2020-12-07

Ranking Aspect-Based Features in Restaurant Reviews

Jacob Ling Hang Chan
Brigham Young University

Ranking Aspect-Based Features in Restaurant Reviews

Jacob Ling Hang Chan

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Arts

Deryle Lonsdale, Chair
Earl Brown
Rob Reynolds

Department of Linguistics

Brigham Young University

ABSTRACT
Ranking Aspect-Based Features in Restaurant Reviews

Jacob Ling Hang Chan
Department of Linguistics, BYU
Master of Arts

Consumers continuously review products and services on the internet. Others have frequently relied on those reviews in making purchasing decisions. Review texts are usually free-form and associated with a star rating on a 5-point scale. The majority of restaurants receive a 3.5 or 4 star rating on average, so a standalone star rating does not provide adequate information for readers to make a decision. Many researchers have approached the problem with sentiment analysis to classify a sentence or a text as expressing a positive or a negative review. Sentiment analysis, even at the fine-grained level, can only provide classification of positive and negative judgments on any particular aspect under consideration.

The novel method proposed in this thesis provides insight into what aspects reviewers deem as relevant when assigning star rating to restaurants. This is accomplished by using an interpretable star rating classification method that predicts star rating based on aspect and polarity score from the review. The model first assigns a polarity score for each aspect in the review text, then predicts a star rating, and outputs a ranked list of aspect importance according to a widely used restaurant reviews dataset. The result from this thesis suggests that the classification model is able to output a reliable ranking from the review texts.

Keywords: sentiment analysis, star rating prediction, feature importance, random forest

ACKNOWLEDGMENTS

I am grateful for all the faculty members in the BYU Linguistics department who have helped me grow in this program. Specifically, I want to thank my thesis chair, Dr. Deryle Lonsdale, for his guidance and support throughout the thesis process. Without his thoroughness, selfless dedication, and sacrifices, I would not have been able to complete this thesis. I am also grateful for my thesis committee members, Dr. Earl Brown and Dr. Rob Reynolds, who have provided valuable feedback and knowledge.

I would also like to thank my parents, Jeremy and Dion, for their immeasurable support in my education, and without whom I would not be who I am today. I am also grateful for my two sons, Levi and Aiden, who were born during my graduate school journey. Last but not least, I am grateful for my wife, Vanessa, for believing in me; it would not have been possible without her never-ending support, sacrifice, and love.

Table of Contents

List of Figures

List of Tables

1. Introduction

In recent years, the rapid growth of ecommerce, social media, and online review websites has provided a place for users to share their experience about product, services, and business (Gretzel & Yoo, 2008). Online reviews have become an important factor for potential consumers to make purchasing decisions. 97% of consumers reported having read online reviews prior to making a decision, and 91% of 18-34 year-olds trust online reviews as much as personal recommendations (Murphy, 2019). Luca (2016) suggests that for every one-star increase on the popular review site Yelp, the restaurant receives a 5-9% increase in revenue. In addition, the study suggests that potential customers tend to select restaurants with more reviews. It is evident that online reviews have become a reliable source of information to assess strengths and weaknesses of a product or service, as well as an important factor in making a decision. Additionally, reviews are a valuable source for companies to learn from their customers. (Ganu et al., 2009).

However, it is impossible for consumers to read through the large volume of reviews of all the businesses they are considering; furthermore, the sheer volume of reviews from various websites make it difficult for companies and consumers to gain insightful information. This has sparked development on a wide range of applications and studies to measure and extract insightful information from reviews, especially in sentiment analysis or opinion mining. Sentiment analysis is a computational method to study human opinions, sentiments, emotions, and attitudes. Sentiment analysis has become an active area of study in natural language processing (Pontiki et al., 2016). Sentiment analysis is a complex task, because review texts are free-form and contain unusual spelling, spelling errors, various syntactic structures, idioms, emojis, quotations, ellipses, and sarcasm. It also requires a computer system to comprehend context, process knowledge of the review domain, and detect reviewer satisfaction. The task also involves various aspects of natural language processing such as part-of-speech tagging, dependency parsing, and entity recognition.

The goal of sentiment analysis is to output a polarity score, which classifies a sentence or a document as positive, negative, or neutral. Most sentiment analysis combines a lexicon with some type of machine learning algorithm to take into consideration syntactic information. The lexicons are usually human-defined and contain words annotated with their polarity such as positive, negative, or neutral, along with syntactic information such as detecting intensifiers or

negation to identify sentiment more accurately. The pre-defined lexicon allows computers to search and classify sentiment on a review or a sentence, however this lacks granularity into which entity receives the classification. Additionally, the predefined lexicon is labor-intensive and domain-specific.

Although many of the sentiment analyzers make use of syntactic information and context to ensure improved accuracy, they still lack granularity into which entities receive negative or positive reviews. For example, consider "Although the service was terrible, I love the food." The review is both positive and negative, and each entity being discussed has a different polarity. This has led to the development of aspect-based-sentiment analysis (ABSA) to identify the object that the opinion is targeted towards. Instead of looking at the document, sentence, clause, or phrase, aspect-level analysis directly looks at the target opinion sentiment. This method provides a more fine-grained view into particular aspects and their sentiment polarity. With the previous example, ABSA is able to extract a positive review of the food and a negative review of the service. ABSA provides a more insightful analysis for consumers and companies to understand which specific areas have positive and negative reviews. An analysis could result in hundreds of aspects and it is difficult to know which aspect is actually helpful for consumers and companies. In addition, ABSA still requires a manually-defined aspect and opinion lexicon, which limits ABSA's domain.

This thesis aims to provide a method to predict review star ratings with features including multiple aspect groups and polarity scores, and to assign rankings from highest to lowest of each aspect group according to importance as viewed by the classification model.

The research questions this thesis addresses are as follows: 1. Can aspect and polarity score predict star rating? 2. Is it possible to rank aspect based on reviewers' opinions?

2. Literature Review

There has been recent interest in analyzing and classifying reviews. The interest is largely due to the difficulty for computers to understand and analyze free-text format reviews. Typical sentiment analysis focuses solely on the text polarity, whereas aspect-based sentiment analysis can provide more detailed results in predicting sentiment polarities on a given aspect or entities in the text. This section will explore different approaches employed by relevant studies to extract user-generated reviews at a fine-grained level. It will also survey different machine learning approaches that have been employed to understand reviewers' attitudes.

2.1 Sentiment Analysis

Sentiment analysis, also referred to as opinion mining, is the process of classifying a text or sentence by its polarity. Polarity classifies text into negative, positive, and neutral. Sentiment analysis has become a major task of natural language processing (NLP) and gained major attention in recent years due to the large volume of online reviews and social media. The task often uses many NLP technologies such as entity recognition, part-of-speech tagging, and dependency parsing, making it a great challenge to investigate and test different natural language processing methodologies. However, the classification is not easy and presents many challenges to accurately detect sentiment from sarcasm, subjectivity, and a long rant in a single sentence. There has also been a surge of new applications being developed for commercial use of sentiment analysis to understand customers, monitor online presence, online marketing analysis, and product recommendations. Most of these studies and applications make use of a predefined lexicon and combine with some type of machine learning method. The prevailing methodology can be divided into lexicon-based methods and hybrid methods.

Lexicon-based approaches rely on building a sentiment lexicon of adjectives and adverbs that have been annotated by polarity. The annotation is usually done manually by humans. Lexicons such as LIWC (Pennebaker et al., 2015) and Hu-Liu04 (Hu & Liu, 2004) categorize words into binary classes of either positive or negative according to context-free semantic orientation. More advanced lexicons such as SentiWordNet associate each word with its valence scores for sentiment intensity (Baccianella et al., 2010).

Sentiment intensity or valence-based methods not only classify words into binary polarity but also note the strength of the sentiment expressed in the text. SentiWordNet is an extension of WordNet (Fellbaum, 1998) with 147,306 synonym sets annotated with three numerical scores

relating to positivity, negativity, and objectivity/neutrality. Each score ranges from 0.0 to 1.0 and their sum is 1 for each synonym set along with positive and negative scores. The scores were calculated using a complex mix of propagation methods and classifiers, unlike LIWC which was curated manually by humans (Baccianella et al., 2010). Although SentiWordNet avoided the time-consuming task of manually creating and validating such lists of opinion lexicon, a large majority of synonym sets have no positive or negative polarity and fail to account for sentiment-bearing lexical features.

Whether a lexicon is binary-based or more nuanced as valence-based, the lexicon method disregards the context and lexical properties that affect the word polarity. Syntactic structures such as negation and intensifiers provide contextual information that affect polarity. Therefore, many researchers approach the problem with hybrid solutions. Hybrid methods employ a wide range of NLP methods to obtain accurate results. For example, a lexicon can be further adjusted with word sense disambiguation to identify which sense of a word is being used in the sentence. Many words have multiple meanings and contextual meanings.

Although hybrid methods provide a more accurate method to make use of a lexicon in a more context-aware manner, they still have trouble with coverage. A predefined lexicon often ignores unseen words and important lexical features such as negation. Moreover, a sentiment intensity score can differ based on the data's genre.

2.2 Aspect-based Sentiment Analysis

Sentiment analysis, when it categorizes a text or a sentence with sentiment polarity, presents problems. For example, "The book was fun to read, but the price is too high," contains two aspects of "book" and "price". At document-level sentiment analysis, it could result in either polarity. Even at the more granular sentence level with contextual information, it is still difficult to discover when one review contains both positive and negative remarks. General sentiment analysis to classify polarity lacks important information to understand people's likes or dislikes from their opinion. Hence aspect-based sentiment analysis (ASBA) is developed as a process to extract aspects and then determine the sentiment that is expressed toward that aspect from the review (Pontiki et al., 2016). Aspect can be defined as the target that the opinion is expressed toward.

Similar to sentiment analysis there are two levels of ABSA: sentence-level ABSA and text-level ABSA. Sentence-level ABSA is to identify the aspect which an opinion term expressed

toward in a given sentence. Usually both the aspect and opinion terms are chosen from a predefined inventory. After the aspect and the opinion terms are identified, the computer will then assign a polarity to the aspect-opinion pair. Text-level ABSA identifies a set of tuples that summarize the opinions expressed in a given review (Pontiki et al., 2016).

There also are two different approaches to ABSA: aspect-term sentiment analysis and aspect-category sentiment analysis (Xue & Li, 2018). Aspect-term sentiment analysis is performed on the aspect or entity terms that are labeled in a given sentence. The analysis uses a tree structure to map syntactic dependencies for the given sentence. Then it determines the opinion term by using the surrounding relations and positions from the aspect term. Aspect-category sentiment analysis predicts the model polarity toward a predefined list of aspects that have been grouped by categories, for example *burger, fries, hotdog, ribs* can be grouped into the category of *food*.

Early approaches to ABSA rely on a pre-defined lexicon, the lexicon consisting of a list of nouns that frequently occur within the dataset and opinion term annotated by polarity. With the lexicon, the model searches for the aspect from each sentence. When matched with an aspect from the lexicon it will then look up the opinion term from the lexicon. Such approaches work well for high frequency nouns and opinion terms. However, it ignores context, negation, and low-frequency aspects and opinion.

Later studies have resolved those problems to present a more accurate and optimal method to conduct ABSA with various machine learning methods and labeled datasets. In 2014, Semantic Evaluation (SemEval) held a shared task workshop to promote the state of the art for ABSA and to provide a baseline (Pontiki et al., 2014). There was an abundant amount of participation in this particular task. At the time, the best performing model was developed by Chernyshevich (2014) using lexicon and syntactic features on each token to train a conditional random field and achieved 79% accuracy for restaurant reviews. Other studies can be generalized and grouped into pre-trained models, rule-based approaches, and recurrent neural networks.

2.2.1 Pre-trained models

Initially, ABSA was conducted as a hybrid method where users defined a list of aspects for the model to find and determine polarity for that aspect using dependency parsing. This method relied heavily on feature engineering. It was domain-specific and hence, difficult to build as a universal approach to ABSA. In recent years, the launch of pre-trained language models and

transfer learning such as ELMo (Peters et al., 2018), OpenAI GPR (Radford et al., 2018), and BERT (Devlin et al., 2019) offer a new solution to approach ABSA.

For example, Sun (2019) uses four different BERT models to detect and conduct ABSA. The study found that BERT-based models outperform the other three models in aspect detection and sentiment analysis with 93.6% accuracy, which is 2.6 % higher compared to previous studies and models. The study and previous studies (Saeidi, 2016; Ma et al., 2018) all suggest that models performing better at aspect detection tend to have weakness in sentiment analysis and vice versa. In addition, although a pre-trained model can achieve state-of-the-art performance in some cases, it is extremely difficult to replicate for a new domain.

2.2.2 Rule-based approach

Aside from using pre-trained models, some have performed sentiment analysis with a rule-based approach. VADER in particular is one of the more popular open-source models that has achieved 96% accuracy (Gilbert & Hutto, 2014). VADER stands for Valence Aware Dictionary for Sentiment Reasoning, which is a simple rule-based model for general sentiment analysis. VADER starts by constructing a list of lexical items common to sentiment expressions from corpora and with sentiment intensity rated by humans. Then the lexicon features are combined with five general rules that embody grammatical and syntactic convention in expressing and emphasizing sentiment intensity. The study found that the five rules with the lexical feature were able to achieve 96% accuracy in Tweets sentiment analysis. Although it performed very well in Tweets, it was not able to achieve the same accuracy for product reviews, with only 64% accuracy. Moreover, the method was labor-intensive. The results also suggest that the rules can achieve high performance when applied to specific genres such as Tweets in the study.

2.2.3 Recurrent neural networks

A transition-based recurrent neural network (RNN) method has also been implemented in various ABSA studies. RNN is a statistical learner for modeling sequential data, such as words, in sentences. RNN provides a framework to condition on the whole sequence and its history.

Many different variations of RNN have been applied to ABSA. Tang (2016) applied target-dependent long short-term memory (TD-LSTM) to incorporate target information. The method selects the most relevant part of the context to infer the sentiment polarity toward the target. This method was able to achieve state-of-the-art performance without an external

sentiment lexicon or syntactic parser. Numerous studies have also investigated the best method to accurately parse and identify the correct opinion term from the target. Most approaches apply different flavors of neural networks or supervised machine learning. Previous studies using neural networks can be generalized into the categories of graph-based versus transition-based parsers. Graph-based parsers use parsing as a search-based structured prediction problem to generate a scoring function on dependency trees to determine the correct tree among all trees (McDonald et al., 2006). Transition-based parsers use parsing as a series of actions that generate a parse tree. The classifier scores the possible action at each stage of the process to assist the parsing. For example, graph-based parsers have employed a tree-based recursive neural network (Socher et al., 2013). Tree-LSTM (Tai et al., 2015) uses syntactic interpretation of sentence structure to conduct ABSA. Recursive neural networks make use of external syntactic parsers that entail long processing time as well as inaccuracy (Xue & Li, 2018).

Kiperwasser & Goldberg (2016) present a different implementation of the long short-term memory (LSTM) method that is simpler and effective for dependency parsing. The method is based on bidirectional-LSTMs (BiLSTMs) where each sentence token is associated with a BiLSTM vector representing the token in sentential context. BiLSTMs employs bidirectional RNN with LSTM that takes into account both the left-hand and right-hand contexts of the vector.

The approach takes each word from a given input sentence along with its POS tag and embedding vector to generate a sequence of input vectors. Each vector is a concatenation of the corresponding words and POS vector. The embedding was trained with the model to encode each word in isolation without context. The context is then introduced by representing each input element as a BiLSTM vector. The feature function then concatenates a small number of BiLSTM vectors to parse and score using a multi-layer perceptron with one hidden layer. The feature extractor uses a greedy transition-based parsing to process the sentence and produces parse trees. The result showed that this method was able to surpass the state of the art in English and achieved 93% accuracy. BiLSTM provides a simpler feature engineering that is very accurate. This method provides an accurate parser for NLP architecture ABSA function that will be used in this thesis; more will be discussed in the methodology section.

Apart from developing the most optimal algorithm for the aspect detection and sentiment lexicon, domain portability is still an issue in many cases due to the models' and lexicons' inability to classify reviews outside of the intended domain. Most studies that are able to achieve

state-of-the-art can only accurately perform within the dataset domain. For example, *food* is a frequently occurring aspect in restaurant reviews, but it does not occur in electronic products reviews. The opinion term *cheap* could be positive in describing a restaurant being affordable, though it would be negative in review concerning the product quality being "cheap".

2.3 Predicting Star Rating

Review rating prediction is a very challenging task, largely due to review text being free-form. The review might contain multiple or conflicting opinions for various aspects. Pang & Lee (2005) first approach this problem in classifying reviews on a one-to-five-star scale instead of the binary output of "thumbs up" or "thumbs down". The study compares human evaluation of classifying star rating to support vector machines (SVM). Although the study can group reviews into 1 star, 2 stars, and 3 stars and above without truly predicting on a 5-point scale, it provided groundwork for many future studies.

Synder & Barzilay (2007) also conducted a multiple aspect rating prediction study using the Good Grief model. This model scores each aspect based on a reviewer's degrees of satisfaction and uses an agreement model to predict whether all ranked aspects are equal. The joint method ranks the review texts on a 5-point scale and it was able to achieve 67% accuracy on the test set compared to the majority baseline of 58%.

Yu et al. (2015) also use a Yelp review dataset to investigate and compare machine learning algorithms to predict star ratings. The goal of the study was to understand user review content to better recommend businesses to the user that would likely rate higher than others. The study utilized information such as a user's review histories, a restaurant's meta information, and sentiment features to predict star ratings. Three machine learning algorithms were used to compare performance: linear regression, latent factor model, and random forest (RF). The study suggested that RF performed better compared to the other two. RF outperforms other models because of its ability to use features such as sentiment parameter and average star rating to predict star rating. This study motivated the current thesis to investigate sentiment analysis features to predict star rating with RF. Moreover, RF also offers feature importance, which allows ranking of each feature in relation of influence to the prediction.

2.4 Feature Importance via Random Forest

RF is an ensemble of decision trees trained through the bagging method. Decision trees is a versatile machine learning method that can perform both classification and regression tasks as

well as multi-output tasks. It is often used to train predictive models using a tree structure to present decision and decision making (Geron, 2019). Suppose the model is to determine the star rating of a restaurant's review based on a 5-point scale. It will start at the root node, where this node asks if the food was scored higher than a 0.8. If it is, then the model will move down to the root's left child node. In this case, it is a leaf node indicating that it does not have any child nodes. The leaf node does not ask any questions and simply assigns the predicted class for the node and the decision tree for restaurant review as 5 stars. Suppose there is another review, where this time the food was scored 0.4 lower than the 0.8 threshold. The model will move down to the root's right child node which is not a leaf node. The node will then ask another question: "Is the score of environment 0.6 or lower?" If it is, then the review is more likely a 3-star review.

A node's samples attribute derives from the number of training instances it applied to. If the training set has 100 instances with food quality greater than 0.8, a node's value attribute will indicate how many instances of each class this node applied to. Moreover, a node's impurity is measured by a special attribute, to be discussed shortly. The decision tree method is intuitive and easily interprets decision paths.

Random forest is an ensemble method that aggregates the predictions of a group of predictors to gain better predictions. The method trains a group of decision tree classifiers each from a randomly selected subset of the data. The predictions combine all trees to predict the class that gets the most votes from the trees, also known as a hard voting classifier. Even if one of the classifiers has low accuracies it will still be able to achieve high accuracies, because the prediction is based on majority vote. However, ensemble methods can only predict accurately when the predictors are independent from one another. Independence implies training diverse classifiers using different algorithms to increase the chance of different errors and improving ensemble accuracy (Ho, 1995). This is done by bagging decision trees in order to reduce variance of a single tree to improve prediction accuracy.

RF became popular for machine learning for its interoperability including providing insight into the data and model. For example, a great quality of employing RF is the ease of measuring the relative importance of each feature in a prediction model. The is done by assigning a score to each input feature. Feature importance has been used to interpret the data, model, and understand which feature is useful in a predictive model (Geron, 2019).

There are two methods to measure feature importance: VI and GI. VI is computed from the average decrease in model accuracy on the training sample, when the values of the randomly selected feature are permuted (Altmann et al., 2010). GI is measured by how much the tree nodes that use that feature reduce impurity on average across all trees in the forest. A node's impurity is measured by gini. A pure node has a gini score of 0 when all training instances it applies to stem from the same class. For example, if the left child node of the root only applies to Price training instances, it is pure and receives a gini score of 0. Equation 1 shows how the training algorithm computes the gini score G of the I node. Furthermore, the score is a weighted average, where the weight of each node is equal to the number of training samples that are associated with it (Strobl et al., 2007).

$$G_i = 1 - \sum_{k=1}^{n} P_{i,k}^{\ 2}$$

Equation 1. Gini Impurity
P is the ratio of class k instances among the training instances in the i node.

3. Dataset

The dataset for the current thesis is provided by Yelp, an American company that publishes business information and crowd-sourced reviews of businesses. The company has an average of 28 million unique users per month. The dataset[1] is a subset of business data open to the public by the company for academic purposes. The Yelp dataset was chosen due to its large quantity of reviews, as well as the site's wide popularity for the public to leave reviews for businesses, especially restaurant reviews. The dataset is from the 2020 data release and it contains 7,734,455 reviews from 209,393 businesses and 1,968,703 users across 11 metropolitan areas.

The dataset includes 5 json files, one for each object type (business, review, user, check-in and tips) where each file consists of one json object per line. The business file contains information including location, categories, names, average star rating, review counts, and other meta information. The review file contains the full review text, user id, business id, star rating, and other metadata. The review text is free-form along with a 5-point star rating scale. All the data required for the current thesis is contained in the business.json and the review.json files; the rest of the files are not necessary for the current thesis.

Within the business file, the dataset describes different categories that each business belongs to. The categories include Entertainment, Beauty, Home service, Automotive, Fitness, Health/Medical, Shopping, Food, Hotels/Travel, and Restaurants. However, many businesses belong to multiple categories, making it impossible to clearly group business per category. For example, a gas station could have all of the following categories indicated in the categories field: Grocery, Auto Parts & Supplies, Automotive, Auto Repair, Convenience Stores, Oil Change Stations, Food, Towing, and Gas Stations. With the majority of businesses representing a wide range of categories, the only category that can be grouped correctly would be the Restaurants category.

The Restaurants category consists of 63,944 restaurants, making up 30.5% of all businesses within the dataset. In addition, restaurant reviews also accounted for 63.1% of all reviews in the dataset with 4,882,741 reviews. With every category having very distinct entities and opinion terms that only exist within the category's domains, it would be difficult to have a

[1] Yelp Open Dataset at `https://www.yelp.com/dataset` (retrieved October 20, 2020).

measure of feature importance in general. Therefore, it is necessary to conduct the analysis per category instead of in general. As most categories are loosely defined and contain insufficient reviews, the current thesis will only use Restaurants, the only clearly classified business categories as well as representing the majority of the reviews from the dataset

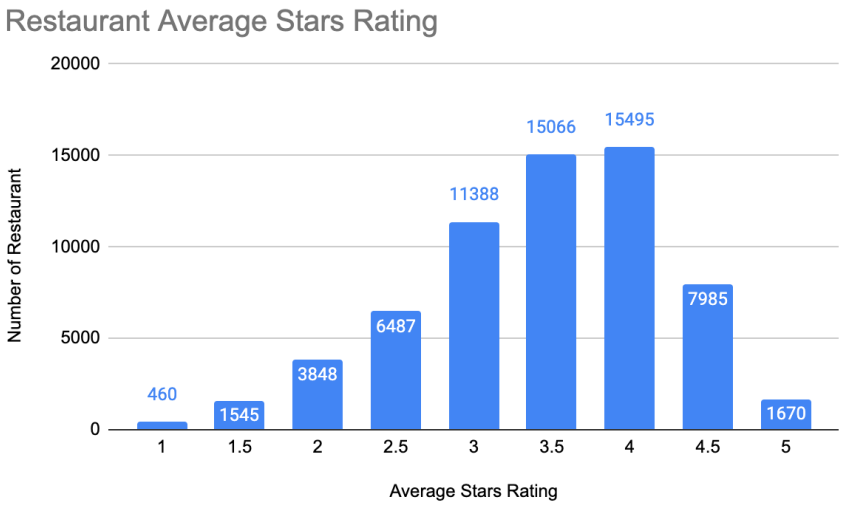

Figure 1. Average star rating in the Restaurants category from the dataset

According to the dataset, each restaurant has an average of 3.4 star ratings and 76 user reviews. 48% of restaurants fall within the 3.5 to 4 stars range, and 15 % of restaurants receive an average rating of 4.5 stars or higher stars (Figure 1). Given that 48% of restaurants fall between 3.5 to 4 star range, there is a need for the current thesis to use natural language processing to analyze review content to distinguish a 4-star-user review from a 5-star-user review. In order for readers to find reviews useful to read and business to improve upon the review, the current thesis implementation of feature importance will be able to provide a more in-depth analysis of reviews in understanding common consensus on what is deemed important to review writers.

The location distribution from the data is spread across 11 states and provinces across the United States and Canada. 25% of the restaurants are in Ontario, Canada while the majority of the reviews are from Nevada and Arizona, with over a million reviews for each state. Others include 17 states or provinces with one to three restaurants per location, with a total of 35 restaurants and 1,572 reviews; hence these are grouped as others to avoid skewing the location data.

Table 1. Number of reviews and restaurants per Location.

State/ Province	Number of Reviews	Number of Restaurants
AB	69,835	3,245
AZ	1,387,325	12,124
IL	31,250	702
NC	328,710	4,655
NV	1,632,060	8,340
OH	263,374	5,914
ON	648,442	1,6221
PA	240,194	4,259
QC	158,132	6,228
SC	18,920	427
WI	102,925	1,794
Others	1574	35

Note. Others include: Alabama, Arkansas, British Columbia, California, Colorado, Florida, Hawaii, Manitoba, Nebraska, New York, Oregon, Texas, Virginia, Vermont, Washington, and Undefined.

As previously mentioned, the majority of the restaurants received an average rating of 3.5 to 4 stars. Yet 44% of the reviews in the dataset had five stars. There are twice as many 5-star reviews compared to 4 stars and more than 3 times compared to 3-star reviews.

Table 2. Number of reviews per star ratings.

Star Ratings	Number of reviews
1	628,004
2	456,590
3	639,748
4	1,254,009
5	2,077,510

4. Method

The goal of this thesis is to predict star rating from aspect-based sentiment analysis results and rank feature[2] importance to produce ranked insight from review text. This section will explain each step of the methodology design and process shown in Figure 2.

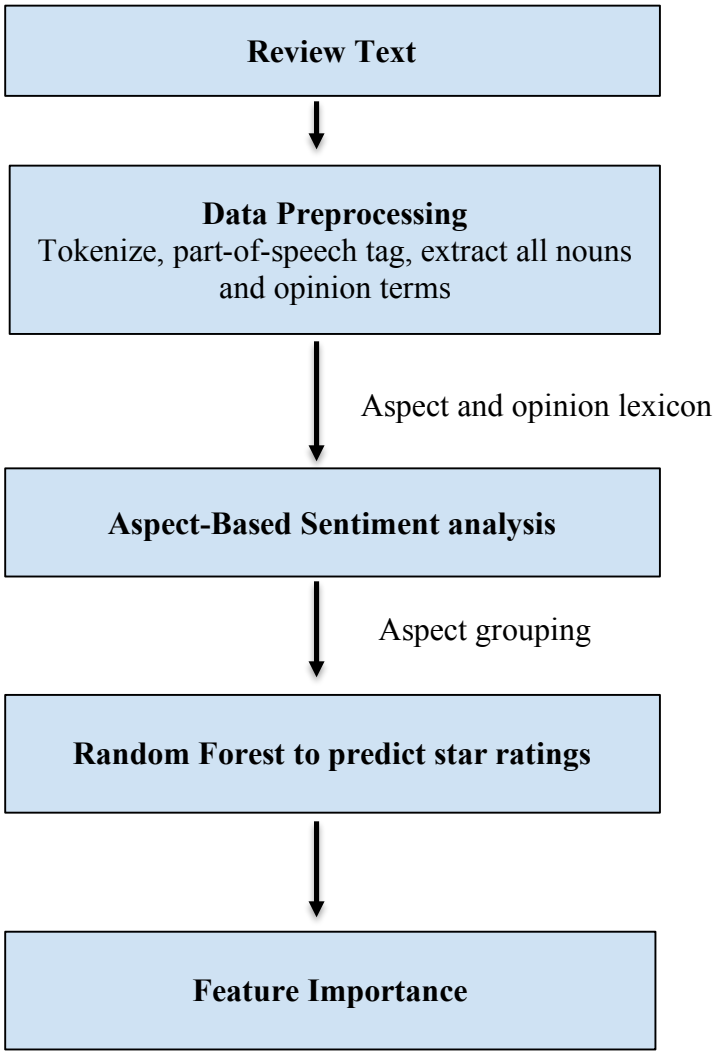

Figure 2. Process of the review analysis pipeline.

4.1 Data Sampling and Preprocessing

Since 60% of the reviews in the dataset constitute Restaurants reviews, this thesis restricts the categories to Restaurants reviews only. The dataset contains 5,055,992 restaurant reviews across the USA and Canada. For the purpose of this thesis, reviews were selected from Las Vegas, Nevada, the most reviewed city within the category. Nevada has 1,632,060 reviews

[2] See code at `https://github.com/jacoblhchan/Aspect-ranking`.

in the dataset; however it is still not feasible to process millions of reviews. To ensure a feasible computation, a random sample of 5000 reviews of each star rating from Las Vegas's restaurants with a total of 25,000 reviews have been randomly selected for processing. As previously mentioned, 44% of all reviews in the dataset are five-star reviews, therefore randomly sampling the dataset would create imbalance of star ratings. Qiu et al. (2018) suggest that an imbalanced dataset can cause significant impact on the effectiveness of the prediction model. Creating a balanced dataset can create more accurate predictions.

The first script is written in Python with the Pandas library to extract a list of restaurant business IDs within Las Vegas from the business.json file. Pandas is an open-source Python library that allows fast data manipulation and analysis. With a large json file of 209,393 json objects, it can read, handle it quickly and output the data into csv format. The list of business ID's is then used to extract the ratings and reviews from the review.json file. Pandas also offers a random sampling feature which allows users to randomly sample the data. This feature was used to randomly select 5000 reviews from each star rating.

An additional step was also conducted to extract English reviews only and avoid any confusion over non-English text in the pipeline during later analysis. This was performed by langdetect, an open-source library in Python.

The final step of preprocessing the pipeline involved cleaning each review text. With Yelp allowing users to write free-form review, resulting in many textual anomalies, it was crucial to ensure that all words were in lowercase to avoid problems in part-of-speech tagging and dependency parsing. This is a typical preprocessing stage in NLP pipelines. Each review was then separated into 5 different datasets into csv files according to its star rating.

4.2 Feature Extraction

The feature extraction phase is to extract the aspect and opinion terms from each sentence in a review. The aspect is the entity that is being described in the review, while opinion term is the word that is used to describe the aspect. With the list of aspects and opinion terms as input, aspect-based sentiment analysis (ABSA) can subsequently be conducted for the next phase.

The next step was to extract aspect terms from the dataset. NLTK is an open-source natural language processing library that can perform tokenization, part-of-speech tag, as well as many other natural language processing features (Bird et al., 2009). NLTK was selected for aspect term extraction because of its outstandingly faster processing time compared to other NLP

Python libraries such as spaCy. In addition, NLTK uses strings as input and output, making it easier to parse out nouns or proper nouns. SpaCy uses an object-oriented approach, where it returns document object with words and sentences as objects themselves. Moreover, spaCy requires loading language models at run-time requiring large memory usage.[3]

Each review was processed by determining a token and its part of speech at this phase. If the token was labeled as a noun (NN, NNS) or proper noun (NNP, NNPS) by NLTK, it was then extracted from the review and appended to a Python list. After each review had been processed, the list of nouns and proper nouns was exported as a csv file from the Python list, with each noun or proper noun per row. This step created a set of predefined aspects for ABSA and opinion term extraction. ABSA requires a pre-defined aspect to ensure the model to search for an opinion term whenever an aspect is present in the review. This is a necessary approach for ABSA to properly locate and analyze the aspect and opinion pair per the review domain.

In order to extract and generate an opinion term corpus from data that exhibits aspect, the spaCy dependency parser was then used for its ability to manage advanced language processes which NLTK lacks, such as dependency parsing with state-of-the-art accuracy. With the spaCy-BIST parser, it was possible to extract a set of opinion terms based on dependencies involving the entity. The BIST parser is a graph-based dependency parser using bidirectional-LSTMs (BiLSTMs), features extractors to determine each word token by associating it with a BiLSTMs vector representing the token in the sentential context. The BIST dependency parser is able to achieve 93.9% accuracy English on the standard Penn Treebank Dataset, reflecting state-of-the-art accuracy in English (Kiperwasser & Goldberg, 2016).

As mentioned previously, the script was written in Python using the spaCy library and took each review as an input for preprocessing. The preprocessing of the review text then first broke the review apart into sentences, then tokenized the sentence and annotated each token with part-of-speech tags and dependencies. The dependency model for this thesis used a pre-trained BIST model provided by spaCy to extract the relationship between the aspect and the opinion term that is modifying it. Upon each review was processed and parsed, the adjective describing the entity was extracted into Python list and also output as a csv file with each opinion term per row.

[3] See `https://spacy.io/`.

4.3 Sentiment Analysis

With the aspects and opinion terms extracted from all the reviews from the Yelp dataset, ABSA was then conducted. ABSA took in three files (the aspects, opinion terms, and all the review text in csv files format) and then it generated a corpus of aspects and opinion terms to the search for matched aspects from each review. It then conducted sentiment analysis to output a list of aspects and opinions from each review, along with each polarity score to indicate the opinion as positive, neutral, or negative.

The ABSA inference step relies on NLP Architect[4], an open-source library developed by the Intel AI Lab. NLP Architect ABSA can take in the three files as input, to detect and extract syntactically related aspects and opinion pairs produced by the BIST parser. The polarity score was estimated using ELMo word embeddings[5] to determine similarities between the extracted opinion term and a set of generic opinion lexical items. It is also simple to use in processing large quantities of reviews. Its command line usage solely requires pointing to the correct aspect and opinion term, and the reviews' text file path. The library processes each review and outputs to a csv file with the aspect and opinion and the polarity score from all the reviews per row. In addition, it is highly accurate and has the ability to detect events from a given set of aspects and opinion terms. Moreover, ABSA can extract events from multi-word aspects such as *chicken wings* from the parse tree, as well as looking for intensifier and negation terms to determine polarity correctly. With ABSA, multi-word aspects that often occur in food dishes' names can be avoided, as well as averting mislabeling reviews polarity by only looking at the event alone.

The process of ABSA first took in the pre-defined aspect, opinions, and reviews file as csv and generated a Python list. Each review was then looped through to determine if any word in the review matches with the aspect list from the review file. If it did, a dependency parse converted the sentence into a tree structure. The tree was then used to detect events as well as to look for negation and intensifiers to determine its polarity and return the result. For each result, it included aspect, opinion, and score stored into a dataframe pending completion. After all the reviews were processed, the result was then aggregated into a csv file as output with each pair of aspect, opinion, and polarity score per row.

[4] See `https://github.com/IntelLabs/nlp-architect` (retrieved October 20, 2020).
[5] See `https://tfhub.dev/google/elmo/2` (retrieved October 20, 2020).

4.4 Feature Grouping and Importance

Given aspect, sentiment, and polarity score as features, RF can measure the relative importance of each feature from the list. This thesis employs an RF classifier along with importance to assign a relative score to aspect features for predicting star ratings.

Each star rating outputs thousands of aspect and opinion pairs, making it impossible to measure each feature importance or variance. Moreover, it could also create an overfitting problem with the RF model with thousands of aspects. Many aspects are similar and only occur one time, hence these can be grouped as one feature. For example, *staff, manager, waiter, waitress, server,* and *waitstaff* can be grouped as "staff".

To ensure a readable feature importance report and accurate model, the current stage performed aspect grouping into ten groups defined as follows: Food, Location, Environment, Reservation, Service, Time, Price, Staff, and Order. The creation of each Python group is a manual process to group similar aspects into a group. Group membership is assigned by a simple Python list lookup; if an aspect belongs to a group, the aspect will be classified to the group that it belongs to.

Scikit-Learn (Pedregosa et al., 2011) is an open-source machine learning library for Python that provides various tools for model fitting, data preprocessing, model selection, and evaluation. Its built-in machine learning algorithms and models are straightforward and have been used by thousands of research projects and organizations. The current thesis uses version 0.23.2 of Scikit-Learn.

Among its capabilities are RF classifier and feature importance analyzer. Scikit-Learn has a default Feature Importance function that provides fast calculation for large datasets and simplicity in retrieving results. However, it is prone to bias in determining feature importance. To combat bias in the result, the default Feature Importance tool employs the Drop Column method to decrease and discover bias that could occur.

Drop Column is a convenient method to investigate feature importance at a more granular level. This approach compares a model with all features versus a model that has dropped a feature and outputs the result. This allows detection of bias that frequently occurs in Scikit-Learn's implementation of RF and Feature Importance. Considering that the dataset only has ten feature groups with 2,5000 reviews, its computation is straightforward.

Other approaches are also available from Scikit-Learn such as permutation feature importance, which directly measures feature importance by observing random re-shuffling. It also preserves the distribution of the variable of each prediction that influences model performance. However, this approach sacrifices information about generalization of the model and can overestimate the importance of correlated predictors (Strobl et al., 2007).

The Scikit-Learn library measures feature importance by looking at tree nodes. It computes the amount of tree nodes that use the feature and reduces impurity on average across every tree in the forest. The average is also weighted by each node's weight; the node weight is equal to the number of training samples that are associated with it. Scikit-Learn can automatically determine the score for each feature after training, then scales all important results to have a sum of one. Therefore, the result is able to break down by percentage and compare by importance. Users can then apply this information to compare businesses across their area or in general.

Feature importance processing for this thesis was written in Python with the Scikit-Learn Random Forest Classifier using 500 estimators with random shuffling with each predictor variable set at 42. The Random Forest features are defined by the ten pre-defined feature groups and each feature group is assigned as the X-axis and the star rating as the Y-axis to build a forest of trees from the input.

The input combined all star rating ABSA results as one csv input file and then transformed it into a usable table to perform Random Forest. The table was designed with feature groups and star rating as column heads and each review in a row. If a feature was present in a review, the polarity score was the value for the review feature, and the star rating was under star rating columns. After the csv file transformation, the file was read into the Python script to perform Random Forest classification and feature importance. The script outputted the feature importance along with its associated feature. The Drop Column method was also conducted after the produced overall result and it was done by Pandas Drop Column function, dropping specific columns by name or at random.

4.5 Evaluation

To determine the model accuracy, this thesis compares the result of two restaurant review datasets extracted from Las Vegas, Nevada. Both datasets are extracted from the same location to

measure if the model can output correlated results. The two datasets were randomly sampled from reviews and each contains 25,000 reviews.

The two ranked feature importance outputs are evaluated with Spearman's rank correlation coefficient (Equation 2). The method measures the strength and direction of the monotonic relationship associated between two ranked variables, which in our case, is aspect group as ranked from the first dataset and from the second dataset. A monotonic relationship assumes that when a value of one rank increases, the value of other dataset will also increase; or as the value of one rank decreases, the other rank value decreases as well. The equation returns the correlation coefficient between the range -1 and 1. The result *r* uses the p-value to interpret the statistical significance of the coefficient (Zar, 1972).

$$r_s = 1 - \frac{6\sum D^2}{n\left(n^2 - 1\right)}$$

Equation 2. Spearman's Rank-Order Correlation Coefficient
D is the difference in paired ranks and n is the number of cases.

This evaluation was chosen because the output is nonparametric—it is ranked by importance but not normal distribution[6]. Since both the first and second dataset were randomly selected samples from the same Las Vegas Restaurant category from Yelp, if the model is accurate, one would expect Spearman's rank correlation to indicate a strong association between the results from both datasets.

[6] Both Kendall's tau and Spearman's rho rank correlation methods achieve similar results.

5. Result and Discussion

After the final step of Feature Importance, the Python program outputted the overall result from all ten groups of features from both datasets. The sum of feature importance is equal to 1. As previously mentioned, since random forest feature importance computations are prone to produce bias, a separate result with the Drop Column method was also conducted on dataset A to detect bias in the RF model (Table 4).

The output of both datasets is shown in Table 3. It is intuitive that "Food" is the most important feature for a restaurant from both datasets, along with "Environment/Atmosphere" being the second, and "Service" comes third. "Location and Order" are nearly as important as "Service" with an importance score of a 0.004 difference in dataset A. However, "Order" is ranked number eight in dataset B. "Staff" has a 0.043 difference between the two datasets; it ranked number six in dataset A but one rank higher in the dataset B. "Experience" ranked number seven in the first dataset with a slight lower score compared to dataset B. "Reservation" ranked as the eighth most important to reviewers. The least important feature groups are "Time", and "Price" in corresponding order in both datasets.

Table 3. Feature Importance result from first and second dataset.

	Dataset A		**Dataset B**	
Feature	**Feature Importance**	**Rank**	**Feature Importance**	**Rank**
Food	0.198	1	0.216	1
Environment	0.164	2	0.114	2
Service	0.139	3	0.108	3
Location	0.111	4	0.097	4
Order	0.108	5	0.087	8
Staff	0.099	6	0.095	5
Experience	0.079	7	0.091	6
Reservation	0.060	8	0.089	7
Time	0.029	9	0.079	9
Price	0.016	10	0.025	10

Results from both datasets showed a strong, positive correlation with a value of 0.93 for the Spearman's Rank-Order Correlation Coefficient. The p-value is 0.0001, indicating that the correlation between both results is statistically significant with 95% confidence.

A final check was conducted by the Python Random package to drop a column from the feature to detect any bias in the previous result. The script randomly selected to drop "Time" as a

feature. Generally, when bias occurs in feature importance ranking would change. In this case, as shown in Table 4, the result closely resembles the previous two results. However, "Order" is now being 0.003 more important than Location.

The difference of ranking in Table 4 does not constitute bias from the Random Forest. Both features have close proximity and feature importance scores from the previous two results. When bias occurs at feature importance, usually the feature is extremely high compared to other or uncommon features being the most prominent. This also indicates that the Random Forest Feature Importance model did not have overfitting or underfitting problems. The consistency between the three result outputs prove that the results are a reliable interpretation of the variable importance measure from the random forest model. Furthermore, the forest is built from unbiased classification trees.

Table 4. Feature Importance after dropping random feature (Dataset A)

	Dataset A		**Drop-Column Method**	
Feature	**Feature Importance**	**Rank**	**Feature Importance**	**Rank**
Food	0.198	1	0.204	1
Environment	0.164	2	0.172	2
Service	0.139	3	0.138	3
Location	0.111	4	0.114	4
Order	0.108	5	0.111	5
Staff	0.099	6	0.106	6
Experience	0.079	7	0.083	7
Reservation	0.060	8	0.058	8
Time	0.029	9	-	-
Price	0.016	10	0.014	9

The accuracy of the star rating prediction model with sentiment features was 52.4%, which was 13.2% higher compared to 39.2% baseline established from the Yelp 2014 dataset (Wu et al., 2018). However, the result was lower compared to a neural network prediction model at 68.6% accuracy (Wu et al., 2018), as well as the current state-of-the-art decision tree prediction model at 82.5% accuracy tested on the Yelp 2019 dataset (Chen & Xia, 2020). All previous studies that have achieving higher accuracy was due to their goal of developing a new state-of-the-art method. This thesis aims to develop a feature importance ranking procedure instead of developing a method to achieve the next state-of-the-art star rating prediction model.

5.1 Result by group

The result from the first dataset suggests that Food, Environment, and Service have effects on star rating. The ten feature groups combined contain over 70,000 aspects and about 1,000 unique aspects excluding misspelling. This discussion section explains and describes each group.

Food: Food includes food names and most commonly using the word *food* by review writers. Writers usually review the food quality and portion sizes. The result being the most significant effect on star rating is reasonable to expect as the main purpose of a restaurant is to provide food. *Food* is a frequently used word in both positive and negative reviews.

Environment: Environment or Atmosphere contains words such as *decoration, bathroom, table,* and *chair*. This is the second most important feature, as the environment such as cleanliness of a restaurant has been described as a crucial factor of the dining experience, which also reflects the customer's comfort in a restaurant and thus future revisitation.

Service: *Service* is a word that usually occurs within 1-4 star ratings, and it is generally associated with negative opinion words and polarity score. It is generally the result of delayed or inattentive services from restaurants' hosts and servers.

Order: This feature group includes all inflectional forms of the root word "order". This feature appears most frequently in 1 or 2 star ratings by the reason of food order inaccuracy. This feature is heavily attributed to restaurants receiving negative reviews. Therefore, it is ranked number four in importance.

Location: The use of location is an immense factor for our dataset, with the fact that Las Vegas had over 42 million visitors in 2019 (LVCVA, 2020). A restaurant location is essential for tourist convenience and an important factor for vacation plans. This aspect suggests that close proximity to tourist attractions and accessibility are important features for Las Vegas restaurant reviewers.

Staff: The group includes words such as *waiter, waitstaff, staff, manager, waitress, busser, cashier,* and *front desk.* The use of this aspect reflects largely on lower star ratings for reviews, with complaints about specific staff service and behavior.

Experience: The feature group only contains the word *experience*. Dining experience as a whole seems to be less important and less mentioned by reviewers, compared to service and environment.

Reservation: A mistake in reservation accounts for 5% importance in restaurant reviews and results in lower star ratings.

Time: The use of this feature group usually describes the slowness of their dining experience or waiting to be seated. This is the second-least important feature from the feature groups.

Price: Price is the least important feature from the feature group. It mainly describes the price of food and drinks within the restaurant. This is a difficult aspect to access with sentiment analysis as food price being cheap can be a great opinion for or against a restaurant. It might impact the polarity score as well as the feature importance. It might suggest that the reviewers already know the price range of the restaurant prior to choosing it.

5.2 Summary

After an analysis of 50,000 randomly chosen reviews from the Las Vegas Restaurant Yelp reviews dataset, each review was processed to extract ABSA features. The aspect and polarity scores from ABSA were grouped manually into ten groups. The data was then trained by a Random Forest prediction model to predict star rating. The prediction model achieved 52.4% accuracy. The model outputted the Feature Importance ranking for each feature grouped from most important to least important; for the first dataset the ranking was: Food, Environment, Service, Order, Location, Staff, Experience, Reservation, Time, and Price.

The feature importance result shows an accounting of reviewers' importance in determining review star rating on a scale of one to five for restaurants in the Las Vegas, Nevada area. Prior analysis of review text mainly classifies each review by polarity score. However, this thesis analyzes reviews by ranking feature importance. The method and result also provide a more in-depth feature extraction analysis on customer reviews at the aspect level. The output also appears to show no bias from feature importance generated from Random Forest. The result also able to shows feature importance from Random Forest based on aspect and polarity score in predicting star rating.

6. Conclusion

6.1 Summary of Findings

The goal of the present thesis was to determine a method to analyze reviews to understand Yelp restaurants review writers' perspectives and rank values via feature importance. The rankings from the two random samples suggest that the model is able to produce highly correlated results.

Although the restaurant review result does not produce a revolutionary outcome, the method can be applied for other applications and industries. For business, it can be adopted to compare feature importance from competing businesses and products to further understand their customers and market. Companies can also apply this method to understand and compare their customers in different countries or regions, thus to compare what the user group indicates as important. Users can benefit from feature importance to compare and make decisions based on differences in feature importance between products and businesses. Instead of reading through 30 to 50 reviews, feature importance can identify reviews that mention top features and recommend those selected reviews for readers.

The methodology in this thesis was able to provide an agnostic lexicon machine learning ABSA model. The part-of-speech parser extracts every possible aspect and opinion term from every review, and to detect aspect and determine opinion term with a state-of-the-art BiLSTM dependency parser. Regardless of the review domain, the method designed by the current thesis is able to auto-generate a lexicon without manual work and without a predefined lexicon. However, there are still limitations with the present methodology employed in the thesis.

6.2 Limitations

One of the potentially biggest limitations is validating the accuracy of the Random Forest classifier in predicting star rating. The prediction model relies on ABSA results as features cannot be validated without a test or validation dataset. Without a human-annotated standard to validate and evaluate the model performance against raters'opinions, it is impossible to determine the most optimal parameters for the random forest classifier. Tuning the model's hyperparameters can impact the output from the model. The setting of hyperparameters includes the number of decision trees in the forest and the number of features considered by each tree when splitting a node. This impacts feature importance. The current thesis did not explore finding the most optimal setting for the model as well as accuracy of the model. In addition, the

feature grouping and the number of feature groups is determined prematurely without gathering more data and additional feature engineering. In the future, the study needs to rely on experimental results in attempts to evaluate different parameter settings. Another method could be to generate a baseline model to compare against the random forest classifier to validate model performance.

A second limitation is the lack of diversity in the dataset. The current thesis only tested on one city (Las Vegas, Nevada) restaurant reviews from Yelp. The research question should be tested on multiple business categories and from different geographical locations. It is difficult to know if the method would perform and produce similar results for other review categories and locations.

The final limitation is the accuracy in extracting aspect and opinion words. Although the model employed for aspect-based sentiment analysis with the BIST parser is able to achieve state-of-the-art accuracy, without labeled data to validate, it is impossible to interpret accuracy for the present dataset.

6.3 Implication

The result can apply the same methodology for different review data to provide more value to users. For example, the method can be adopted for businesses or products to better understand their users or customer base. Furthermore, companies can apply this method to compare rivalries products and businesses to interpret their differences based on differences in importance for improvement.

Moreover, review readers can filter and identify helpful and essential reviews with feature importance from a large volume of reviews. Many review websites such as Yelp provide useful voting features from review reader ratings. Previous studies (Diaz & Ng, 2018; Hu & Chen, 2016; Ganu et al., 2013; Ghose & Ipeirotis, 2007) have investigated automated processes to identify helpful reviews from the large volume of reviews on a single product. Feature importance ranking can provide data and metric to measure review helpfulness to the readers, thus to filter review with low ranking features to provide useful reviews that drives decision making.

6.4 Future Research

Additional research should address issues mentioned previously in the limitation section.

Specifically, a method to better group aspects into feature groups. The selected feature group can impact the parameter and the result of the random classifier. The current manual grouping process is not scalable for a large or a multiple-domain dataset.

The thesis uses the Drop column method to detect bias with Random Forest Classifier feature importance output. Strobl et al. (2007) suggest permutation the feature importance method to consider a variable important if it has a positive effect on the prediction accuracy. Studies have proven that the permutation method is more reliable compared to the Drop Column method, which this thesis employed. Investigating and comparing the result can be beneficial to find the most optimal methods.

The methodology of this thesis uses many different technical methods to extract data such as aspect, opinion, and polarity score. Many steps can be avoided with labeled data, moreover, it will assist and simplify the feature engineering process. With transfer learning and other machine learning methods, the labeled data may also benefit from extracting more data from unlabeled data and reviews.

7. References

Altmann, A., Toloşi, L., Sander, O., & Lengauer, T. (2010). Permutation importance: a corrected feature importance measure. *Bioinformatics*, *26*(10), 1340-1347.

Baccianella, S., Esuli, A., & Sebastiani, F. (2010, May). Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. *Proceedings of the 7th Conference on International Language Resources and Evaluation (LREC '10)*; European Language Resources Association (ELRA): Valetta, Malta; pp. 1660-1664.

Bird, S., Loper, E., & Klein, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.

Chen, Y., & Xia, F. (2020, August). Restaurants' Rating Prediction Using Yelp Dataset. In *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)* (pp. 113-117). IEEE.

Chernyshevich, M. (2014). IHS R&D Belarus: Cross-domain extraction of product features using conditional random fields. *Proceeding of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. pp. 309-313.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of North American Chapter of the Association for Computational Linguistics-Human Language Technologies 2019,* pp. 4171–4186.

Diaz, Gerardo Ocampo, and Vincent Ng. (2018). Modeling and prediction of online product review helpfulness: a survey. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 698-708.

Fellbaum, C., (1998, ed.) WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press.

Ganu, G., Elhadad, N., & Marian, A. (2009, June). Beyond the stars: improving rating predictions using review text content. In Proceedings of *Twelfth International Workshop on the Web and Databases (WebDB 2009)*, (Vol. 9, pp. 1-6).

Ganu, G., Kakodkar, Y., & Marian, A. (2013). Improving the quality of predictions using textual information in online user reviews. *Information Systems*, *38*(1), 1-15.

Geron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media.

Gilbert, C. H. E., & Hutto, E. (2014, June). Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Proceeding of the Eighth International Conference on Weblogs and Social Media (ICWSM-14),* (Vol. 81, p. 82).

Ghose, A., & Ipeirotis, P. G. (2007, August). Designing novel review ranking systems: predicting the usefulness and impact of reviews. In *Proceedings of the Ninth International Conference on Electronic Commerce,* (pp. 303-310).

Gretzel, U., & Yoo, K. H. (2008). Use and impact of online travel reviews. *Information and Communication Technologies in Tourism 2008*, 35-46.

Ho, T. K. (1995, August). Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition (ICDAR),* (Vol. 1, pp. 278-282). IEEE.

Hu, M., & Liu, B. (2004, August). Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 168-177).

Hu, Y. H., & Chen, K. (2016). Predicting hotel review helpfulness: The impact of review visibility, and interaction between hotel stars and review ratings. *International Journal of Information Management*, *36*(6), 929-944.

Karpathy, A., Johnson, J., & Fei-Fei, L. (2015). Visualizing and understanding recurrent networks. *Oral presented at the 4th International Conference on Learning Representations Workshop.* San Juan, Puerto Rico.

Kiperwasser, E., & Goldberg, Y. (2016). Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, *4*, 313-327.

Luca, M. (2016). Reviews, reputation, and revenue: The case of Yelp. com *(March 15, 2016). Harvard Business School NOM Unit Working Paper*, (12-016).

LVCVA Research Center, Las Vegas Convention and Visitors Authority. (2020). *Historical Visitation Statistics: 1970-2019 A historical review of key Las Vegas tourism indicators from 1970 to present.* [Press release]. Retrieved October 6, 2020, from https://www.lvcva.com/research/visitor-statistics/

Ma, Y., Peng, H., & Cambria, E. (2018, February). Targeted Aspect-Based Sentiment Analysis via Embedding Commonsense Knowledge into an Attentive LSTM. *Proceedings of Association for the Advancement of Artificial Intelligence*, pp. 5876-5883.

McDonald, R., Lerman, K., & Pereira, F. (2006, June). Multilingual dependency analysis with a two-stage discriminative parser. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X),* pp. 216-220.

Murphy, R. (2019, December 11). Local Consumer Review Survey. Retrieved October 15, 2020, from https://www.brightlocal.com/research/local-consumer-review-survey/

Pang, B., & Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *Proceedings of the Association of Computational Linguistics.* pp. 115-124.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237. Association for Computational Linguistics.

Pennebaker, J. W., Boyd, R. L., Jordan, K., & Blackburn, K. (2015). *The development and psychometric properties of LIWC2015*. Austin, TX: University of Texas at Austin.

Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., Al-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B., De Clercq, O., Hoste, V., Apidianaki, M., Tannier, X., Loukachevitch, N., Kotelnikov, E., Bel, N., Jimenez-Zafra, S., Eryigit, G. (2016). SemEval-2016 Task 5: Aspect Based Sentiment Analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 19–30. Association for Computational Linguistics.

Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., Manandhar, S. (2014). SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)* (pp. 27–35). Association for Computational Linguistics.

Qiu, J., Liu, C., Li, Y., & Lin, Z. (2018). Leveraging sentiment analysis at the aspects level to predict ratings of reviews. *Information Sciences*, *451*, 295-309.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving Language Understanding by Generative Pre-Training. (n.d.). Retrieved November 5, 2020, from https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

Saeidi, S. (2016). SentiHood: Targeted Aspect Based Sentiment Analysis Dataset for Urban Neighbourhoods. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 1546–1556. The COLING 2016 Organizing Committee.

Snyder, B., & Barzilay, R. (2007, April). Multiple Aspect Ranking using the Good Grief Algorithm. In *Human Language Technologies 2007: The Conference of the North*

American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference, pp. 300-307.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013, October). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631-1642.

Strobl, C., Boulesteix, A., Zeileis, A., & Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, *8*(1), 25.

Sun, X. (2019). Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 380–385. Association for Computational Linguistics.

Tai, C., Socher, R., Manning, C. (2015). Improved Semantic Representations from Tree-Structured Long Short-Term Memory Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1556–1566. Association for Computational Linguistics.

Tang, T. (2016). Effective LSTMs for Target-Dependent Sentiment Classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 3298–3307. The COLING 2016 Organizing Committee.

Wang, Y., Huang, M., Zhu, X., & Zhao, L. (2016, November). Attention-based LSTM for aspect-level sentiment classification. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 606-615. Association for Computational Linguistics.

Wu, Z., Dai, X., Yin, C., Huang, S., Chen, H. (2018) Improving review representations with user attention and product attention for sentiment classification. Proceedings of the 32nd AAAI Conference on Artificial Intelligence, New Orleans, 2018. 5989–5995.

Xue, T., Li, T. (2018). Aspect Based Sentiment Analysis with Gated Convolutional Networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2514–2523. Association for Computational Linguistics.

Yu, M., Xue, M., & Ouyang, W. (2015). Restaurants Review Star Prediction for Yelp Dataset. Technical Report 17. University of California San Diego.

Zar, J. H. (1972). Significance testing of the Spearman rank correlation coefficient. *Journal of the American Statistical Association*, *67*(339), 578-580.