CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

The project methodology involves four steps which are data collection, data preprocessing, feature extraction, sentiment algorithm, and evaluation. For the data collection, the data will be downloaded from an open source. For data preprocessing, it is carried out to eliminate noise and address inconsistent data.. For feature extraction, it is carried out as word weighing. Lastly, for sentiment algorithm, it is carried out to classify the review as positive or negative.

3.1.1 Proposed Method

There are five steps that need to be followed to conduct comprehensive evaluations on the developed predictive model and build an interactive dashboard of sentiment analysis on hotel review using machine learning which are data collection, data preprocessing, feature extraction, sentiment algorithm and evaluation. Figure 3.1 shows all of the four steps.

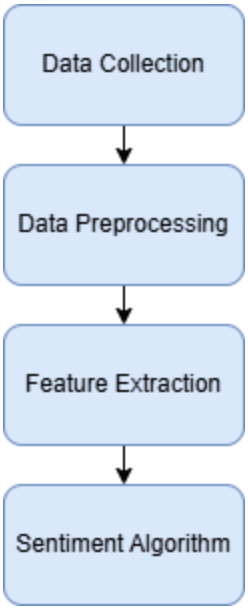

Figure 3.1 The steps of sentiment analysis

Phase	Objective	Activities
Data collection	• To identify the attributes of sentiment analysis on hotel review	• Download dataset from an open source
Data Preprocessing	• To eliminate noise and address inconsistent data.	• Normalization • Tokenization • Lemmatization
Feature extraction	• To evaluate the word weighing	• TF-IDF
Sentiment algorithm	• To classify the sentences as positive or negative	• Random Forest • Support Vector Machine
Evaluation	• To evaluate the accuracy of the sentiment algorithm	•

Table 3.1 The details of project development

3.2 Data Collection

Data collection of sentiment analysis of hotel reviews using machine learning is extracted from the open source which is Kaggle. The size of the dataset is 9,835,858 KB. It has

two attributes which are ID and Story. ID stands for the continuous number starting from one. Story stands from the sentence of reviews from the users.

3.3 Data Pre-processing

Data Pre-processing is a process of cleaning the raw data to become structured data. Online reviews are usually not clean and contain a lot of special characters, emoticons, URLs, hashtags and other text that are not necessary. Data pre-processing is one of the steps to ensure the high-quality data is used for the process of sentiment analysis. The process includes normalization, tokenization and lemmatization.

3.3.1 Normalization

Normalization is an important aspect in text analysis. It ensures that the data is consistent and comparable. Normalization can clean data in multiple ways. The procedure consists of converting all text to either all uppercase or all lowercase, eliminating punctuation, and transforming numbers into their written word forms. Apart from that, it can also remove redundant or duplicate data and handle missing values. For example, if there is a column with missing data, it can replace the missing data with 0 or the mean of the column.

Before Normalization	After Normalization
The Hotel is great. I give 4 stars and will come back again!!!!	the hotel is great i give four stars and will come back again
This hotel is AWESOME ❤	this hotel is awesome

Figure 3.2 Example after normalization

Figure 3.2 shows before and after normalization. The second sentence, “This hotel is awesome ❤” is raw data, and when it is normalized, the emoticon is removed, resulting in all the sentences becoming lowercase, “this hotel is awesome.”

3.3.2 Tokenization

Tokenization is the process of cutting the input string based on each compiler word. (Farisi et al., 2019) In a simple word, it separates the sentences into each word which is referred to as a token. It is a fundamental step in natural language processing (NLP).

Normalization	Tokenization
the hotel is great i give four stars and will come back again	['the', 'hotel', 'is', 'great', 'i', 'give', 'four', 'stars', 'and', 'will', 'come', 'back', 'again']
this hotel is awesome	['this', 'hotel', 'is', 'awesome']

Figure 3.3 Example of tokenization

Figure 3.3 shows how tokenization works. It is breaking down the sentences into words as a token. The second sentences “this hotel is awesome” is breaking down into four words which are ['this', 'hotel', 'is', 'awesome'].

3.3.3 Lemmatization

Lemmatization involves transforming a word into its base form for every word that has been tokenized. By employing lemmatization, every prefix and suffix is stripped from each word, converting them into their base forms to improve efficiency in text processing. For example, "running" will be transformed into "run". Examples of another word that will be taken out during lemmatization are “were”, “and”, “an”, “are” and others.

Original	Lemmatization
The geese are flying towards the mountains and running fast.	the goose fly towards the mountain run fast

Figure 3.4 Example of lemmatization

Figure 3.4 shows the lemmatization result. The raw sentence is “The geese are flying towards the mountains and running fast.” After lemmatization, the sentence becomes “the goose fly towards the mountain and run fast,” where flying and running are present participles that change to the base form fly and run.

3.4 Feature Extraction

Feature extraction is one of the essential steps after data pre-processing. Feature extraction has multiple techniques that can be used. In this project, TF-IDF will be used to calculate the weight of the sentence. TF-IDF is one of the most used techniques for text extraction. TF-IDF transforms the review text data into numbers. The steps in calculating the TF-IDF as follows:

1) 1) Determining the frequency of each word's Term Frequency (TF).

   - The number of Term Frequency (TF) is calculated by separating sentences into one word and each word is given a value of 1.

2) Calculating the frequency of documents (DF) for each word.

   - The document frequency (DF) is calculated by adding up the TF values for each word.

3) Determine the value of inverse document frequency (IDF).

$$IDF(w) = \log(\frac{N}{DF}) \quad (3.1)$$

Where N is the number of documents and DF is the number of documents containing the term t.

4) Determine the weight by multiplying the TF value with the IDF.

$$W_{ij= tf_{ij} \log (\frac{D}{df_j})} \quad (3.2)$$

For instance, consider a document that consists of 100 words, with the word “happy” occurring 10 times. In this case, the term frequency would be calculated as 10/100=0.1. Now, let's assume there are 50000 documents in total, and only 500 of those contain the word “happy.” Therefore, the IDF (happy) can be expressed as 50000/500=100, resulting in log(100) = 2. Consequently, the TF-IDF (happy) would be 0.1*2= 0.2

3.5 Sentiment Algorithm

The sentiment algorithm used for this project is Random Forest. It is an ensemble of decision tree algorithms that can be used for both classification and regression. In this algorithm generally, more trees correspond to better performance and efficiency. In a given training set, extract a sample set of data points by using the bootstrap method. After this construct a decision tree based on the output of the previous step. Apply the previous two steps and we will get the number of trees.