in a text object.

LDA assumes that documents are produced from a mixture of topics and those topics then generate words based on their probability distribution.

LDA processes each document as bag of words and returns the probability distribution of each topic (here, aspect categories).

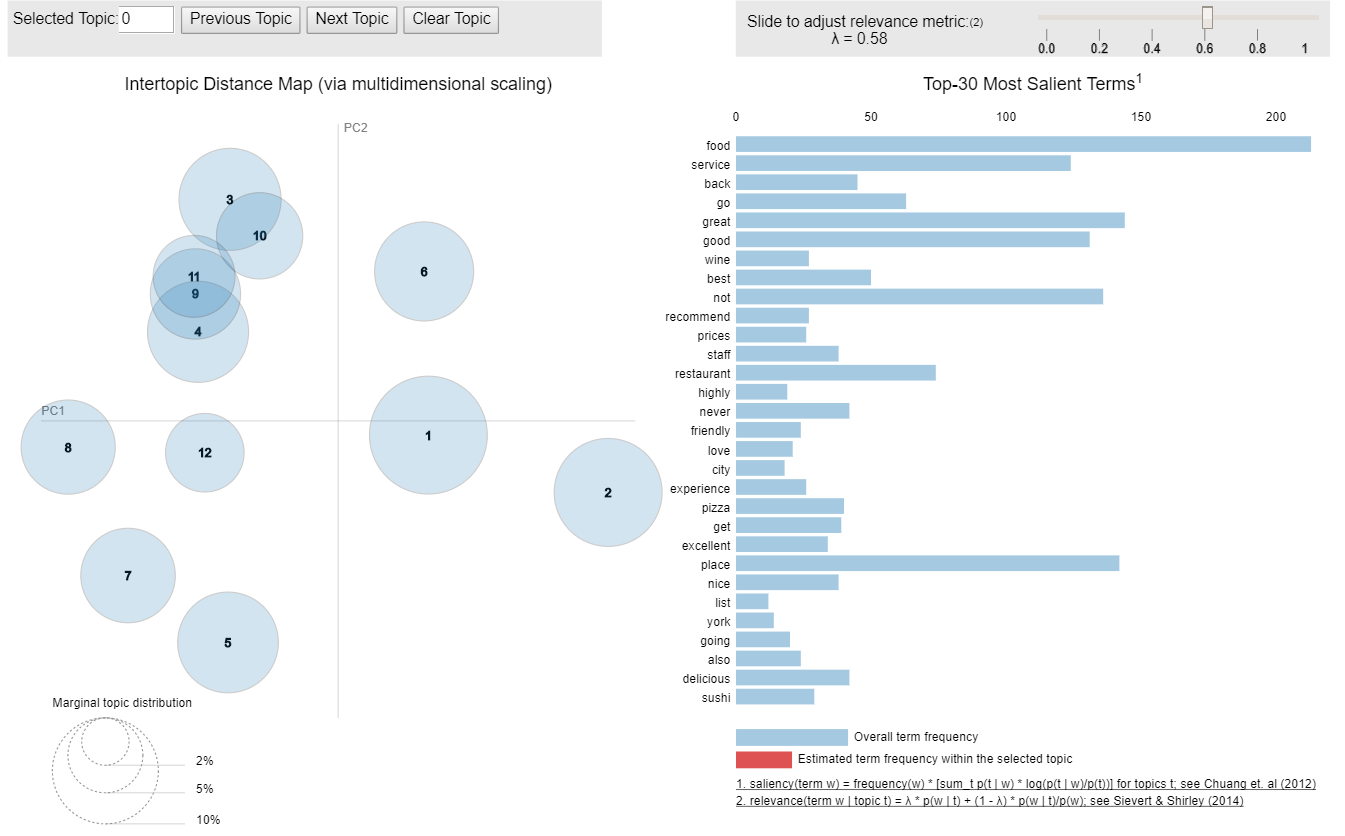
For the purposes of LDA we need a way of summarizing our documents. We need to keep count of the following:

1. Topic Term Matrix: In the case of LDA, a single unique word can be assigned a different topic for each instance of the word
2. Document Topic Matrix: This is the total number of words assigned to each topic in each document
3. Topic Sum Vector: It keeps count of the total number of words in the entire corpus(all docs together) assigned to each topic

**Pre-processing applied:**

* Removed punctuations
* Removed stop words
* Lemmatization of words
* Eliminated words with POS tags: 'CC','CD','DT','EX','IN','JJS','MD','PRP','PRP$','RBS','RP','TO','UH','WDT','WP','WRB'
* Analysed word count and removed infrequent words (words occurring less than 5 times)
* Created the dictionary and document topic matrix

Following are the visualisations of topic distributions generated by pyLDAvis, a package in python which provides a proper understanding of meaning, prevalence of each topic as well as the relation between topics.

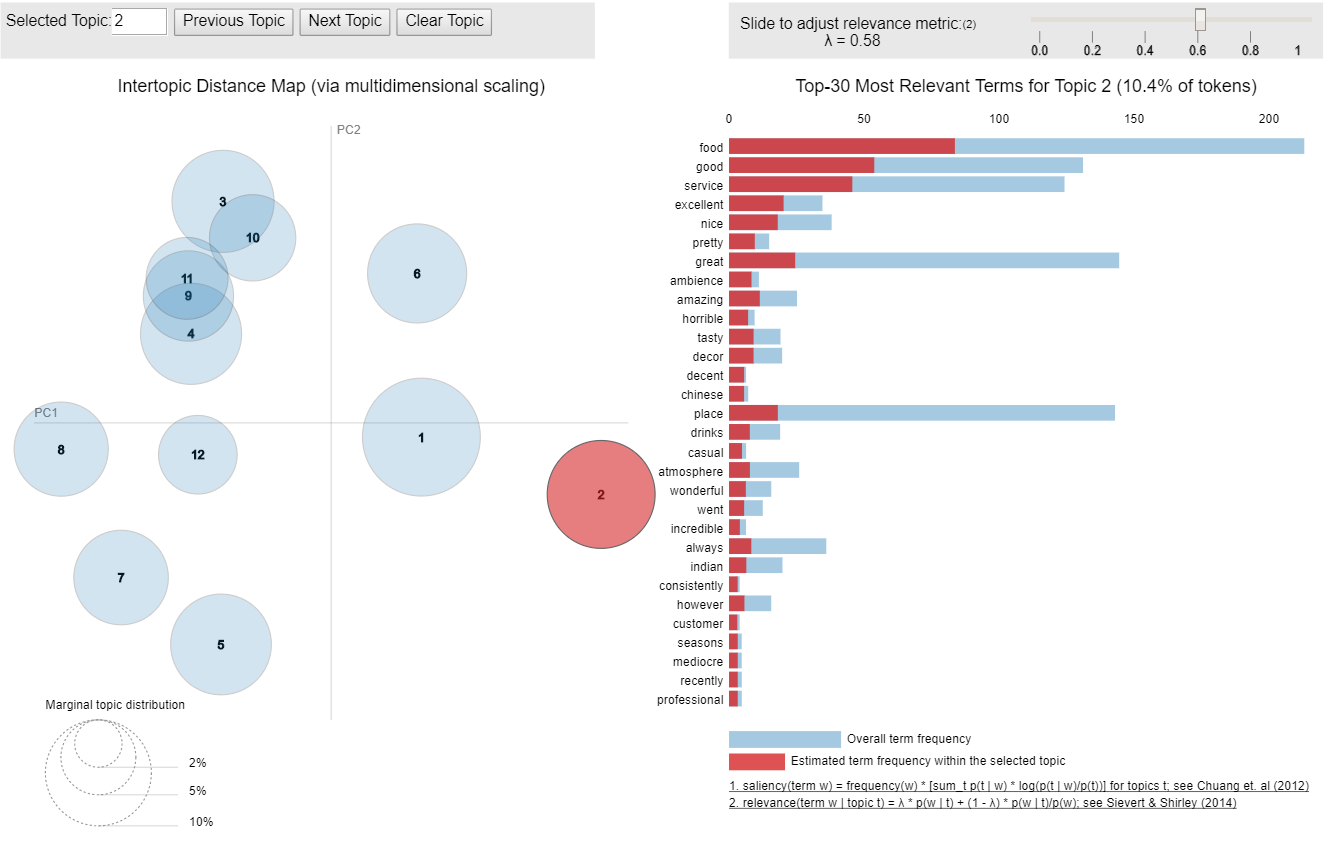


**Fig:** Topic Circles and Top – 30 Most Salient Terms

In these visualisations, the topics are plotted as circles each separated based on the intertopic distance. The size (area) of the circle indicates it prevalence.

The left and right panels of our visualization are linked such that selecting a topic (on the left) reveals the most useful terms (on the right) for interpreting the selected topic. In addition, selecting a term (on the right) reveals the conditional distribution over topics (on the left) for the selected term.

On the right, two juxtaposed bars showing the topic-specific frequency of each term (in red) and the corpus-wide frequency (in blueish grey).



**Fig:** Top – 30 Most Terms Relevant to Topic 2

In the above figure, topic 2 is selected on the left panel and the right panel displays the top 30 terms relevant to topic 2. Based on the terms ambience, décor, decent, atmosphere etc., we might assume that it belongs to ambience aspect. However, it contains other words not relevant to the ambience aspect as well which might be quite confusing to distinguish between various topics.

**Evaluation of LDA:**

Now let us try to look at a more qualitative metric to evaluate our LDA model. Topic Coherence is a measure used to evaluate topic models. It is defined as the average of the pairwise word-similarity scores of the words in the topic.

The coherence score achieved for the LDA model built on the 12 aspect categories is 0.4.

As the coherence score is low and there is also not enough differentiation provided between the topics, we cannot depend on LDA alone for our aspect category classification.

**Classification of Opinion Categories Using TF-IDF Vectors:**

The data containing multiple aspects and polarity values have been pre-processed and classified using machine learning algorithms.

**Pre-processing:**

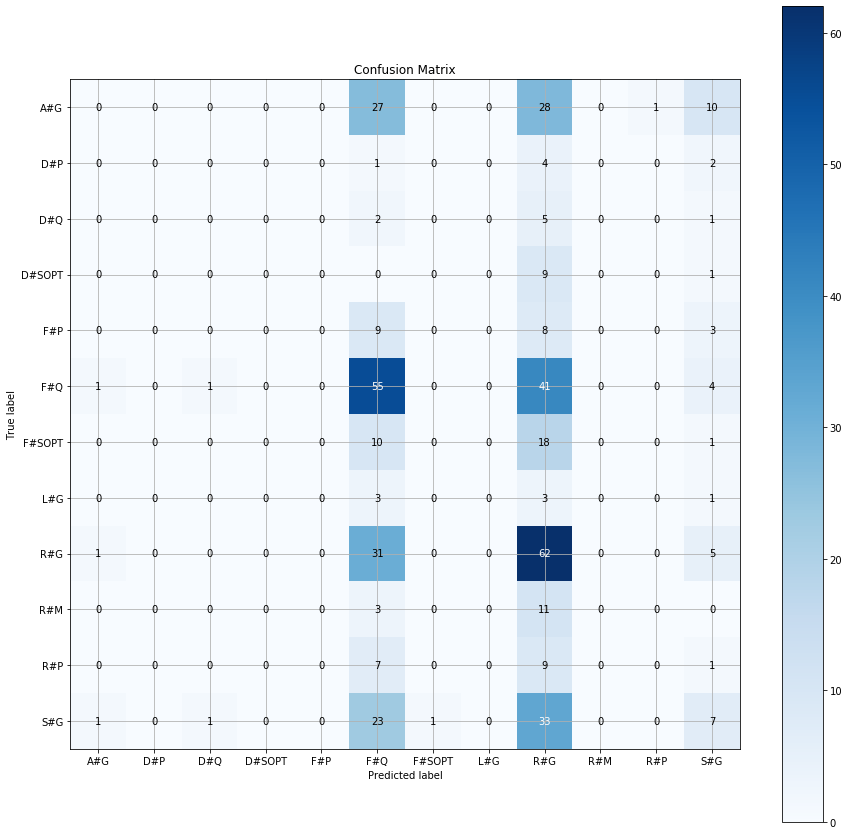
* Removed stop words
* Removed punctuations
* Lemmatization of words
* Eliminated words with POS tags: 'CC','CD','DT','EX','IN','JJS','MD','PRP','PRP$','RBS','RP','TO','UH','WDT','WP','WRB'
* Built TF-IDF Vectors

**Classification:**The given category classes are highly imbalanced and thus need special attention to avoid misclassification of data into majority classes. Tree based algorithms (Random Forest) and cost sensitive learning (SVM) have been applied to increase the cost of misclassification.

However, SVM gave the best performance and was more generalised to test cases. The performance of SVM is shown below.

**Evaluation of SVM Model:**

The accuracy achieved using SVM model is 76% but as we have multiple classes and the data is highly imbalanced, we cannot depend on our accuracy measure to evaluate our model. Let’s look at the confusion matrix to get more clarity on the classification results.



A#G – AMBIENCE#GENERAL

D#P – DRINKS#PRICES

D#Q – DRINKS#QUALITY

D#SOPT – DRINKS#STYLEOPTIONS

F#P – FOOD#PRICES

F#SOPT – FOOD#STYLEOPTIONS

L#G – LOCATION#GENERAL

R#G –RESTAURANT#GENERAL

R#M – RESTAURANT#MISCELLANEOUS

R#P – RESTAURANT#PRICES

S#G – SERVICE#GENERAL

**Fig: Confusion Matrix for SVM Model**

From the confusion matrix, it is evident that majority of the predictions belong to only two classes, namely FOOD#QUALITY and RESTAURANT#GENERAL and very few of them belong to the SERVICE#GENERAL class. There are almost no predictions belonging to any other class.

Furthermore, even between the majority prediction classes there is lot of confusion observed i.e., 31% of RESTAURANT#GENERAL have been wrongly predicted as FOOD#QUALITY and 40% of FOOD#QUALITY have been wrongly predicted as RESTAURANT#GENERAL. Insufficient training data and the presence of multiple aspects in every sentence of a review limits the effectiveness of these models. Efficient methods that reduce the ambiguity in the data need to be used to improve the performance of classification models.

**Using Dependency Parsers for Reduction of Ambiguity in Data:**

Each sentence in a review can be made up of multiple sentences and thus be attached with multiple aspect tags.

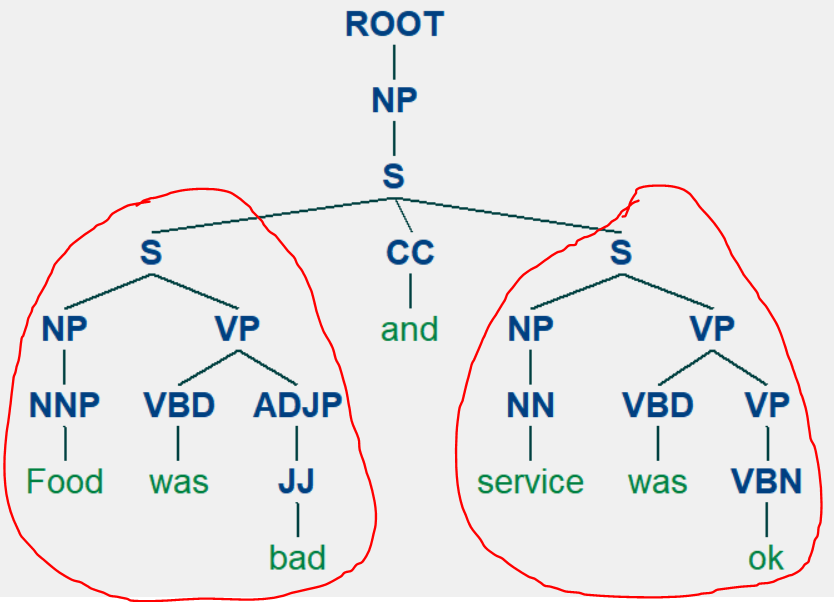
Eg: Food was bad and service was ok – Category: FOOD#QUALITY; Polarity: negative

Food was bad and service was ok – Category: SERVICE#GENERAL; Polarity: neutral

Training our models on such data causes lot of ambiguity between the target classes.

To overcome this problem, an ideal solution would be to break the sentence into sub sentences by analysing its structure. Each sub sentence can then be tagged with the corresponding aspect category.

Here, the sentences have been split into sub sentences based on the notion of S->S (S: Sentence).



Once all the sentences are split and assigned with the corresponding aspect category and polarity value, the data is trained using SVM classifier with TF-IDF vectors to predict aspect category and polarity value for each sub sentence in a sentence. It is expected to generate far better results as it reduces a lot of noise in the data.

**Using LIME package to understand the predictions:**

Even though we have predictions out of a classification model and a good metric value, to fully trust and improve the model, we need to understand on what basis our model has classified a certain sentence as positive, negative or neutral review. For this purpose, LIME, an inbuilt package in python has been used that helps us understand the major terms (features) taken into consideration that led to a certain classification result. Following are the sample results.

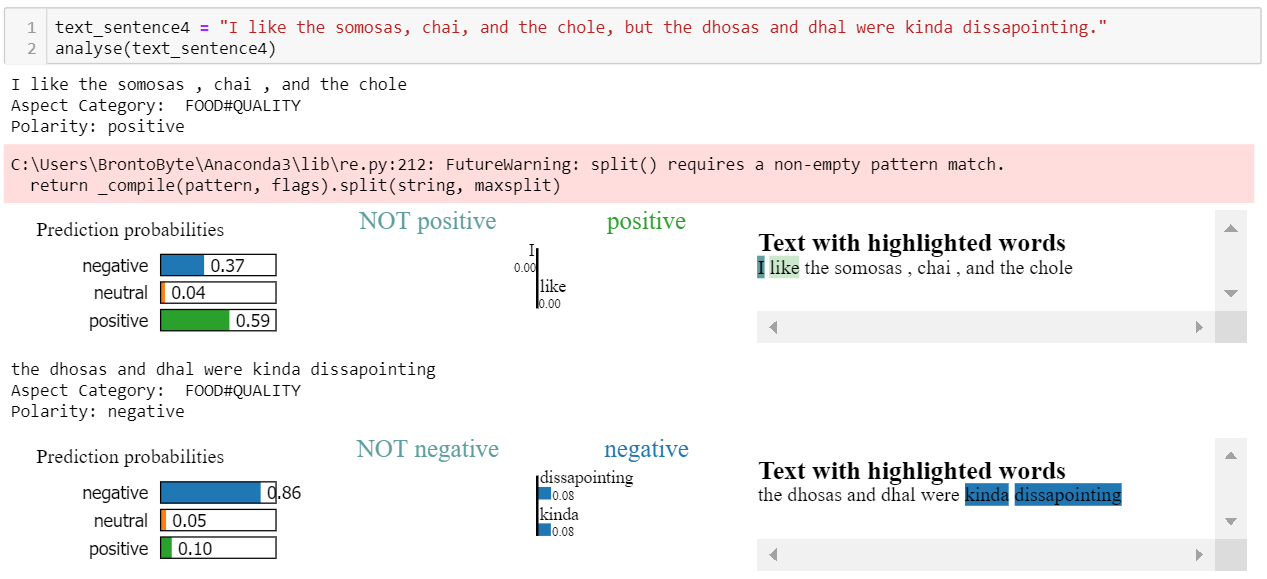
**Results:**

As observed from the below visualisation, the given sentence is split into sub sentence and then for each sub sentence its corresponding aspect and polarity is predicted. Thereafter, the probabilities related to each polarity class is specified and the relevant terms (features) pertaining to the highest probability class are displayed and highlighted in the text.

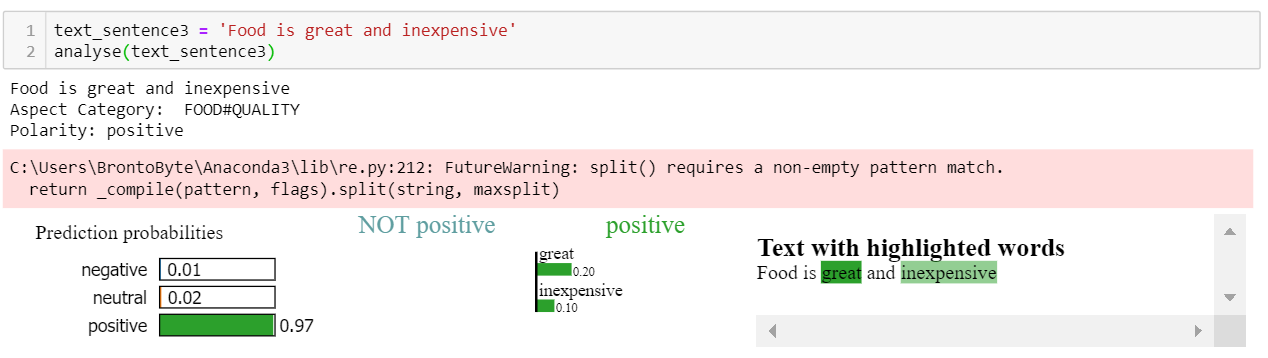
Sample1:



Sample 2:



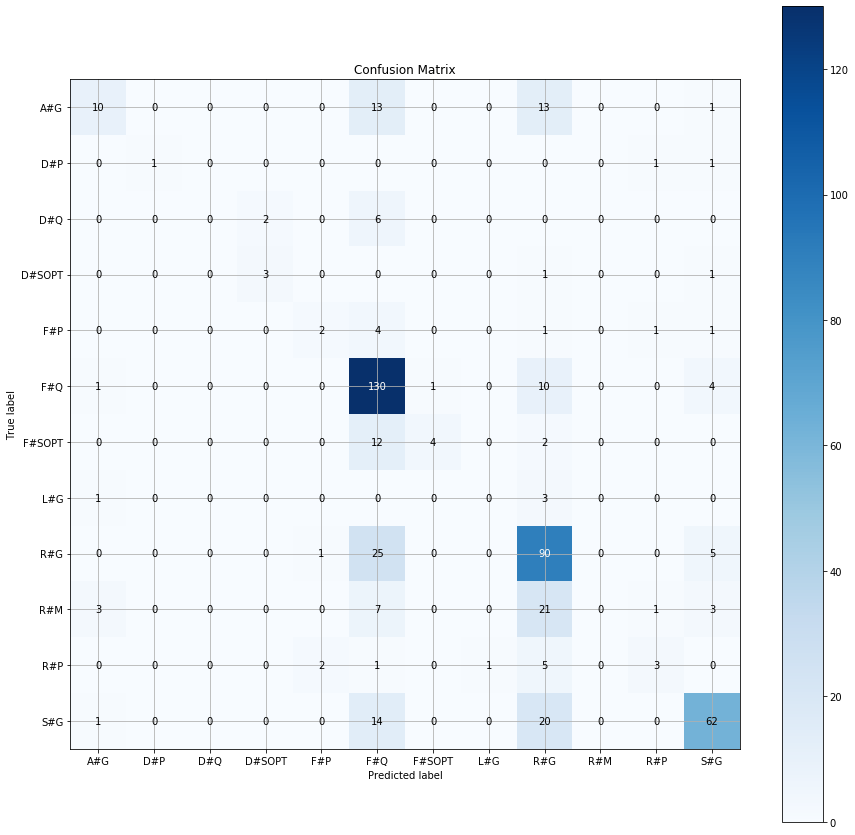
Sample3:



The method of splitting based on sub sentences fails when the subject is missing in the sentence. For example, the above sentence has two aspects FOOD#QUALITY and FOOD#PRICES but as there is no sub sentence detected, it misses the aspect of price.

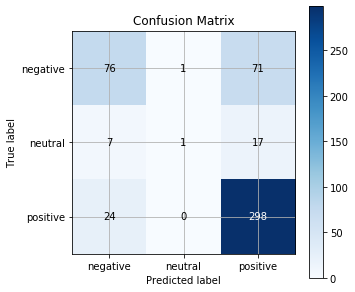
**Evaluation of SVM Model on Parsed Data:**

We observe that the performance of the models have increased by very large amount (27% to 76% accuracy and F-score of 0.16 to F-score of 0.57). Majority of the misclassification cases are because of the skew in the data.



**Fig:** Confusion Matrix for SVM Model for Classification of Aspect Categories on Parsed Data

The skew in the data has affected the classification of polarity more than that of aspect categories. It is important to account for this skew in this data to improve the classification results.



**Fig:** Confusion Matrix for SVM Model for Classification of Polarity Values on Parsed Data

**Future Scope:**

* Improving the method of parsing the sentences to capture all the possible scenarios
* Fine tune the models to account for class imbalance in the data

**References:**

* <https://medium.com/@pmin91/aspect-based-opinion-mining-nlp-with-python-a53eb4752800>
* <https://www.analyticsvidhya.com/blog/2016/08/beginners-guide-to-topic-modeling-in-python/>
* <https://www.machinelearningplus.com/nlp/topic-modeling-gensim-python/>
* <https://remicnrd.github.io/Aspect-based-sentiment-analysis/>
* <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

**Related Datasets**

* <https://www.kaggle.com/snap/amazon-fine-food-reviews>
* <https://inclass.kaggle.com/c/restaurant-reviews>