To Follow Or Buck An Existing Restaurant Trend With Sentiment Analysis

Poojaanjali Vasu

CS648 Written Report

Mathematics and Statistics Department
Maynooth University
Ireland
August 2020

Contents

0.1 Abstract

The developing prevalence of E-business, social medias, discussions, online journals and so on created another stage where anybody can discuss and trade his/her views,ideas , recommendation and experience about any item or services. This pattern collected a large amount of customers produced information on the web. In the event that this content can be extracted and analysed appropriately, at that point it can be an important factor in decision making. Be that as it may, manual extraction and examination of this substance is an impossible undertaking, as the content is not very well structured in nature and it is composed in human language. This situation opened a new area of research called Opinion Mining and Sentiment Analysis. Opinion Mining and Sentiment Analysis is an extension of Data Mining that extracts and analyzes the unstructured data automatically
In this work, we focus on sentiment analysis techniques such as Vader Sentiment Analysis, Hybrid Approach, LSTM, Topic Modelling and Aspect-based sentiment analysis to predict sentiments for each review.

Chapter 1

Introduction

Paul Mangiamele, president and CEO of Bennigan's, says, "Although we all love it, this business is very difficult. It's a wonderful business, a great business, a satisfying business. It's a lucrative business. But there are a thousand moving parts, and you need to be knowledgeable of all of them [**?**]." With that said, a hard reality is that many restaurants fail during their first year, frequently due to a lack of understanding and planning on dine in, location, service, additional facilities. In order to help clients who want to open new restaurants or to bring a change in their existing restaurant to compete in the market, we analyse the customer feedback.

The proliferation of a wide variety of communication media has provided customers with the capability to write and express their experiences about the products and services availed of. Crowdsourcing feedback gives the customers the power to influence prospective customers decision's to avail of the products and services offered. Crowdsourcing applications have gained a lot of attention because it harnesses the potential of a diverse group of people to provide information through various media. Zomato is one of the crowdsourcing applications that gather customer feedback on restaurants [25]. Customers are usually asked to assign a star rating in the range of 1 to 5 to assess the overall experience which is in conjunction with the textual feedback.

This study aims to develop restaurant business model for the businessmen in Ireland/India Locations by analysing sentiments based on several factors like location, ambience, service, quality, food, cuisine, price in reviews to give them insights for making good business decisions. The data used data for this analysis is taken from Zomato API.

Chapter 2

Literature Review

Customer satisfaction is an essential concern in the field of marketing and research. As in the habits of consumers when they get excellent service, they will transmit to others by mouth to mouth [29]. The text mining techniques extract data from documents and produce information that can be used to increase profits and services.

2.1 Sentiment Analysis

Sentiment Analysis, also known as opinion mining, is a powerful tool you can use to build smarter products. It's a natural language processing algorithm that gives you a general idea about the positive, neutral, and negative sentiment of texts as illustrated in figure 2.1 [30].

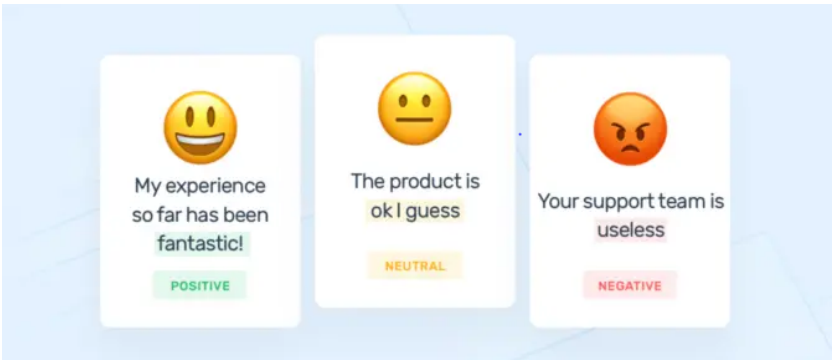

Figure 2.1: Different Sentiments expressed by People [1]

The main types of algorithms used in sentiment analysis are :[1] [32]

1. **Rule-based systems** that perform sentiment analysis based on a set of manually crafted rules.
2. **Automatic systems** that rely on machine learning techniques to learn from data.
3. **Hybrid systems** that combine both rule-based and automatic approaches.
4. Vader Sentiment Analysis [7]
5. Topic Modelling by LDA [13]
6. LSTM - Keras [21], [22] , [23]
7. Ontology driven - Aspect Based Sentiment Analysis(ABSA) [25]

2.2 Rule-based Approaches

Typically, a rule- based system utilizes a lot of human-created rules to recognize subjectivity, polarity, or the subject of a sentiment. These rules may incorporate different methods created in computational linguistics, such as [32]:

1. Stemming, tokenization, part-of-speech tagging and parsing.
2. Lexicons (i.e. lists of words and expressions).

Here's a basic illustration of how a rule-based system works [1]:

1. The two list of sentiment words can be defined in rule-based sentiment analysis(for example: bad, worse, awful are the sets of negative words and better, good and amazing can be considered as the sets of positive words).
2. Counts the number of positive and negative words that appear in a given text.
3. If the number of positive word appearances is greater than the number of negative word appearances, the system returns a positive sentiment, and vice versa. If the numbers are even, the system will return a neutral sentiment [32].

Rule-based frameworks are exceptionally gullible since they don't consider how words are joined in a sequence. Obviously, further developed processing procedures can be utilized, and new rules can be updated to help new vocabulary. Be that as it may, including new rules may influence past outcomes, and the entire system can get extremely complicated.

2.3 Automatic Approaches

On the other hand to rule based system, Automatic Approach don't depend on manually created rules and uses machine learning algorithms. The classification model takes text as input and returns the respective sentiments as output such as positive, negative and neutral. [32].
Here's how a machine learning classifier can be implemented as shown in figure 2.2 :

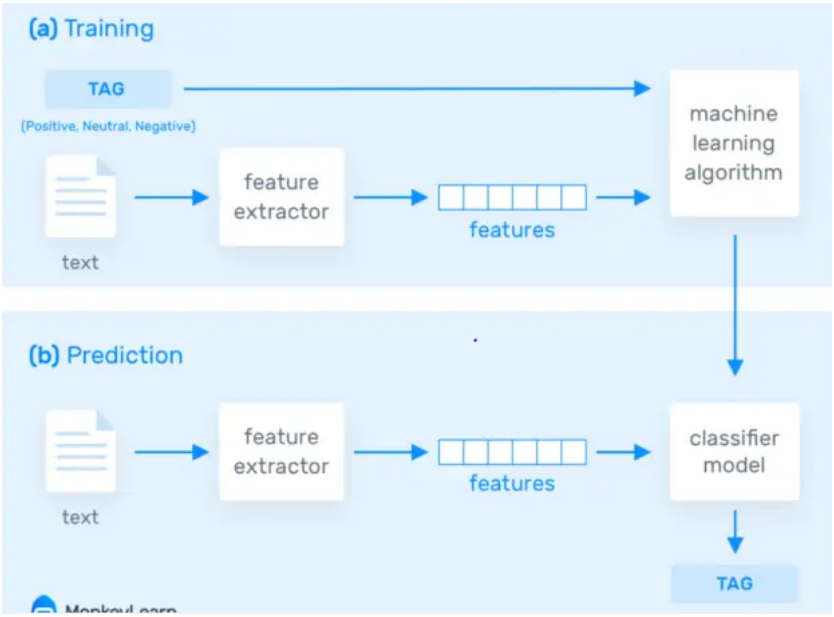

Figure 2.2: Machine Learning Classfier [1]

2.3.1 The Training and Prediction Processes

In the training process of figure 2.2, our model learns to associate a particular input (i.e. a text) to the corresponding output (tag) based on the

samples used for training. The feature extractor transfers the text input into a feature vector which will be detailed in 2.1.3.2 subsection. Pairs of feature vectors and tags (e.g. positive, negative, or neutral) are fed into the machine learning algorithm to generate a model.
In the prediction process of figure 2.2, the feature extractor is used to transform unseen text inputs into feature vectors. These feature vectors are then fed into the model, which generates predicted tags (positive, negative, or neutral).

2.3.2 Feature Extraction from Text

Machine learning algorithms cannot work with raw text directly, the text must be converted into a vector of numbers. This is called feature extraction.The classical approach for feature extraction has been bag-of-words. [9].

2.3.2.1 BOW (Bag-Of-Words)

It is called a "bag" of words, because any information about the order or structure of words in the document is discarded. The model is only concerned with whether known words occur in the document, not where in the document.
A bag-of-words is a representation of text that describes the occurrence of words within a document. It involves two things [9] -

1. A vocabulary of known words.
2. A measure of the presence of known words.

The below figure 2.3 presents the BoW technique with two document example, listing also the relevant stop words found in the two texts-

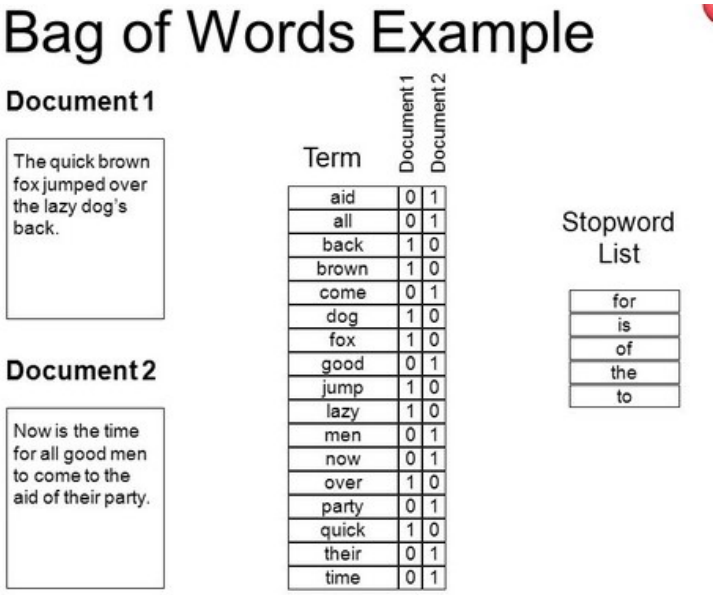

Figure 2.3: Bag Of Words Example [26]

2.3.3 Classification Algorithms

The classification step usually involves the statistical models below [3]:

1. **Naïve Bayes:** A family of probabilistic algorithms that uses Bayes's Theorem to predict the category of a text.

2. **Support Vector Machines:** A non-probabilistic model which uses a representation of text examples as points in a multidimensional space. Examples of different categories (sentiments) are mapped to distinct regions within that space. Then, new texts are assigned a category based on similarities with existing texts and the regions they're mapped to.

3. **Multi-Layer Perceptron:** MLP is a neural network in which data flows in one direction i.e., from input layer to output layer with one or more layers between input and output.

4. **Maximum Entropy:** The principle behind this algorithm is to find from the prior test data the best probability distribution.

The advantages and disadvantages of the above models are specified in the table 2.1 [3].

Sr. No.	Technique	Remarks	Advantage	Disadvantage
1	Naïve Bayes	It is implemented to calculate the probability of a data to be negative or positive.	1. Model is easy to interpret. 2. Fast and efficient computation. 3. Not affected by irrelevant features	1. Assumes independent attributes
2	Support Vector Machine (SVM)	It is implemented to develop a hyper plane in order to separate the data points of two classes from one another.	1. Very good performance 2. Data set dimensionality has low dependency. 3. Produces accurate and robust classifications	1. Lack of transparent of results. 2. Difficult interpretation of resulting model.
3	Multi Layer Perceptron	MLP is a neural network in which data flows in one direction i.e., from input layer to output layer with one or more layers between input and output.	1. Most used type of neural network 2. Capable of learning almost any relationship between input and output variable.	1. Requires more time for execution. 2. Flexibility depends on enough training data need. 3. It is somewhat considered as complex 'black box"
4	Maximum Entropy	The principle behind this algorithm is to find from the prior test data, the best probability distribution.	1. Provides proper distribution. 2. Do not assume statistical independence of random variables.	1. Requires more of the human efforts in the form of additional resource or annotations. 2. Cannot model the data that require p(alb) = 1 or 0

Table 2.1: Advantages and Disadvantages of Statistical Models

2.4 Hybrid Approaches

Hybrid systems combine the desirable elements of rule-based and automatic techniques into one system. One huge benefit of these systems is that results are often more accurate. The basic goal of this combination is to yield the best and optimum results using the effective feature set of both rule and auto approaches, and to overcome their deficiencies and limitations [23]. The Following steps were carried out by many researcher's for the implementation of the hybrid technique: [4]

1. The features to be used by the hybrid approach are identified and separated.

2. The annotated corpus to be used for training and validation of the best classifier at different corpus sizes is built by the system.

3. A Sentiment lexicon of different sizes is built using the annotated cor-

pus.

4. These different approaches are combined and tested for better and optimized results.

5. A Straight forward and simple method is crafted to detect negations in the hybrid approach.

2.5 Vader Sentiment Analysis

Sentiment analysis analyzes people's opinions, sentiments, evaluations, attitudes, and emotions via the computational treatment of subjectivity in text. Manually creating and validating such lists of opinion-bearing features is one of the most robust methods for generating reliable sentiment lexicons and is also one of the most time-consuming [7].
So, we use a simple rule-based, lexicon model for sentiment analysis called Vader (Valence Aware Dictionary and sEntiment Reasoner) to eliminate the manual work. This is geared specifically to emotions shared in social media. VADER uses a variation of a sentiment lexicon, which is a collection of lexical characteristics (e.g, words) usually classified as either positive or negative according to their semantic orientation.

2.6 Topic Modelling

Topic modeling is another hot topic in the text mining field, used for uncovering hidden structure in a collection of texts [13] . The topic modeling is a dimensionality reduction technique, where rather than representing a text T in its feature space (Word_i: count(Word_i, T)), you can represent it in a topic space (Topic_i: Weight(Topic_i, T)) [17].
By doing topic modeling, we build clusters of words rather than clusters of texts. A text is thus a mixture of all the topics, each having a specific weight.

2.7 Sentiment Analyis using LSTM

One of an important type of Recurrent Neural Network is Long Short-Term Memory (LSTM). This architecture was constructed on simple RNNs. The temporal sequence and its long range dependecies were modeled more accurately using LSTM. [21], [22] , [23].
The temporal state of the network is stored in the memory blocks of LSTM,

which is in the recurrent hidden layer. The gate are multiplicative units, which are used for information flow control. The output gate and input gate is contained by every memory block. The flowing of input activations inside the memory cell is controlled by the input gate whereas the flowing of cell activation to the other parts of the network is controlled by the output gate. The memory block consists of the forget gate which helps in scaling of internal conditions of the cell before feeding as input to the cell by the method of self-recurrent connection. In this way, it adapts to forged and reset the memory of the cell as required. [21], [22] , [23].

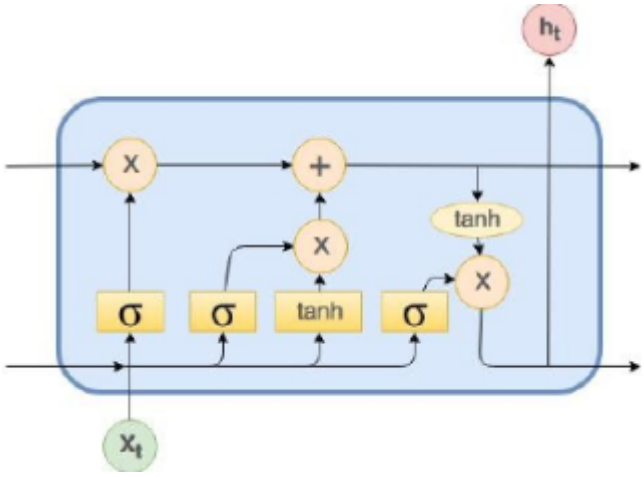

Figure 2.4: Single Cell - LSTM

An LSTM network computes a mapping from an input sequence $x = (x1, ..., x\ T)$ to an output sequence $h = (h1, ...,hT\ )$ by calculating the network unit activations using the following equations [7] iteratively from $t = 1$ to T :

$$f_t = \sigma(W_f . [h_{t-1}, x_t] + b_t) \quad (1)$$

$$i_t = \sigma(W_i . [h_{t-1}, x_t] + b_i) \quad (2)$$

$$\tilde{C}_t = \tanh(W_c . [h_{t-1}, x_t] + b_c) \quad (3)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

$$o_t = \sigma(W_o . [h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

Figure 2.5: Network Unit Activation Equations

Where *W denotes the weight matrices, Ct is the cell state and b is the*

input bias vector. And the i, f, o are the input, forget and the output gate layer. Cell out activation function is *tanh*.The sentiment is predicted using the activation function by the last layer of the network which is the output layer. Here, we use softmax activation function. [21], [22] , [23].

2.8 Aspect based Sentiment Analysis(ABSA)

We need to check sentiment not on entire reviews but on several aspects in each review like ambience, service, food, restaurant, location, drinks, price. By doing this the business can be improved on these particular aspects by taking better decisions.
For implementing the above scenario we do aspect based sentiment analysis. This is a text analysis technique that breaks down a text into aspects (attributes or components of a product or service), and then allocates each one a sentiment level (positive, negative or neutral).

Chapter 3

Data Preparation

Data preparation is typically an iterative process of manipulating raw data, which is often unstructured and messy, into a more structured and useful form that is ready for further analysis. The whole preparation process consists of a series of major activities including data profiling, cleansing, integration, and transformation that uses the dataset made available through ZOMATO API. [5].
We are going to see the data preparation for each of the techniques used in this work -

1. Vader Sentiment Analysis
2. Hybrid Approach
3. Topic Modelling by LDA
4. LSTM - Keras
5. Ontology driven - Aspect Based Sentiment Analysis(ABSA)

3.1 For Vader, Hybrid Approach, Topic Modelling -

The analysis presented in this paper uses the restaurant dataset that is collected using the Zomato API as explained in chapter 2. Here I have taken the restaurant details in json format in locations of Ireland and India through the Zomato API. Once the places in Ireland/India are entered as input, their location details like latitude, longitude, entity id and type is received and stored using the python code. Then people can choose the cuisine they want to try in the entered place and all the restaurants relating to that cuisine

type are displayed on the screen based on it's rating from top to bottom as displayed in figure 4.1.

Figure 3.1: Top-10 Restaurants Listed On A Rating Basis

The bar plot in figure 4.2 shows the restaurants ordered in descending order by rating in Dublin location.

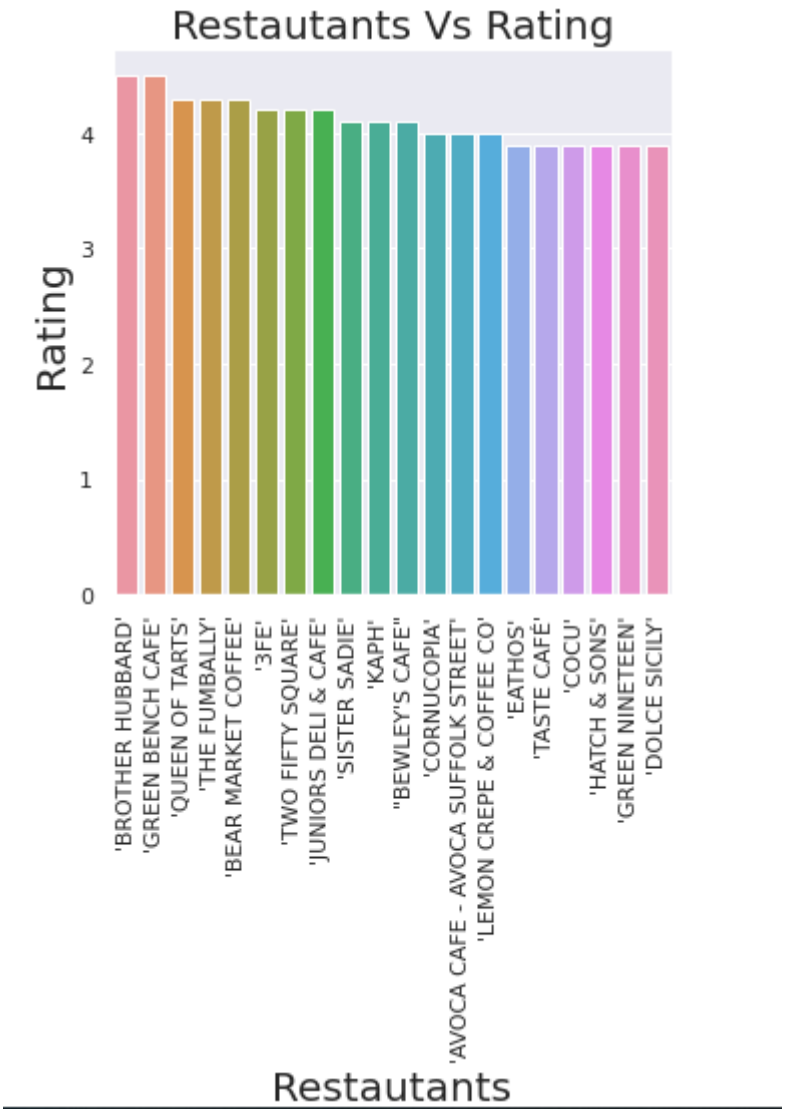

Figure 3.2: Bar Graph Visualisation for Restaurants - Rating wise

The resulting restaurant reviews are collected for the years 2020 and 2019 and stored in two separate files for further analysis.

3.2 LSTM

We collected on extra and dynamic dataset of restaurants located in India/Ireland than in section 4.1 using Zomato API which includes id, name,

type, cuisines, average_cost_for_two, location, city, Has_online_delivery, rating, Has_table_booking, votes as columns for analysing several factors, thus to improve the restaurant business by taking right decision.

3.3 Aspect based Sentiment Analysis(ABSA)

For evaluating the aspects in reviews such as Service, Ambience, Price, Food, we use a different dataset (i/e) of restaurant reviews from SemEval2016 as training data [**?**]. The training data contains the sentence id, review, aspect category, aspect sentiment develops and trains our machine learning algorithm on several aspects and its sentiment.

	sid	review	category	polarity
1	1004293:0	Judging from previous posts this used to be a ...	RESTAURANT#GENERAL	negative
2	1004293:1	We, there were four of us, arrived at noon - t...	SERVICE#GENERAL	negative
3	1004293:2	They never brought us complimentary noodles, i...	FOOD#QUALITY	negative
4	1004293:3	The food was lousy - too sweet or too salty an...	FOOD#STYLE_OPTIONS	negative
5	1004293:4	After all that, they complained to me about th...	nan	nan

Figure 3.3: Different Aspects in Reviews

The test dataset consists of 500 reviews, collected for the year 2020 using Zomato API. It is used to evaluate the performance of our algorithm to find out the aspects and the corresponding sentiment .

Chapter 4

VADER Sentiment Analysis

As explained in Section 2.5, VADER outperforms individual human raters, and generalizes more favorably across contexts than any of our benchmarks [7].The biggest advantage is it doesn't require any training data but it's constructed from a generalizable, valence-based, human-curated gold standard sentiment lexicon attuned to microblog-like contexts.The figure 4.1 provides an overview of the research process and summarizes the methods used in this study.

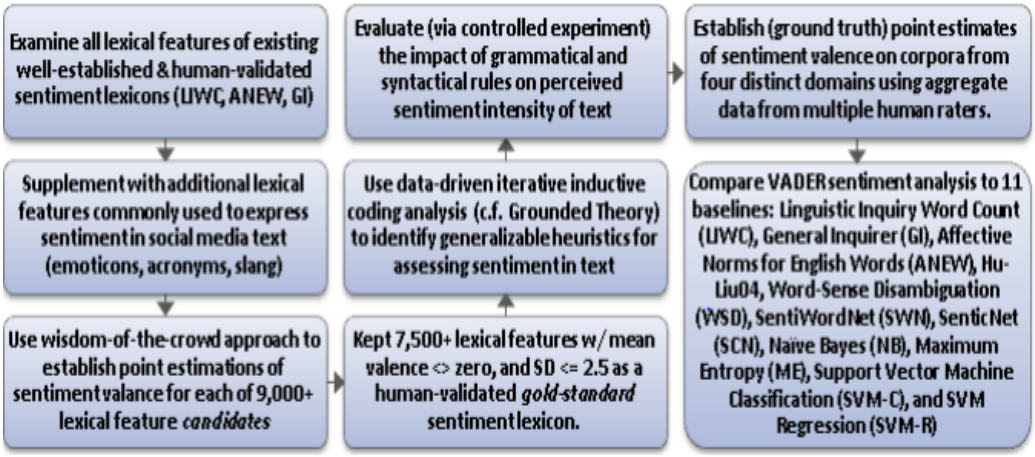

Figure 4.1: Methods in Vader Sentiment Analysis [7]

VADER next uses lexical features with consideration for five generalizable rules mentioned below [8] -

1. Conjunctions: There can be a shift in polarity by the use of conjunc-

tions such as "but" in the text.

2. Punctuation: The magnitude of the intensity of sentiment orientation is increased by the use of exclamation mark(!).

3. Capitalization: The magnitude of the sentiment intensity is increased, when uppercase letters are emphasized.

4. Preceding Trigrams: The polarity of the text gets flipped using a preceding trigram that catches almost 90% of the negation.

5. Degree modifiers: Degree modifier generally increases or decreases the sentiment intensity.

The dataset contains restaurant reviews from the time period 2019- 2020. The VaderSentiment library in python uses the function polarity_scores(sentence) to calculate the polarity indices(i.e) positive/negative/neutral scores for each sentence. The Positive/ Negative/ Neutral scores represent the sentiment proportion of each review and a compound score is calculated based on these categories. This means, for example if our sentence was rated as 0.56 Positive, 0.44 Neutral and 0.00 Negative and the compound score is greater than 0.5 , then the review is classified as positive . If the compound score is lesser than 0.5 , then the review is classified as negative or else neutral. By this method, all the reviews are classified as shown in figure 5.1 as either positive(1) /negative(-1)/neutral(0) sentiment for both the years and stored in the Google Colab as datasets using python code.

```
#Baseline sentence
sentiment_analyzer_scores('The service here is good')
```

```
The service here is good---------------- {'neg': 0.0, 'neu': 0.58, 'pos': 0.42,
'compound': 0.4404}
```

```
#Degree Modifiers
print(sentiment_analyzer_scores('The service here is extremely good'))
print(sentiment_analyzer_scores('The service here is marginally good'))
```

```
The service here is extremely good------ {'neg': 0.0, 'neu': 0.61, 'pos': 0.39,
'compound': 0.4927}
None
The service here is marginally good----- {'neg': 0.0, 'neu': 0.657, 'pos': 0.34
3, 'compound': 0.3832}
None
```

Figure 4.2: Predicting Sentiment by Vader Sentiment Technique

We find that incorporating these heuristics in figure 4.1, gives us a very remarkable and encouraging results. Thus we can incorporate the VADER

technique for several domain contexts (social media text, NY Times editorials, movie reviews, and product reviews). We can visualise the summary of positive/negative/neutral sentiment reviews for the past 2 years that was classified using Vader Sentiment Analysis in the pie charts given in figures 4.2 and 4.3.

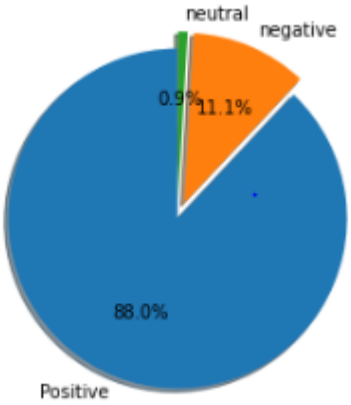

Figure 4.3: Proportion of Sentiment Reviews as Positive/Negative/Neutral for year 2020

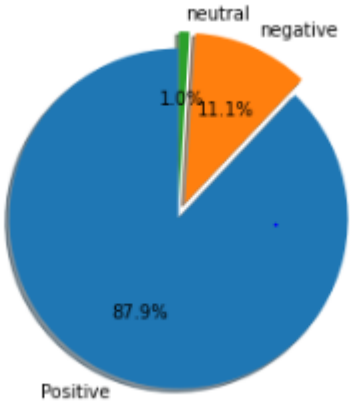

Figure 4.4: Proportion of Sentiment Reviews as Positive/Negative/Neutral for year 2019

Chapter 5

Hybrid Approach

As explained previously in section 2.4 about the Hybrid Approach, in this study We implement a rule based approach using list of words (unigrams) to calculate the positive/negative sentiment/tag of each review. Further, implement automatic approach of feeding the reviews into the classifier after data pre-processing techniques.

5.1 Cleaning

The reviews are converted to lowercase and punctuations are removed. The stop words and the lemmatization technique is performed on each word in the sentences. The lemmatization is a technique where each word is converted to its base word(eg: going -¿ go , goes -¿ go) . All the processed sentences are stored in a corpus model.

5.2 Feature Extraction

As already explained in section 2.3.2, feature extraction is performed using Bag-Of-Words model.

5.3 Classifier

In this automatic approach, the feature vectors obtained from BOW model and the sentiments calculated using the positive/negative unigrams are fed to the classifier as input to train the Model. We split the dataset into training

and test data with proportions of 80:20 respectively. The classifiers used are Naïve Bayes, Random Forest, SVM and KNN. Thus the training dataset is scaled and fit to each of the classification models and the sentiments for the test dataset is predicted. The Naïve Bayes gives an accuracy of 91.4% , SVM gives accuracy upto 90.7%, and KNN gives the least accuracy of 88.2% and the Random Forest classification model stands out here by giving the highest accuracy result of 94.8%.
A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known and compares it with predicted values.[33] The figure 5.1 displays the confusion matrix for the positive/negative/neutral labels of the test dataset for the random forest classification model.

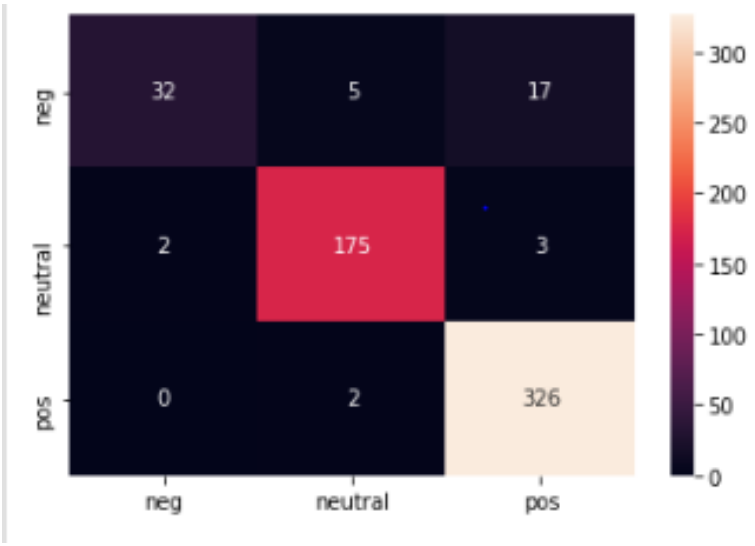

Figure 5.1: Confusion Matrix of Random Forest Classifer

Chapter 6

Topic Modelling- LDA

As explained in section 2.6, there are several existing algorithms you can use to perform the topic modeling. One common method of unsupervised learning is (Latent Dirichlet Allocation) LDA to discover hidden topics in documents. It assumes that there are latent variables that reflect the thematic structure of the documents.

6.1 Parameters

LDA treats the probability distribution of each document over topics as a *K-parameter* hidden random variable (K is the number of hidden topics). The *Alpha parameter* is the Dirichlet prior concentration parameter that represents the document-topic density — with a higher alpha, documents are assumed to be made up of more topics and result in more specific topic distributions per document. The *Beta parameter* is the same prior concentration parameter that represents the topic-word density — with high beta, topics are assumed to be made up of more words and results in a more specific word distribution per topic [17].

6.2 Data Preparation

The reviews from the time period 2019-2020 are converted to lowercase and punctuations were removed. The stop words and the lemmatization technique is performed on each word in the sentences. The lemmatization is a technique where each word is converted to its base word(eg: going *will be converted to* go , goes *will be converted to* go) . All the processed sentences are stored in a corpus model for both the years.

6.3 Exploratory Data Analysis

All the similar words (i.e) n-grams in the reviews are clustered together by the K-Means Clustering Algorithm with the Word2Vec model [18].

6.3.1 Word2Vec

A vector space is created by Word2vec after taking huge collection of text, normally with a size of some hundred where each distinct word in the collection is allocated to the respective vector space. The words which share relevant contexts are stored nearby to each other in vector space in word vectors..

6.3.2 K Means Clustering

K-Means is one of the simplest and most popular machine learning algorithms out there. It is a unsupervised algorithm as it doesn't use labelled data, in our case it means that no single text belongs to a class or group. It is also a clustering algorithm that classifies a dataset into a K number of clusters [27].Here, We use nltk.cluster.util.cosine_distance(u, v) function in KMeans Algorithm to assign clusters based on similarity for the words obtained from the Word2Vec model [18]. The figure 6.1 shows there are 4(i.e) 0,1,2,3 labelled clusters and the cluster '0' has the highest count of similar words and the cluster '3' has the lowest count of similar words.

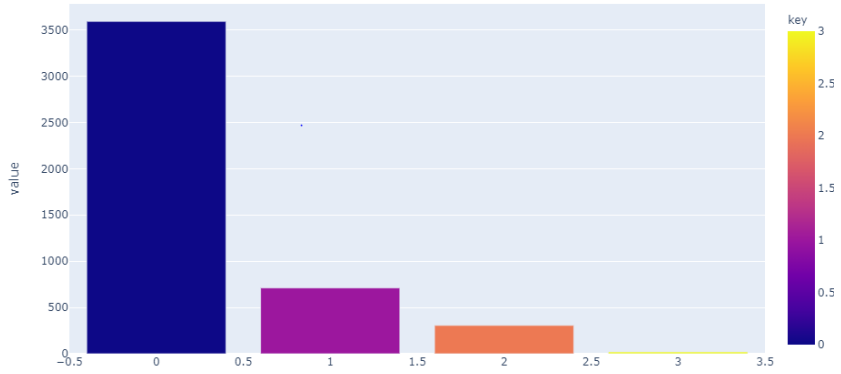

Figure 6.1: Clustering Based On Similarity of Words using Word2Vec model

Figure 6.2: Word Cloud Visualisation of Unigrams in Reviews

The figure 6.2 shows a visual representation of most common words used in reviews using word cloud package.
We also observe that the most frequently used tags in the reviews were nouns, adverbs in figure 6.3 and the commonly used nouns are food, service and adverbs are good, taste in our reviews from figure 6.2.

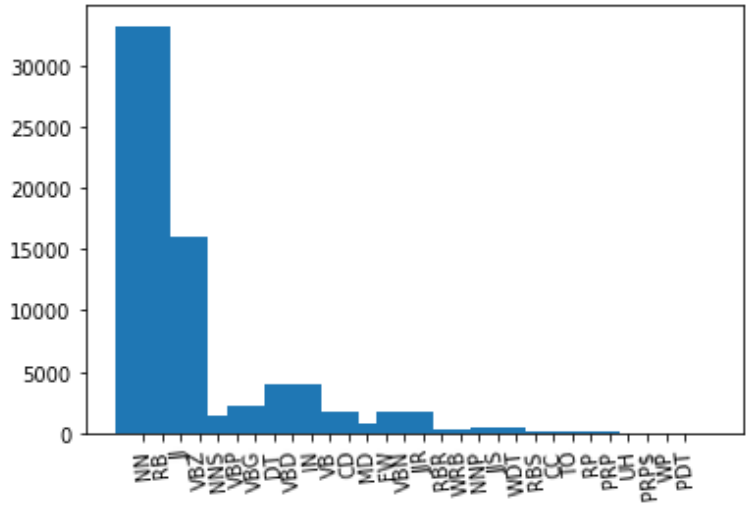

Figure 6.3: Commonly Used Terms in Reviews

6.4 Training model Using LDA

The LDA model accepts input as Bag-Of-Words [16]. For this,We apply the doc2bow function on reviews. The doc2bow function is responsible for calculating the number of occurring each unique word, converting the word to its numeric word id and gives sparse vector as output [28]. The Gensim's LDA model, is a Python library used for topic modelling and document indexing in large corpus models. It requires the following parameters to extract the hidden structure within documents.

6.4.1 Parameters

1. **input -** input - The stream of document vectors (corpus).
2. **num_topics** – The count of on-demand latent topics which have to be fragmented from the collection (corpus).
3. **id2word** – Maps word IDs to words.
4. **chunksize** – The count of documents that will be using in each chunk of training.
5. **passes** – The count which passes through the collection during training.

6.5 Analyzing LDA model results

The pyLDAvis visualisation answers in the figure 6.4 to the following questions on the fitted topic model: (1) what does each topic explain? (2) What is the prevalence of each topic? and (3) How the topics are inter-related?
There are two main parts in our visualization. The left side represents the topic model in global view. The topics are plotted as circles in two dimensional plane and the centers are computed using the distance between the topics. The projection of inter topic distances onto two dimensions is performed by multidimensional scaling. The topics are sorted in the decreasing order of prevalence by using the area of the circles.
The topic currently selected on the left side depicts the most useful individual terms for interpretation, which are the horizontal bars of the right panel. This answers to the question 1. The topic-specific and corpus-wide frequencies are represented by a pair of overlaid bars.
The topic selected (on the left) reveals the most appropriate terms (on the right) for illustrating the selected topic. Further, the conditional distribution over topics (on the left) is revealed by selecting a term (on the right).Thus, it helps us to analyse a huge number of topic-term dependencies. [14] [15].

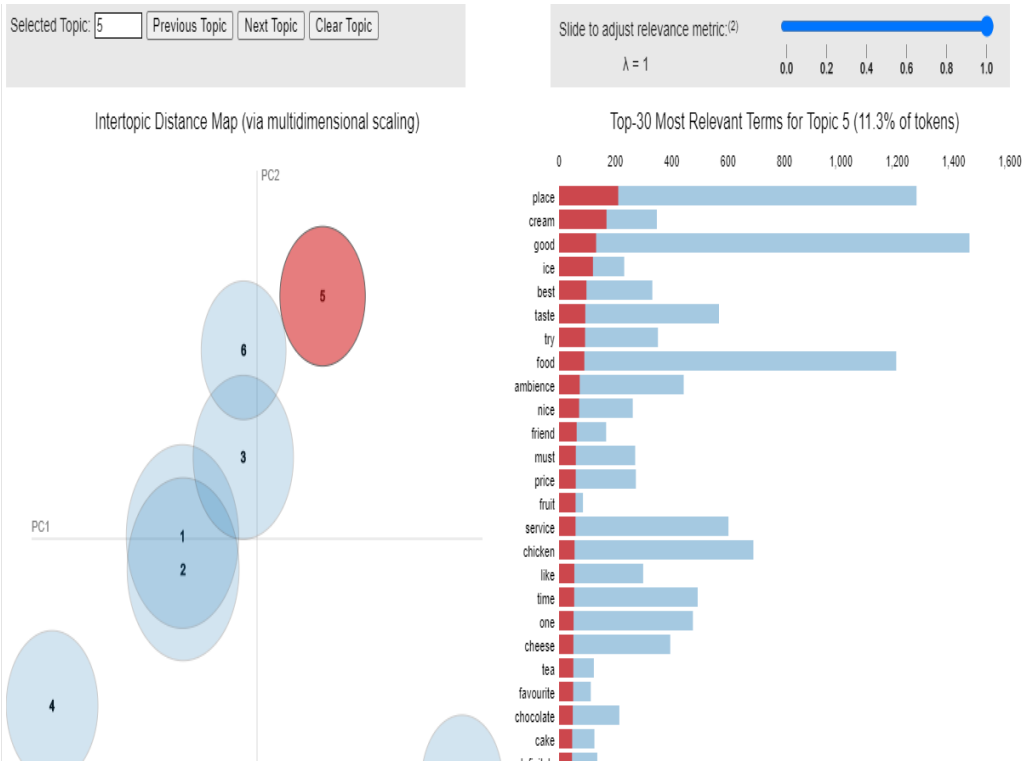

Figure 6.4: PyLDAvis Visualisation for Topic Modelling

6.6 Predicting Sentiment Using Hidden Topic Distribution

The idea here is to test **whether the topic distribution per review/hidden topic structure could predict positive/negative/neutral sentiment** as illustrated in figure 7.5 [19].

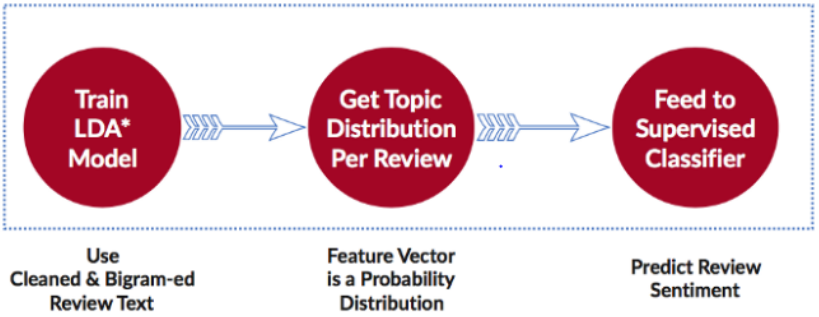

Figure 6.5: LDA Model Decomposition [19]

6.6.1 Steps to Perform

1. Train an LDA Model on 3000 Restaurant Reviews for the year 2020 year.
2. LDA model should be used for grabbing Topic Distributions of all reviews.
3. In supervised classification models, Topic Distributions should be used straightly as feature vectors and accuracy should be checked.
4. Use the same 2020 LDA model to get topic distributions for the year 2019 (the LDA model did not see this data!)
5. Run supervised classification models again on the 2019 feature vectors and check the accuracy to see if the structure generalizes.

The above steps are implemented by performing the below processes -

6.6.2 Train LDA Model

A data frame is created with the 2020 year reviews and stopwords, punctuations are removed. A Gensim Dictionary with a list of unigrams and bigrams is built using the bigram_mod function. A mapping between words and the integer ids are created for the bigrams and unigrams using id2word function. A corpus model is also built on the preprocessed reviews using the doc2bow function and returns a sparse vector. With these as inputs the LDA model is trained.

6.6.3 Grab Topic Distributions

We're going to use the trained LDA model to identify the distribution of the 20 hidden topics/structure using *lda_train.get_document_topics* function in python for each review. These are the feature vectors used as an input for the supervised classification models to determine the output(positive/negative/neutral).

6.6.4 Applying the Model on Unseen Data

The classification models such as Random Forest, SVM are trained with feature vectors and the sentiments mentioned in 6.6.3. Here the test dataset is 2019 reviews. Then, grab the test feature vectors from 2019 reviews using Trained LDA Model of 2020 year. Then, run the classification models on the test feature vectors and predict the sentiment. It was observed that Random Forest gives the highest accuracy of 94% and SVM with 87%.

Since the supervised models on the unseen data(2019) performed well, we can tell that the **2020 topic model has identified the latent semantic structure** that persists over time in this restaurant review domain.

Chapter 7

Analysis using LSTM-Keras

Until now we have considered only the words used in the reviews and not in the order which they occur. To consider the order of occurance of words in sentiment analysis, we have used Deep LSTM RNNs for our study to explore deeper semantics of words by learning the long distance contextual dependency among them for better classification.
Apart from the sentiment and the structure of the reviews, we will analyse several factors in detail that helps the entrepreneurs with restaurant business. [24]-

In this work, we choose to analyse the factors for the location of Bangalore in India [11]. So we collect data through Zomato API for this location and store it as a dataset. The dataset contains features such as id, name, type, cuisines,Average_cost_for_two, location, city, Has_online_delivery, has_table_booking, rating, votes, reviews to provide insights to entrepreneurs.

7.1 Exploratory Data Analysis

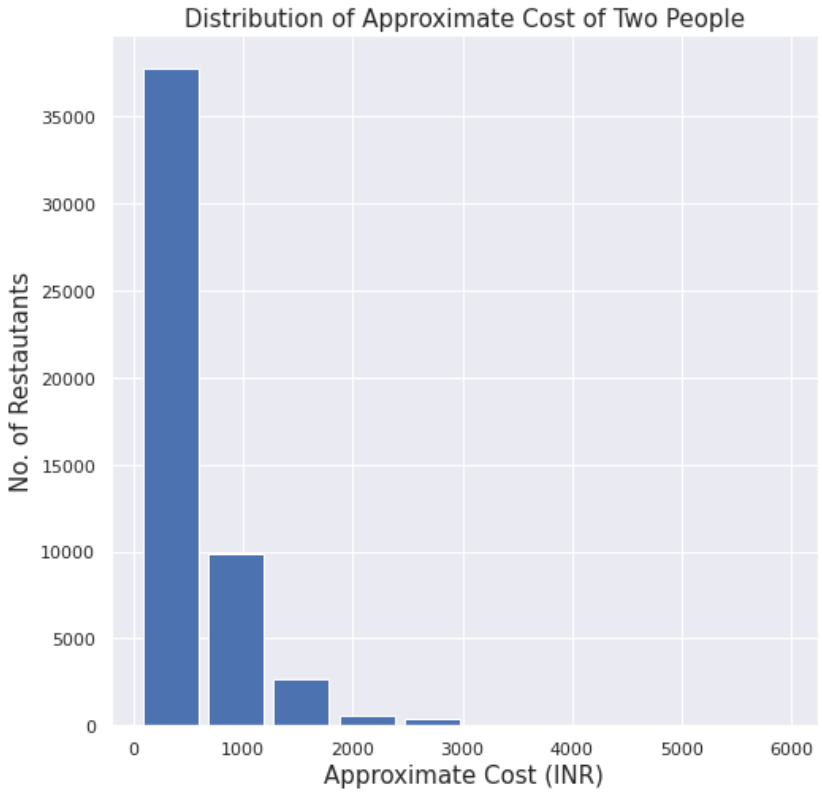

Figure 7.1: Approximate cost estimation at restaurants [12]

The above figure 7.1 indicates that the maximum number of restaurants serve food for an approximate cost of 1000 Rupees for two people, thus giving the entrepreneur an insight about setting the price. Is this information enough to decide on the cost ? No, We need to dig deeper and check the cost for each type of restaurant to make a strong decision.

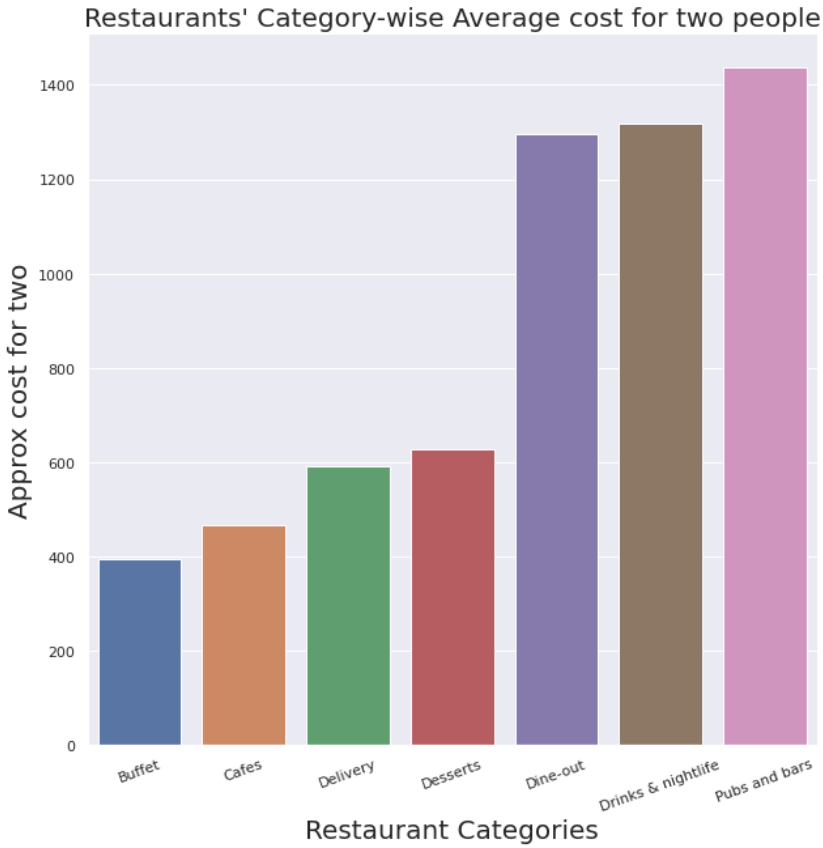

Figure 7.2: Approximate cost estimation for each categories [31]

We observe from figure 7.2 that the Buffets are having a low cost of around 400 Rupees, whereas Pubs and bars are having higher cost around 1500 Rupees. This may be due to the work life of people, bangaloreans interest and so on. Thus the above graph helps in establishing the cost for each type of restaurant.
Now we have analysed the cost factor and the type of restaurant, but where should we open the restaurant?? To get an answer to this question, lets us analyse the number of restaurants in each location of Bangalore.
The bar plot in figure 7.3 shows that the maximum number of restaurants are in the BTM location, because it's a highly populated and a frequently visited area. Thus, it offers an insight about starting their business in a dense/sparse area by visualisation of restaurant density at each location in bangalore .

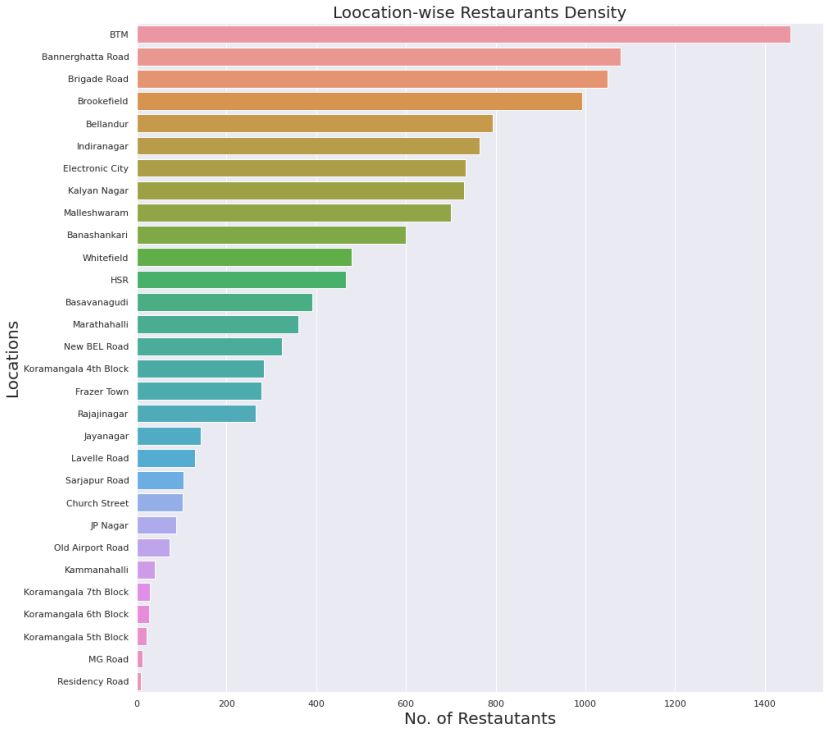

Figure 7.3: Density Estimation of Restaurants [31]

Our world has become so digitised, that everything arrives at our door stop and hence we should further talk about several facilities about restaurant. So let us analyse the proportion of restaurants that have facilities such as - online order, table booking, both/none of the options.

From the figure 7.4 below, we can observe that there are plenty of restaurants at each location with the online ordering facilities and also only a handful of restaurants have enabled the table booking options. A few number of restaurants have both the options enabled and we cannot ignore the fact that none of the facilities are available for quite a large percentage of restaurants at every location. From this analysis, we can understand that many restaurants have enabled online ordering for saving cost in infrastructure.

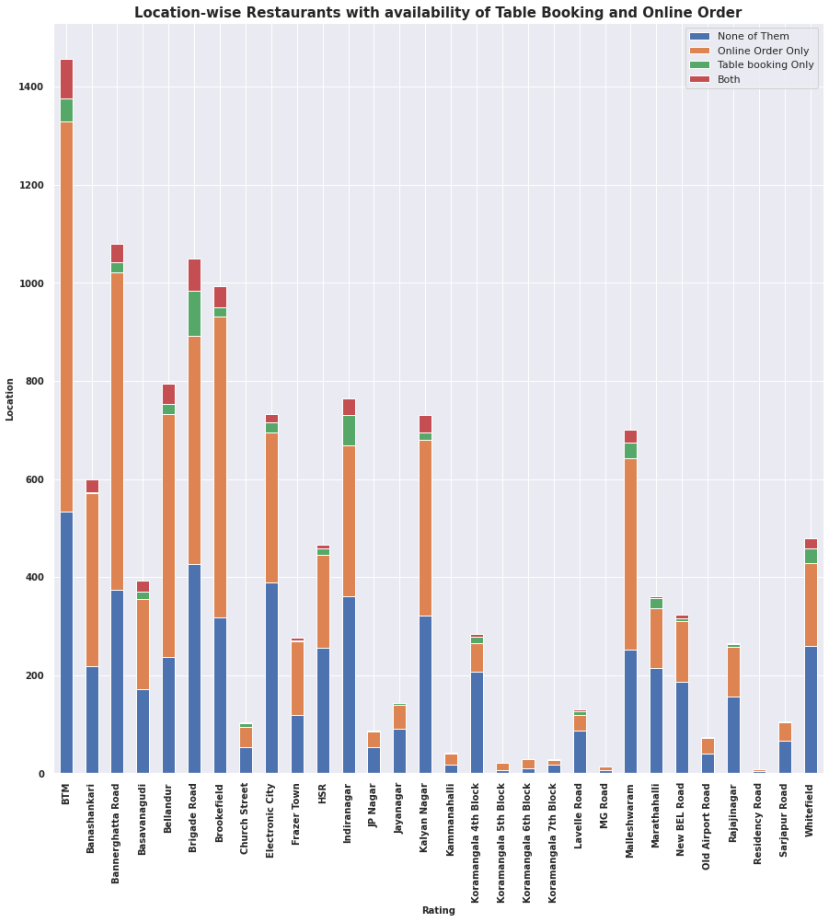

Figure 7.4: Online/Table Booking Facilities

The figure 7.5 shows the variation of restaurant densities on a map. For this purpose, we use the geocoding concept for converting addresses into sets of latitude and longitude and vice-versa. We use a folium library that depicts each type of restaurant with a unique colour in the map.

Figure 7.5: Variation of Restaurant Densities On A Map [31]

We incorporate a heatmap technique as seen in figure 7.6 to show the densities of restaurants at several locations as variations of color in two dimensions. This variation of color occurs because of the intensity of clusters.

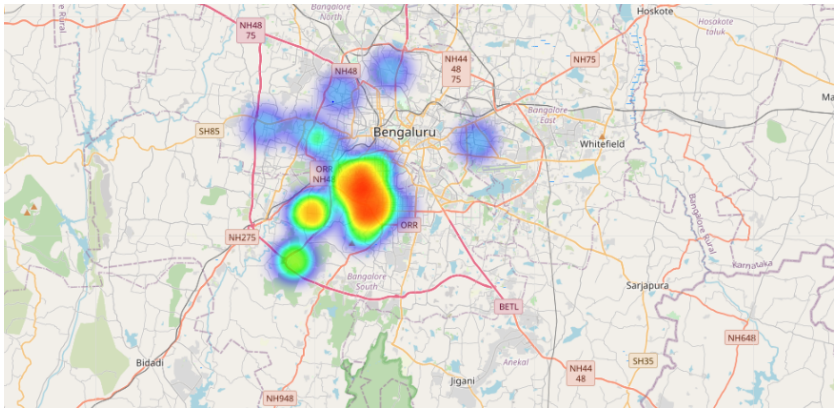

Figure 7.6: Heat Map Technique [21]

We have analysed several factors like cost and facilities, but is the rating by the customers dependent on these factors?? We plot a correlation matrix to understand how each factor is correlated with another.

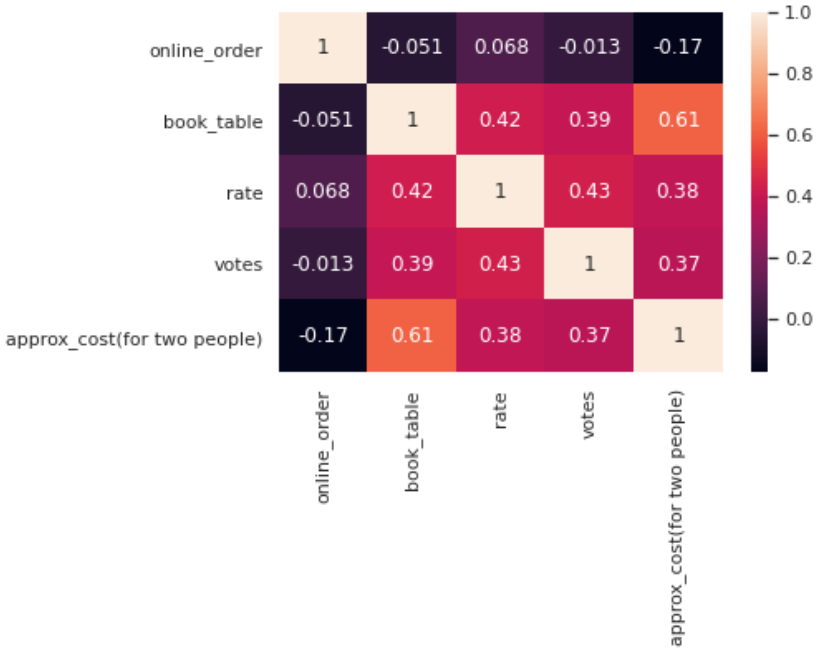

Figure 7.7: Coorelation Matrix to check dependency between factors

The figure 7.7 shows there is a high positive correlation between booking a table and cost indicating that the cost of food will increase if you have a

table booking at the restaurant at a table. Also there is a negative correlation between online ordering facility and the cost, indicating that the cost of food will be the least when you order it online.

We introduce the word cloud visualisation in figure, where the size of a word shows how often it appears in a text.

Figure 7.8: word cloud visualisation for popular dish [31]

We can observe from figure 7.8 that Chicken Biryani, Mutton Biryani, Ice cream, Butter Chicken, Peri Peri, Masala Dosa seems to be the favourite foods for the bangaloreans. What about the cuisines?

Figure 7.9: word cloud visualisation for popular cuisines [31]

We can observe that the North Indian , South Indian, and Chinese seems

to top the list from figure 7.9. So if an entrepreneur includes North Indian and South Indian food varieties in their restaurants, they can attract more customers. Also not to ignore the ice creams(desserts) which was liked by most of the Bangaloreans.
Let us also see what our customers commonly speak about in our reviews dataset. The unigrams and bigrams plots in figures 7.10 and 7.11 will give us an insight on these. The customers mostly talk about food, good, place, chicken, and service in our reviews is observed from these figures.

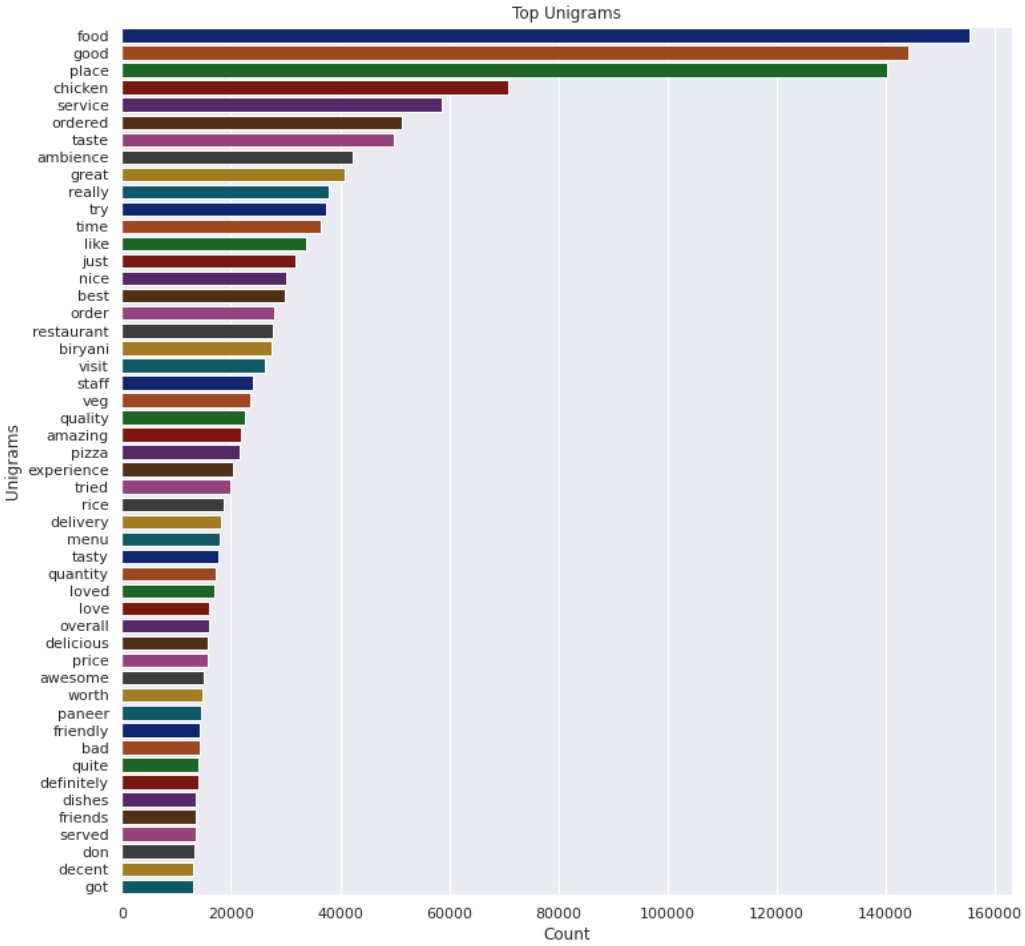

Figure 7.10: Unigrams in Reviews

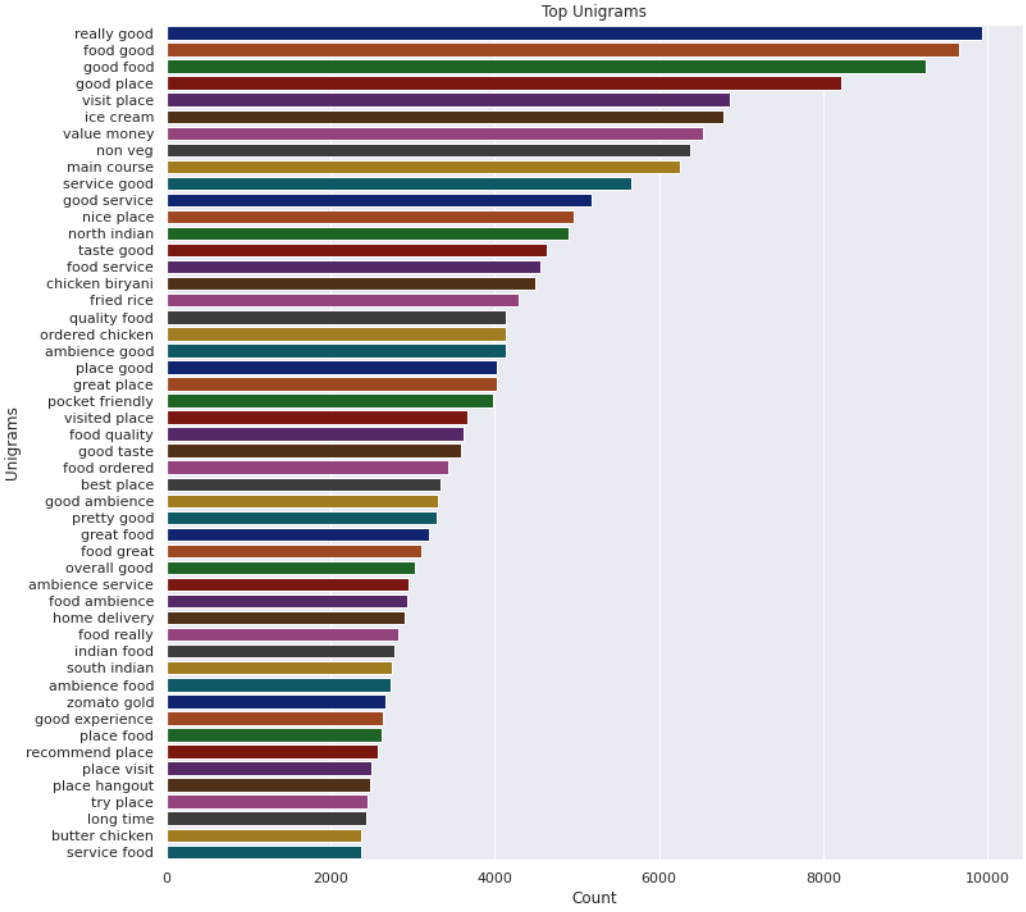

Figure 7.11: Bigrams in Reviews

7.2 Sentiment Classification by LSTM

7.2.1 Architecture Summary

Stacking many LSTM layers is the building block for Deep LSTM. As it known that LSTM has deep architecture. Besides the propagation through time, the input to the network passes through many LSTM layers at a given time frame. Thus, the deep LSTM are better since they distribute the parameters over the space through many layers..[21], [22] , [23].
The embedding layer will be fed with words. The cells in LSTM will be fed with the latest representations from the output of the embedding layer. As the network is added with recurrent connections, the details for order of words can be added in the data. Gradually, the output layer of sigmoid is fed

with the LSTM cells. The sigmoid is used for predicting textual sentiment of positive or negative. The activation function of sigmoid forms the individual output layer as shown in figure 7.12.

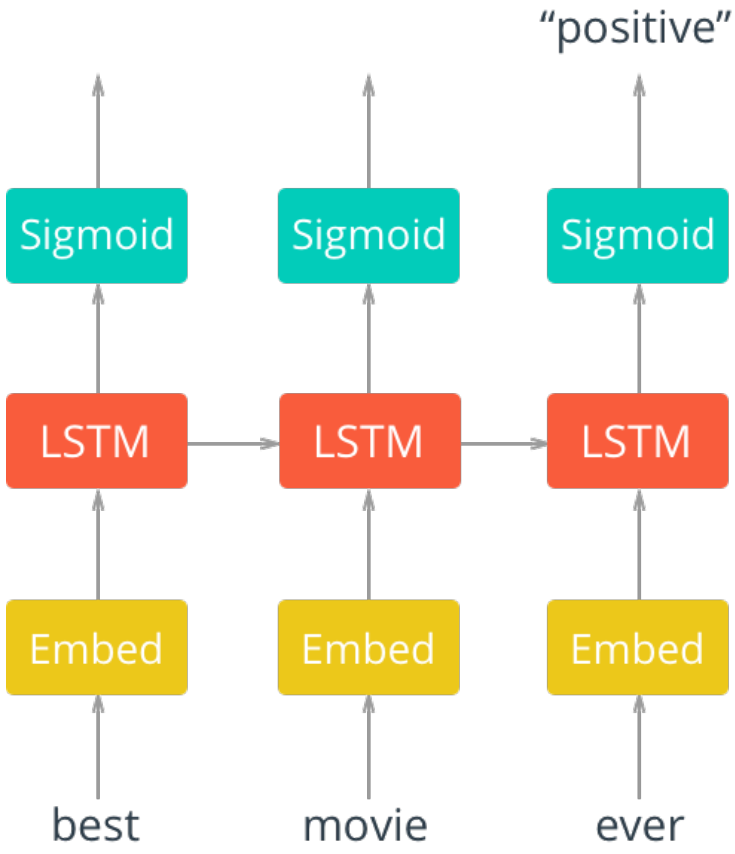

Figure 7.12: LSTM Architecture for Sentiment Analysis [20]

7.2.2 Tokenize

The Keras Tokenizer class automates the tokenization on reviews by turning the texts into space-separated sequences of words. Then the fit_on_texts function of the tokenizer class updates it's internal vocabulary based on a list of texts. Thus, a vocabulary index based on word frequency is created.

7.2.3 Padding

The list of texts is converted to a sequence of integers based on the tokens in the dictionary using the text_to_sequences function of the tokenizer class.

Then, the sequences which are shorter than the specified length are padded with zeros and the longer sequences are trimmed to that specified length.

7.2.4 Working

We build a sequential model using all layers shown in the figure 7.12. First comes the Embedding layer, that is initialised with random weights. It will learn an embedding for all of the words in the reviews. The 2d vector with a single embedding for each word in the series of input words which forms the output of the Embedding layer.

A parameter named Dropout is used for improving the model performance by reducing the effect of overfitting. This model uses a Sigmoid activation function to predict the probability of an output as 0 for small values (less than 5) and as 1 for large values (greater than 5). We add loss and optimisation functions to reduce the prediction error. The figure 7.13 shows that the embedded dimension chooses 32 layers for the input. There are 65 hidden layers and the total number of parameters used are 1,720,993.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 128, 32)           1696096
_________________________________________________________________
lstm_1 (LSTM)                (None, 64)                24832
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 65
=================================================================
Total params: 1,720,993
Trainable params: 1,720,993
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 7.13: Model Training Using LSTM

Then, we take ratings below 2.5 as bad sentiment and ratings above 2.5 as good sentiment to be fed as input to the model along with the sequence of integers values. The model is fit with the training dataset which is around 80% of the entire dataset. The fitted model uses an parameter labelled as epoch to allocate the time required to use all of the training vectors to update the weights. The callbacks are used for writing logs to monitor every

batch metrics. The Early stopping function in Keras checks the metrics and stops the training of the model if the loss and mean squared error exceeds the minimum criteria.

```
loss: 0.17%
accuracy: 93.54%
76747/76747 [==============================] - 43s 562us/step
              precision    recall  f1-score   support

           0       0.85      0.86      0.86     17539
           1       0.96      0.96      0.96     59208

    accuracy                           0.94     76747
   macro avg       0.91      0.91      0.91     76747
weighted avg       0.94      0.94      0.94     76747
```

Figure 7.14: performance metrics on the test dataset

This model has evaluated all the performance metrics on the test dataset and displayed an accuracy of 93.54% and also a loss function of 0.17%, which is the prediction error as shown in figure 7.14 and 7.15.

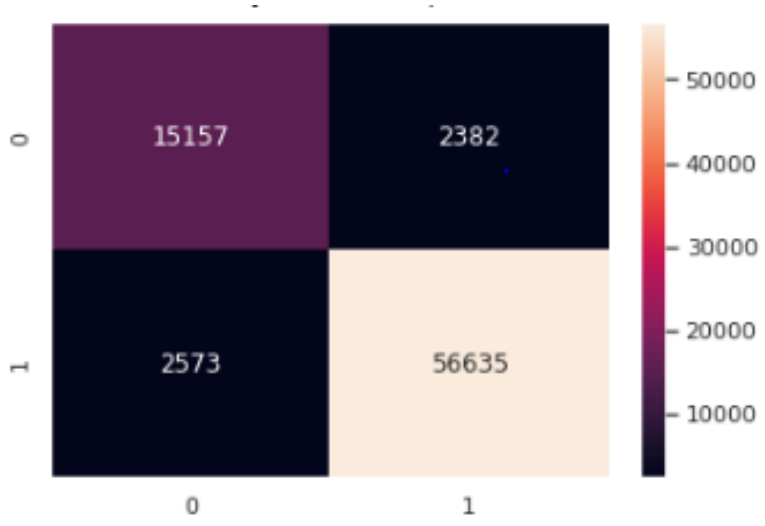

Figure 7.15: Confusion matrix for LSTM Model

Chapter 8

Aspect Based Sentiment Analysis(ABSA)

The figure 8.1 (i.e) ABSA model depicts that we can find the aspects and the related sentiment on a dataset in four distinct main processes that is explained below -

1. Learning the aspect categories in the given training dataset.
2. Extract aspect terms using Noun-Adjective Pair method on the training dataset.
3. Determine aspect categories on a test dataset using aspect terms.
4. Determine sentiment terms on the test dataset and assign polarity as positive/negative/neutral accordingly

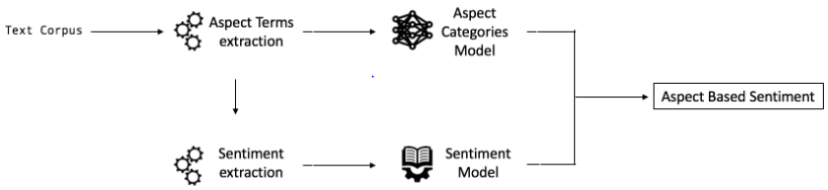

Figure 8.1: ABSA Model Decomposition [10]

From our SemEval dataset, the aspect categories are learnt. There are 12 aspect categories written below in the form of CATEGORYENTITY [10] -

1. AMBIENCE#GENERAL
2. DRINKS#PRICES
3. DRINKS#QUALITY
4. DRINKS#STYLE_OPTIONS
5. FOOD#PRICES
6. FOOD#QUALITY
7. FOOD#STYLE_OPTIONS
8. LOCATION#GENERAL
9. RESTAURANT#GENERAL
10. RESTAURANT#MISCELLANEOUS
11. SERVICE#GENERAL

8.1 Get the Aspect Terms

By using the Noun chunk dependence parser from spaCy, a very efficient NLP library written in Python, we get the aspect terms of the training dataset in each review as shown in figure 8.2.

sid	review	category	polarity	aspect_terms
1004293:0	Judging from previous posts this used to be a ...	RESTAURANT#GENERAL	negative	posts place
1004293:1	We, there were four of us, arrived at noon - t...	SERVICE#GENERAL	negative	noon place staff
1004293:2	They never brought us complimentary noodles, i...	FOOD#QUALITY	negative	noodles requests sugar dishes table
1004293:3	The food was lousy - too sweet or too salty an...	FOOD#STYLE_OPTIONS	negative	food portions
1004293:4	After all that, they complained to me about th...	nan	nan	tip

Figure 8.2: Training Dataset of ABSA

8.2 Get the Aspect Categories

To find out our aspect categories in our test dataset given the aspect terms, keras library is used for building convolutional neural network.
We use Sequential model and add layers to it-

1. The first layer has a dense layer of 512 nodes, and the shape of the input is the shape of our word vectors and the relay activation function is used to transform using weighted input value quantities.

2. The second layer is the layer for our production. Its node number is the number of outputs that we want. To predict the Aspect Groups, the softmax activation feature is used. We have 12 types of Things, so we want to have 12 nodes.

However, we can not feed words directly into the CNN in the form of strings, which is why we must encode our terms of the Dimension as vectors. This technique is called Word Embedding and consists of representing a word in a high dimension space as a vector. The Word embedding technique we are going to use here is called Word Bag and is very simple:

1. We build a matrix of all the words that exist in our vocabulary.
2. Then every vector is a hot encoded representation according to the word presence.

We will encode the aspect terms and categories in Keras and fit the keras model using the training dataset with the encoded aspect terms as input, and the encoded aspect categories as output. To prevent overfitting we use the epoch parameter and set it to 5. On the test dataset, we have to do the same processing operations, and then predict using model.predict and transform the output using inverse encode functionality to get the aspect categories.

8.3 Get the Sentiment Terms

We will get the sentiment words with the aspect categories to determine the polarities as in figure 8.3. In spaCy library, we use the Part Of Speech Tagging method to extract adjectives and verbs.

sid	review	category	polarity	aspect_terms	sentiment_terms
1004293:0	Judging from previous posts this used to be a ...	RESTAURANT#GENERAL	negative	posts place	judge previous good
1004293:1	We, there were four of us, arrived at noon - t...	SERVICE#GENERAL	negative	noon place staff	arrive act impose rude
1004293:2	They never brought us complimentary noodles, i...	FOOD#QUALITY	negative	noodles requests sugar dishes table	bring complimentary ignore repeat throw
1004293:3	The food was lousy - too sweet or too salty an...	FOOD#STYLE_OPTIONS	negative	food portions	lousy sweet salty tiny
1004293:4	After all that, they complained to me about th...	nan	nan	tip	complain small

Figure 8.3: Extraction Of Sentiment Terms From Reviews

8.4 Build the Sentiment Model

For the sentiment model, we use a very similar architecture in step 2:

1. A dense layer that takes dimensional 6000 as input word vectors, with a relu activation function and 512 nodes.
2. An output layer featuring a softmax activation function to predict distribution of output likelihood. This time with just 3 nodes, since we want optimistic, negative or neutral predictions.

The input is encoded sentiment_terms by Word Embedding Technique (explained in STEP 2: Get the Aspect Categories), and the polarities are encoded using Label Encoder() is the output. Now again we fit the CNN model with 5 epochs to avoid overfitting.
We have to do the same processing operations on the test dataset and then predict using sentiment_model.predict and transform the output using inverse_encode functionality to get the sentiment for each category in the review as shown in figure 8.4.

```
The wine list is interesting and has many good values. is expressing a  positive opinion about FOOD#QUALITY
Went on a 3 day oyster binge, with Fish bringing up the closing, and I am so glad this was the place it O
trip ended, because it was so great! is expressing a  positive opinion about RESTAURANT#GENERAL
This place has got to be the best japanese restaurant in the new york area. is expressing a  positive
opinion about FOOD#STYLE_OPTIONS
She lives nearby but had never gone to this establishment thinking that it might be too touristy. is
expressing a  negative opinion about FOOD#STYLE_OPTIONS
The pizza is overpriced and soggy. is expressing a  negative opinion about FOOD#QUALITY
Once you step into Cosette, you're miraculously in a small, off-the-beaten path Parisian bistro. is
expressing a  positive opinion about RESTAURANT#MISCELLANEOUS
Leon is an East Village gem: casual but hip, with well prepared basic French bistro fare, good specials,
a warm and lively atmosphere. is expressing a  positive opinion about AMBIENCE#GENERAL
```

Figure 8.4: Sentiment and Category Expressed In Each Test Review

We have indeed identified the sentiment expressed for the aspects in each review, thus these insights will be useful for business growth.

Chapter 9

Conclusion

In the above project, we worked in details of analyzing the text, creating aspects from it based on ontology and deriving quantifiable features out of it. This feature set helped in building the necessary models based on NLP, Machine and Deep Learning Algorithms. These models greatly assisted in uncovering the business problems and capabilities, which inturn can be progressed to business actions. Based on the review rating, restaurants are provided with sufficient data on areas to improve, their strength and weakness. Also, with the derived outputs, new entrepreneurs will have an insight and fact to expand or create outlets with higher confidence than before in a given region(India/Ireland).

Bibliography

[1] Federico Pascual, *Sentiment Analysis*, COO Co-Founder @MonkeyLearn(2019).

[2] McTear Michael, *The Conversational Interface. Springer International Publishing.*(et al) (2016).

[3] Mohini Chaudhari , *International Journal on Computational Science Applications (IJCSA) Vol.5, No.3, June 2015*

[4] Munir Ahmad , Shabib Aftab ,*INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY SCIENCES AND ENGINEERING, VOL. 8, NO. 4, JUNE 2017*

[5] Zahraa S. Abdallah, *Encyclopedia of Machine Learning and Data Mining, Springer, 2017*

[6] Maria Pontiki ,*Institute for Language and Speech Processing, Athena R.C., Greece.,2016*

[7] C.J.Hutto, *A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text,2014*

[8] Parul Pandey, *Simplifying Sentiment Analysis using VADER in Python (on Social Media Text),2018*

[9] Jason Brownlee, *A Gentle Introduction to the Bag-of-Words Model,2017*

[10] Remi Canard, *Aspect Based Sentiment Analysis,2018*

[11] Himanshu Poddar, *Zomato Bangalore Restaurants,2019*

[12] Shubhankar Rawat, *Zomato, Bangalore Data Analysis,2019*

[13] Ying Lang, *Yelp Rating Prediction with Sentiment and Topic Models. A Master's Paper for the M.S. in I.S. degree. May, 2017. 42 pages*

[14] Carson Sievert, *LDAvis: A method for visualizing and interpreting topics*

[15] Kenneth E. Shirley, *LDAvis: A method for visualizing and interpreting topics*

[16] Selva Prabhakaran , *Topic Modeling with Gensim (Python)*

[17] Shashank Kapadia ,*Topic Modeling in Python: Latent Dirichlet Allocation (LDA),2015*

[18] owygs156 ,*K Means Clustering Example with Word2Vec in Data Mining or Machine Learning,2017*

[19] Marc Kelechava ,*Using LDA Topic Models as a Classification Model Input,2019*

[20] Manish Chablani ,*Sentiment analysis using RNNs(LSTM),2017*

[21] Subarno Pala, Dr. Soumadip Ghosha, Dr. Amitava Nag ,*Sentiment Analysis in the light of LSTM RNN,2018*

[22] Dr. Soumadip Ghosha, Dr. Amitava Nag ,*Sentiment Analysis in the light of LSTM RNN,2018*

[23] Dr. Amitava Nag ,*Sentiment Analysis in the light of LSTM RNN,2018*

[24] Entrepreneur , *https://www.entrepreneur.com/article/73384*

[25] Jovelyn C. Cuizon ,*TEXT MINING CUSTOMER REVIEWS FOR ASPECT BASED RESTAURANT RATING*

[26] Sissy Themeli ,*Hate Speech Detection using different text representations in online user comments,2018*

[27] Lucas de Sá, *Text Clustering with K-Means,2017*

[28] John Coene, *Models*

[29] K Berezina, A. Bilgihan, C. Cobanoglu, and F. Okurnu s, *"Understanding satisfied and dissatisfied hotel customers: text mining of online hotel reviews*

[30] Algorithmia, *Introduction to sentiment analysis,2020*

[31] Shahules, *Zomato Complete EDA and LSTM model*

[32] Monkey Website, *https://monkeylearn.com/sentiment-analysis/?cv=1*

[33] python,*Python Machine Learning Tutorial*, python-course