ASPECT-BASED SENTIMENT ANALYSIS ON HOTEL REVIEWS

CS4200 SENIOR PROJECT II

FINAL REPORT

William Sivutha Poch 5983122

Navin Singh 5935221

Jiadong Jiao 5918459

Submitted in Partial Fulfillment of the Requirements for

the Degree of Bachelor of Science in Computer Science

Semester 1/2020

ACKNOWLEDGEMENT

Our team would like to thank all those who have helped us for their support of this project. Otherwise, we can't push the project forward.

First of all, we are very grateful to our project advisor A. Kwankamol Nongpong. Under her constant guidance and supervision, our project has been effectively completed. It is thanks to her we have the necessary information about the project and the necessary data resources. At the same time, we are also very grateful to our committee members A. Tapanan Yeophantong and A. Pawut Satitsuksanoh as well as the intelligent systems laboratory for their assistance in this project.

Secondly, we would like to thank the members of the Assumption University for their encouragement and cooperation in our work, which helped us to complete the project.

In addition, we would like to thank our family and friends for their strong support and long-term encouragement.

We would like to express our sincere thanks to those who have worked with us to develop this project,

ABSTRACT

More and more companies and enterprises begin to attach importance to the user experience. They try to know the customers' views on their services or products through various ways, especially the hotel should attach great importance to the customer's feelings and comments. Today, most hotels have their own channels for customers to share their experiences. But if you encounter a large number of comments, it is very time and labor consuming just through human effort. Therefore, our team has developed a web application to analyze user comments using aspect-based sentiment analysis. In this project, the procedure is not limited to analyzing whether the overall comments are negative, positive or neutral but the sentiment analysis will be on finer details in which aspects the customers are expressing their attitudes on. As the domain of this work is hotel, we focus on detecting and analyzing sentiment on four aspects, which are service, amenities, conditions and the hotel cleanliness. The application provides visualization, which could help hotel managers to better understand the feedbacks from their customers.

Table of Contents

Chapter 1 Introduction 6

1.1 Rationale & Motivation 6

1.2 Background on Aspect-Based Sentiment Analytics 7

1.3 Project Goals & Scope of Work 8

Chapter 2 Project Framework & Methodology 10

2.1 Application Framework 10

2.2 Aspect Classification 11

2.3 Sentiment Classification 14

2.4 Analytics Module 15

Chapter 3 Project Implementation 16

3.1 Project Architecture 16

3.2 UI/UX Design & Storyboard 17

Chapter 4 Performance Evaluation 20

4.1 Project Dataset 20

4.2 Model Performance and Evaluation 21

Chapter 5 Conclusion 27

References 29

List of figures

Figure 2.1: Framework diagram 10
Figure 2.2: Text Preprocessing 12
Figure 2.3: Aspect Model Layers 13
Figure 2.4 Sentiment Model layers 15
Figure 3.1: Project architecture 16
Figure 3.2 Visualization dashboard 18
Figure 3.3: Reviews and sentiments 19
Figure 3.4: Program Test Page 19
Table 4.1 Sentiment and aspect distribution 20
Table 4.2: Experiment results of various deep learning aspect models 21
Table 4.3 Multilabel Confusion Matrix 22
Table 4.4 Aspect ground truths 23
Table 4.5 Experiment results of various deep learning sentiment models. 24
Figure 4.6 Sentiment confusion matrix 25
Table 4.7 Sentiment model ground truths 26

Chapter 1

Introduction

People commonly leave reviews online about the hotels they have stayed at to share their experiences and opinions. However, it is time consuming and tedious to search through and extract meaningful information from all the reviews. With our web application it becomes possible to extract sentiments and aspects from hotel reviews.

1.1 Rationale & Motivation

The hotel industry is one that must always take into account what exactly it is that their guests enjoy or dislike about the hotel and leaving a review is a common way guests give feedback. Whether they recommend that particular place or advise people to stay away. But with thousands to hundreds of thousands of reviews it becomes tedious and difficult, if not impossible, for a person to sort through and extract key information from these reviews. A solution to this is to create a web application which uses sentiment analysis to analyze and predict sentiment from reviews. However, typical sentiment analysis methods only focus on classifying text into positive or negative sentiments regardless of its length or of its contents [1]. The issue with this is that normal text may contain more than one aspect or entity within it. In the case of a hotel review, that can mean a customer might write a review where they strongly complain about the staff or service but complement the hotel on how clean it is. Normal sentiment analysis methods might only extract a negative sentiment which neglects the positive aspect of the review. And so in our project we explore aspect-based sentiment analysis to extract aspects and then predict sentiments, and then create a web application to display this information.

1.2 Background on Aspect-Based Sentiment Analytics

Aspect-based sentiment analysis is a form of sentiment analysis, one that focuses on extracting the most important aspects and then predicting the polarity of each aspect from the text [2]. Sentiment analysis techniques can be categorized into 3 main categories: knowledge-based techniques, statistical methods, and hybrid approaches. Knowledge-based techniques utilize human defined rules such as a list of opinion words to identify polarity. Statistical methods utilize machine learning techniques to extract features and classify polarity from text. Hybrid approaches combine elements from knowledge-based techniques and statistical methods, using grammatical heuristics and combining these with machine learning allows them to detect subtler details and semantics within text [3, 4].

Aspect detection techniques can be grouped into the same categories as sentiment analysis: knowledge-based techniques, statistical methods, and hybrid approaches. Knowledge-based techniques typically employ heuristic rules to create lists of aspect candidates from an unlabeled dataset. Statistical methods utilize labeled datasets and classification methods in machine learning and more advanced techniques involve calculating co-occurrences of features to extract and classify aspects. Hybrid approaches, again, combine these two techniques to create a more precise method [5, 6, 7].

Aspect-based sentiment analysis combines both sentiment analysis with aspect detection creating two main parts, an aspect detector and a sentiment classifier. Each part utilizes one of the aforementioned techniques. For instance, one proposed model utilizes statistical methods (deep learning) in both aspect detection and sentiment classification [1], while another proposed model uses unsupervised knowledge-based techniques [7]. In short, the aspect detector breaks down the text into aspects and the sentiment classifier classifies the text at its aspect level rather than document or sentence level [1, 2]. Currently, there a number of methods when it comes to aspect-based sentiment analysis categorized into language rule, sequential, topic modelling, deep learning, and hybrid methods [2].

In recent years, there has been great interest and growth in this field of research. The reason behind the spark of interest and growth is due to businesses trying to predict the nature of opinions and product preferences from consumers so they can adapt strategies and marketing campaigns. Typical sentiment analysis utilizes document or sentence level to predict polarity but these do not provide enough information on preferences or opinions. Aspect level can provide much more information [1, 2]. By breaking down a sentence into the aspects, we can examine what entities or aspects a user mentions and then from there summarize the users' opinions on them. This allows businesses and companies to clearly see subtleties of entities and aspects making it much easier to determine consumers like and dislike. Information such as this would assist companies in improving their products or services. However, despite advancements and interest in this newer field of sentiment analysis, there is still a long way for it to go [8].

1.3 Project Goals & Scope of Work

The objectives and scope of our project are as follows:

1. Create and train a model to classify hotel reviews into 4 aspects; Hotel staff, amenities, condition, and cleanliness.

2. Create and train a model to classify each aspect and the overall sentiment into 3 polarities; Positive, negative, and neutral.
3. The user can view data analytics on the website.
4. The user can input their own reviews by typing or uploading a csv file.
5. There are features for the user to sort and explore the data analytics.

The main focus of this project is to implement an aspect-based sentiment analysis model and test it using a web application. This project is a part of a proof of concept for Infosearch [9]. Infosearch is one of the top market research and consultant companies in Thailand. In this project we explore using aspect-based sentiment analysis and its feasibility along with a web application. This project uses a hotel review dataset from Datafiniti [10] and Nasket, which was provided by Infosearch, that we have manually labeled ourselves into 4 aspect categories and their sentiments. The 4 aspect categories: Staff, amenities, condition, and cleanliness were also set by Infosearch.

Chapter 2

Project Framework & Methodology

To create the aspect-based sentiment analysis system, there are 2 main parts that must be created. The aspect detector and the sentiment classifier. The application's framework along with these 2 main parts will be outlined in the next sections followed by a brief overview of the analytics module of the web application.

2.1 Application Framework

The figure 2.1 below shows the framework of the application from frontend to backend and then back. The process of aspect-based sentiment analysis is described below as well.

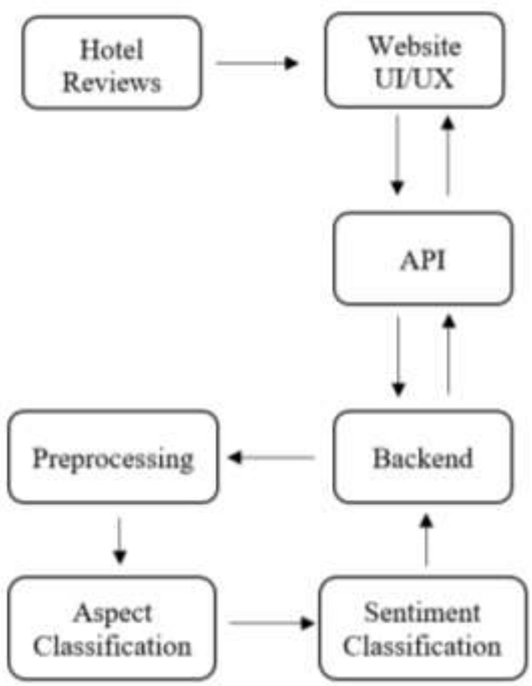

Figure 2.1: Framework diagram

First, the user inputs or uploads reviews through the website. The user can choose to upload a csv file containing review text data or manually type in some sentences for testing. The data is stored in Firebase where the website is also hosted. From there, the API retrieves the data from Firebase and passes it to the backend.

Once at the backend, the text is first preprocessed and split into sentences, then sent to the next parts. The aspect model will classify each sentence into aspects; Any combination of staff, amenities, condition, and cleanliness or no aspect at all. After which, any aspects obtained are added to their sentence and then passed to the next step. The sentiment model will then classify the sentence’s aspect sentiments as well as the overall sentiment. Once all the sentences have had their aspects and sentiments classified, the data is aggregated and sent through the API to Firebase and updates the database. Finally, the website will display the data.

2.2 Aspect Classification

Figure 2.2 gives an overview of how text from a review is preprocessed before it is sent to a model for prediction and classification. Below, we discuss each step of the preprocessing.

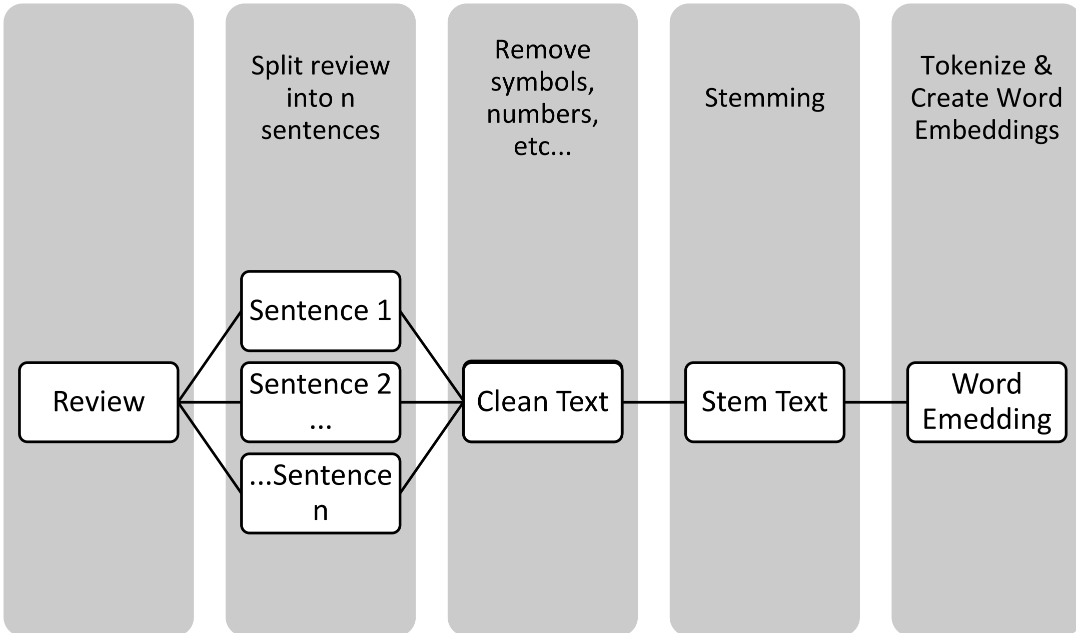

Figure 2.2: Text Preprocessing

Before moving on to the aspect model, the review text must first be preprocessed. In this project, we focus on simple preprocessing techniques. Since the text is in the form of a review, it can contain many sentences, so the reviews are split into individual sentences using the natural language processor spaCy [11]. spaCy provides pre-trained models for the English language which can be used to automatically segment text into sentences. The reviews are split into sentences to get more fine grain details in the reviews. Then the text is cleaned of numbers, symbols, html or http links, and then all text is set to lowercase. Cleaning the text in such a way reduces the amount of noise, irregularities, and irrelevant things in reviews which helps put more focus on words which can convey meaning.

Next, the text undergoes stemming. Stemming is the process of reducing a word to its root or base form [12]. Instead of having multiple words that differ only due to affixes, stemming removes the affixes reducing the word to its base form. This reduces the overall vocabulary size and refines the results. For this step we use the Natural Language Toolkit (NLTK) Porter Stemmer [13] to stem every word in each sentence. The last step is to create word embeddings. Word embeddings represent words as dense vectors and maps semantic meaning into a geometric space also known as embedding space [14]. To create word embeddings, the sentences are tokenized and converted to integer sequences, and then padded with zeros using a Keras Tokenizer. This tokenizer was fit on our dataset with a limit of top 5000 words. Once word embedding's are created for all the sentences, the preprocessing steps are complete.

Below in figure 2.3 shows the layers for our aspect model. The model was created using Keras library. The model consists of an embedding input layer and a layer to flatten and then the output dense layer. The activation function is sigmoid for the output layer, with the optimizer adam and loss function categorical_crossentropy. These layers and their hyper parameters were chosen from our own testing with different model layouts, which is discussed later on, and based on research and testing of other proposed models [1, 8].

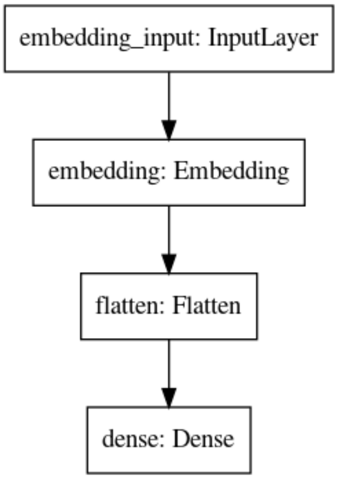

Figure 2.3: Aspect Model Layers

The sentence word embedding's from before are passed to the model for prediction. This results in a probability distribution over the 4 aspects. Since more than one class may occur, the probabilities must be filtered through a threshold. The threshold used in this project is an arbitrary value of 0.3, selected after testing. Testing was performed by setting the threshold to values between 0 and 1 and comparing their results against ground truths. The value of 0.3 produced the best results and so another test was done on values between 0 and 0.3 but the results did not improve beyond this threshold. This means that any class with a probability above 0.3 will be kept and any below will be dropped.

2.3 Sentiment Classification

Since all the reviews have already been preprocessed and their aspects extracted, we can simply pass the preprocessed text into the sentiment model. The sentiment model is a convolutional neural network (CNN) which was selected after testing various other proposed models, the results of this testing is discussed later. The model consists of an embedding input layer, followed by a convolutional 1D layer. Then a global max pooling 1D layer, 1 dense layer, then finally the output dense layer. This model uses relu for the activation function in hidden layers and softmax in the output layer. The optimizer adam is again used here. Each sentence's word embeddings are fed into the sentiment model, which outputs the probability distribution of the sentiments. Since we only need one sentiment, we select the sentiment with the highest probability. This is done for each aspect that the sentence contains, from the aspect model, and the predicted sentiment is used for the overall sentiment. If the sentence does not contain any aspect, the same sentiment model simply predicts only the overall sentiment of the sentence. The figure 2.4 below shows the sentiment model layout.

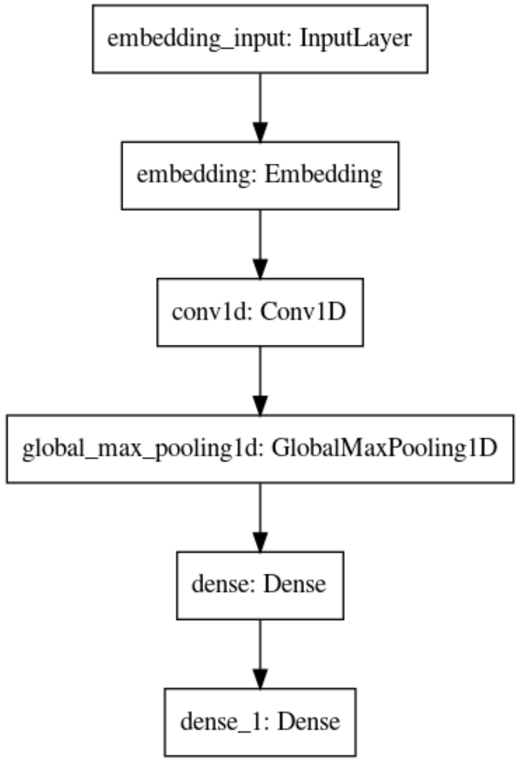

Figure 2.4 Sentiment Model layers

2.4 Analytics Module

Data collected from the aspect-based sentiment analysis is sent to a Firebase storage, where the API will retrieve and pass it to the website. Anytime a user uploads a file or inputs their reviews, the data will first be stored in Firebase, then the API will retrieve the data and pass it to the program, where it undergoes analysis and predictions. The resulting predictions are aggregated and then sent back to Firebase via the API. Once in Firebase the website can use the data and display the analytics. The website will display the total numbers of positive, neutral, and negative sentiments as well as the total sentiments for each of the aspects. As well as provide a more detailed breakdown of each sentence describing the sentence's overall sentiment, aspect, and aspect sentiment.

Chapter 3

Project Implementation

In the following sections of this chapter, we discuss the project's architecture and the overall flow and process of the web application. From running the program and uploading a hotel review dataset to viewing the results on the web application. The UI/UX of the web application is also depicted and its functionality is described.

3.1 Project Architecture

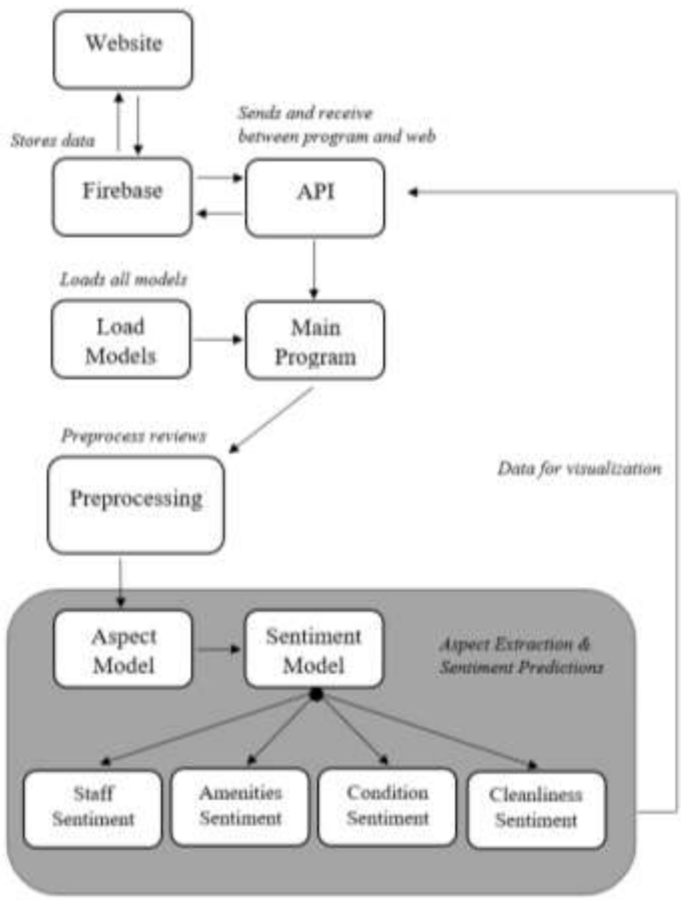

Figure 3.1: Project architecture

Figure 3.1 shows the architecture of the project. Once the main program is run, it first loads all models by reading the model from a json file and then loading the model's weights from an h5 file. Then the program is ready to receive any text from the API. When a file is passed from the website to API then to the program, it loads it into a dataframe and passes it to the preprocess function. In preprocessing, the reviews are split into sentences using the spaCy library. Then the text is cleaned of symbols, numbers, https and html tags then finally the text is converted to its stem or base form using Porter Stemmer. Now the text is sent to the aspect model first, where it is analyzed and the aspect is extracted. The text is then passed to the sentiment model, where the sentiment of any aspect it contains is predicted as well as the overall sentiment of the text. Once all the text and their aspects and sentiments predicted, it is aggregated and then the API will send the data to the firebase database. Once there, the data can be viewed through the web application. If data has already been uploaded, then the user may input their own review sentences in the program testing page.

3.2 UI/UX Design & Storyboard

The web application features 3 main pages, a visualization page, a details page, and a testing page with an additional page for uploading a csv file of reviews. The figures 3.2 to 3.4 below depict the 3 main pages and describe their functions in detail.

Figure 3.2 Visualization dashboard

Figure 3.2 shows the visual page of hotel review analysis. The first histogram is a general overview of the forecast results of all hotel reviews, which includes how many negative, positive or neutral comments are contained in the text data as a whole, and emotional information in each aspect. The pie charts show the proportion of three sentiments in the whole and each aspect.

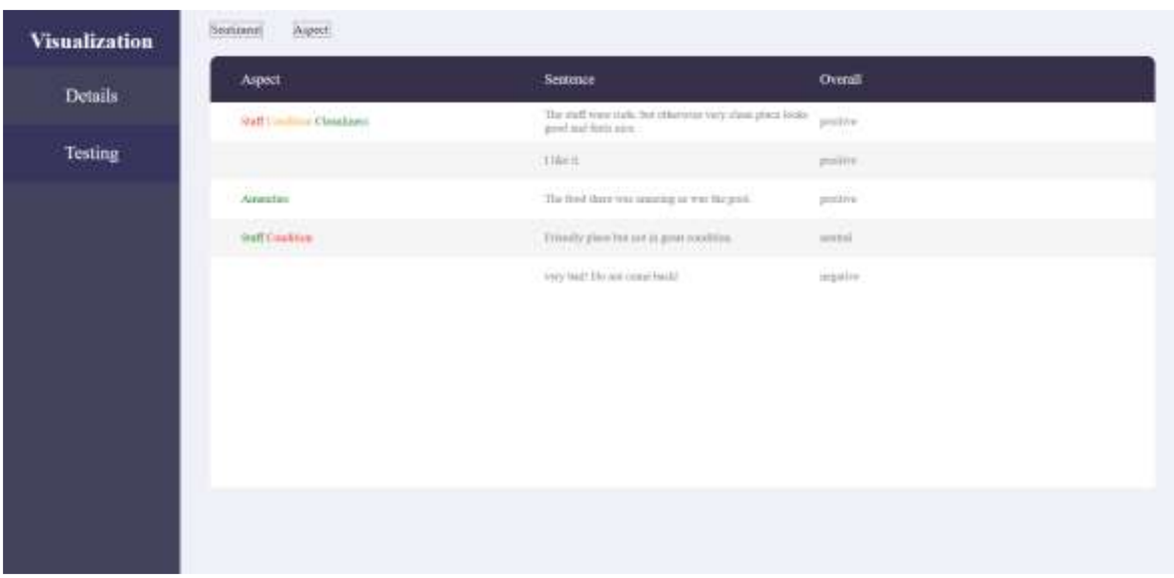

Figure 3.3: Reviews and sentiments

In addition, the predicted comment text can be viewed on the detail page shown in figure 3.3. Each review has been divided into sentences. Each row in the table describes a sentence and its overall sentiment as well as any aspect mentioned and that aspects sentiment. The mentioned aspect is shown the on left in a color denoting the sentiment. Green for positive, orange for neutral, and red for negative sentiment.

Figure 3.4: Program Test Page

Furthermore, the web application features a testing page as shown in figure 3.4. Users can enter the sentence that they want to predict in the text input box. After clicking the submit button, it will be handed over to the backend for prediction. Afterwards, the prediction results will update the pie charts below.

Chapter 4

Performance Evaluation

4.1 Project Dataset

Originally the dataset that was used was Nasket dataset of hotel reviews. This dataset featured 1357 reviews split into 6306 sentences. These reviews were labeled using columns for overall sentiment and mentioned aspects and their sentiments. To increase our dataset size, another dataset from Datafiniti, which can be found on Kaggle [10], was included. Combining these datasets provided us with more robust data with more variance in reviews. The Datafiniti dataset in total contains 20,000 hotel reviews from 1,000 different hotels. Of these 20,000 we labeled a total of 1992 reviews, which when split into sentences amounted to a total of 11,173 sentences. Combining both datasets provided us with 14,585 sentences for training and testing. For our training and testing of our models, both Nasket and Datafiniti were used. The table 4.1 below shows the distribution sentiments and aspects.

Table 4.1 Sentiment and aspect distribution

	Overall	Staff	Amenities	Condition	Cleanliness
Positive	8,215	1,330	782	237	660
Negative	2,945	300	324	238	118
Neutral	3,425	96	250	79	36
Total:	14,585	1,727	1,356	554	814

Out of a total of 14585 sentences there are 8215 overall positive, 3425 overall neutral, and 2945 overall negative sentiments. Staff aspect amounted to 1727 sentences with 1330 positive, 300 negative, and 96 neutral sentiments within. Amenities aspect contained 1356 sentences with 782 positive, 324 negative, and 250 neutral sentiments. Condition aspect contained 554 sentences with 237 positive, 238 negative, and 79 neutral sentiments. And lastly, cleanliness aspect totaled 814 sentences with 660 positive, 118 negative, and 36 neutral sentiments.

4.2 Model Performance and Evaluation

Table 4.2 showcases the comparison between a few of the experimental aspect models that we tested and the current model used in this work. Classes 0-3 denotes the aspects staff, amenities, condition, and cleanliness respectively. The first model describes a simplistic neural network model. Whilst GloVe Embedding + CNN LSTM model describes a complex neural network model that uses a pre-trained word embedding from Stanford NLP [15]. The current model is in between these two models in terms of complexity.

Table 4.2: Experiment results of various deep learning aspect models

	Embedding + CNN			**GloVe Embedding + CNN LSTM**			**Current model**		
Class	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
0	97%	72%	82%	96%	41%	57%	86%	88%	87%
1	91%	71%	79%	34%	81%	48%	77%	90%	83%
2	73%	58%	64%	12%	62%	20%	68%	74%	71%
3	97%	57%	72%	22%	84%	35%	88%	86%	87%
Macro avg.	91%	67%	77%	41%	67%	40%	81%	86%	83%
Micro avg.	89%	64%	74%	82%	49%	54%	84%	85%	83%

The results of precision, recall and f1 score were obtained from using scikit-learn’s classification report [16] which supports multilabels. From the comparison, the current model performs much better across all classes overall than the other two models, which have the most trouble on classes 1 (amenities), 2 (condition) and 3 (cleanliness) although some class 2 still has low scores. The reason as to why condition performed much lower is from the vagueness of the condition aspect. Other aspects can be categorized using easily distinguishable terms while the condition aspect has terms which are less apparent. The aspect category of condition itself is quite vague, the condition of a hotel could be anything from the age of the hotel or if there's anything broken or damaged in the hotel room. As a result, this aspect has lower scores. The table 4.3 below shows the mutlilabel confusion matrix for the aspect model.

Table 4.3 Multilabel Confusion Matrix

Class	**TP**	**FP**	**FN**	**TN**
Staff	647	217	142	5,243
Amenities	556	433	56	5,204
Condition	209	25	105	5,910
Cleanliness	285	8	170	5,786

From the table we can see the total amount of true positives of each class, as well as the false positives, false negatives, and true negatives. This gives some more insight into each class's prediction results.

Table 4.4 gives a closer look at the results of the model by comparing the ground truths with the predicted aspects labels. It shows some examples out of the 6,249 sentences and their predictions compared to ground truths.

Table 4.4 Aspect ground truths

Text	Predicted	Ground Truth
I did have a wonderful massage and it was a great value .	Amenities	Amenities
If you would like to have the massage of your life , go see Elvira !	Amenities	Amenities
the service in the restaurants was very slow	Staff	Staff
room was clean and comfortable	Cleanliness	Cleanliness
She is awesome .		
Stayed here because rooms were blocked for a wedding .	Condition	
i found the casino to be very dreary and outdated	Condition	Condition
The room itself was spacious , clean and well priced .	Condition, Cleanliness	Cleanliness
The service in the restaurants was very slow .	Staff	Staff
They definitely need to move the air around better .		
…	…	…
Total sentences:		6249 sentences
Hamming loss = 0.046247399		

Since the aspects are multilabeled, a better measure than accuracy to evaluate the results is to the hamming loss function, which shows the fraction of labels that are incorrectly predicted. Instead of penalizing an entire set of labels if one label is incorrect, hamming loss only penalizes individual labels giving a more accurate representation of correct vs incorrect labels [17]. The hamming loss value of 0.046 represents the represents the fraction of incorrect labels to the total number of labels, this means out of all the labels around 4% of the labels were incorrectly predicted. This metric is a much better indicator of performance than accuracy as it considers each individual label rather than each set of labels. To get a better understanding of which aspects are being misclassified we can take a look at the confusion matrix. The figure

Table 4.5 showcases the comparison between the model experiments for sentiment analysis and the current model. Classes 0-2 denote the sentiment classes of positive, neutral and negative respectively. Similar to the previous experiment, the first model is a shallow neural network model and the CNN LSTM model is a more complex neural network model. The current model falling in between the two in terms of complexity.

Table 4.5 Experiment results of various deep learning sentiment models.

	Shallow NN			**Embedding + CNN LSTM**			**Current model**		
Class	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
0	85%	85%	84%	84%	82%	83%	85%	80%	82%
1	53%	52%	53%	49%	52%	51%	67%	78%	72%
2	62%	60%	61%	61%	61%	61%	69%	53%	60%
Macro avg.	66%	66%	66%	65%	65%	65%	74%	70%	71%
Micro avg.	71%	71%	71%	70%	70%	70%	75%	75%	75%

The results of these experiments show that high model complexity produces somewhat worse results than simpler models. The current model has better results overall across all the classes. The current model f1 scores for each sentiment are higher and more consistent than in the other models. Additionally, the figure 4.6 below shows the confusion matrix of the sentiments after prediction compared to actual sentiments. This gives a better look at which sentiments classify correctly and which ones are falsely labeled.

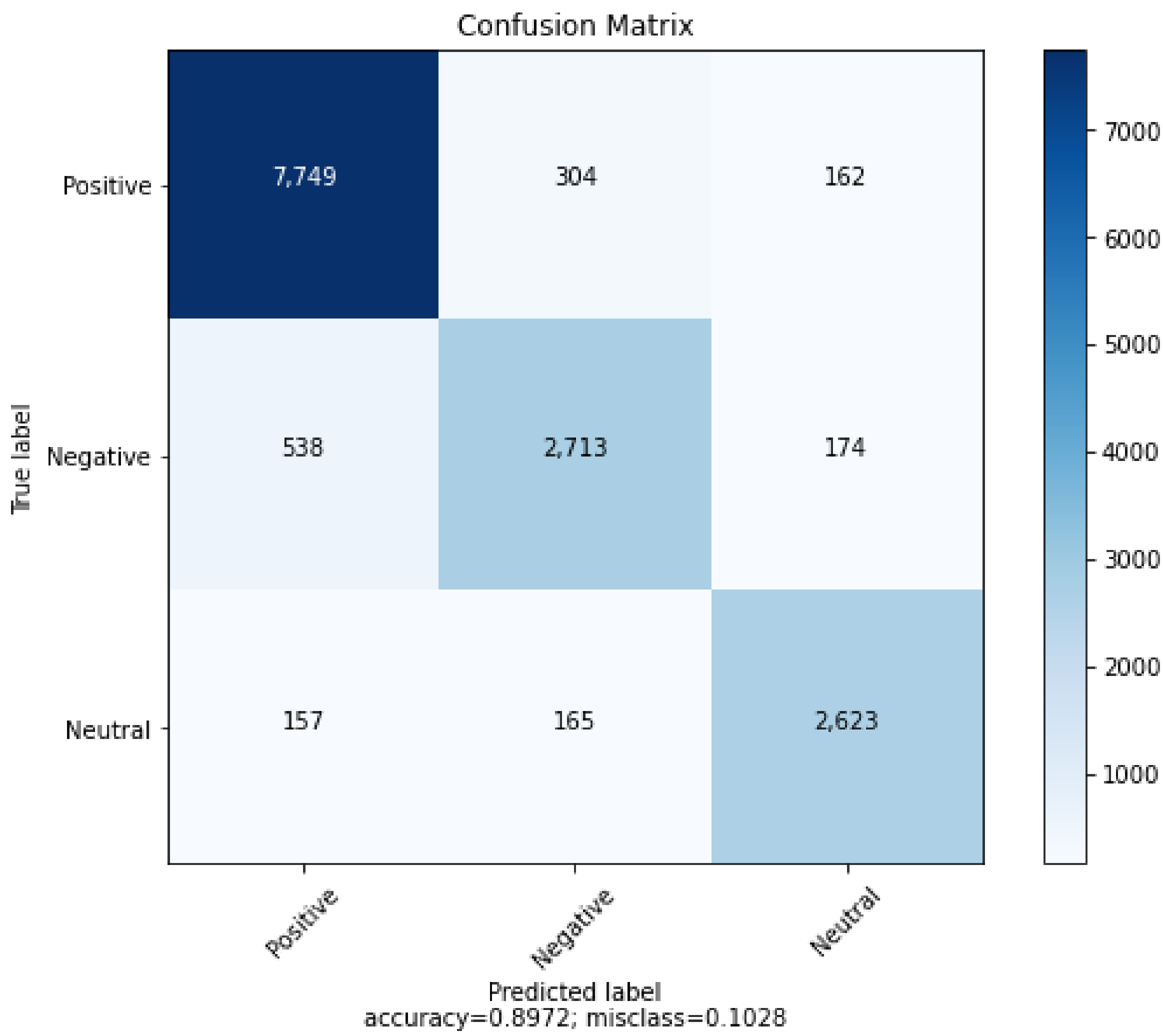

Figure 4.6 Sentiment confusion matrix

From the figure above, we can see a large portion of the negative class predicted as positive instead. As well as the 304 sentences of positive sentiment which falsely predicted as negative sentiment.

The table 4.7 below shows some examples of sentences and their predicted sentiment compared to the ground truth. In total, 14,585 sentences were tested resulting in 13,085 correct predictions.

Table 4.7 Sentiment model ground truths

Sentence	Predicted	Actual
best romantic vacation ever	Positive	Positive
Hampton inn Vancouver wa	Neutral	Neutral
king bed in too small space so that one side did not even have a lamp	Negative	Negative
after getting the bait and switch I decided I'd rather stay anywhere else	Negative	Negative
pool but no hot tub	Negative	Neutral
will definately stay there again	Negative	Positive
i cant comment on the breakfast because we slept through it but everything else was excellent	Neutral	Positive

Chapter 5

Conclusion

In this project we developed a web application and an aspect-based sentiment analysis system which allows users to input hotel reviews for analysis. The web application presents a user interface for us to showcase and perform testing of the aspect-based sentiment analysis model. The reviews are analyzed using deep learning models which combine aspect prediction and sentiment prediction to create an aspect-based sentiment analysis system. Our project breaks down reviews into aspect levels and then predicts individual aspect polarity, providing a more detailed look at what users express and feel in their reviews. This can allow for the user to easily analyze and view their hotel's reviews without having to go through the tedious task of manually reading and analyzing customer reviews themselves. The aspect-based sentiment analysis can provide deeper insight than typical sentiment analysis methods. The models were evaluated by comparing them against the ground truths of our labeled dataset. The performance of the models is compared to the various experiments that we had conducted and it shows that the models we used performed the best out of our other experiments.

To further our project, the web application can be improved and enhanced to include new categories for aspects, add new options and visualizations for data as well as improving the overall flow and design. For the aspect-based sentiment analysis models, they can both be further improved upon by training more and more data. This would allow them to be continuously improved becoming more robust and precise in their predictions. And rather than only focusing on hotel reviews, the system can be trained to analyze different categories of reviews. The current model is designed to detect predetermined aspects rather than automatically detecting new aspects. To do so may require the use of unsupervised learning or the other techniques used for aspect-based sentiment analysis. Other methods of aspect-based sentiment analysis can also be explored such as topic modelling or hybrid approaches. Given enough data on different categories, and continuous learning algorithms, it would be possible to create a robust system which can detect aspects and predict sentiments from all kinds of reviews.

References

[1] M. L. Bo Wang, "Deep Learning for Aspect-Based Sentiment Analysis," Stanford, [Online]. Available: https://cs224d.stanford.edu/reports/WangBo.pdf. [Accessed 2020].

[2] R. A. Z. S. Madhoush Z, "Aspect-Based Sentiment Analysis Methods in Recent Years," June 2019. [Online]. Available: https://www.researchgate.net/publication/335421201_Aspect-Based_Sentiment_Analysis_Methods_in_Recent_Years. [Accessed 2020].

[3] "Sentiment Analysis: A Definitive Guide," monkeylearn, [Online]. Available: https://monkeylearn.com/sentiment-analysis/. [Accessed 2020].

[4] A. H. H. K. Walaa Medhat, "Sentiment analysis algorithms and applications: A survey," 2014.

[5] W. O. F. F. D. R. Schouten Kim, "Supervised and Unsupervised Aspect Category Detection for Sentiment Analysis With Co-Occurrence Data," 2017.

[6] S. Byrne, "Aspect Identification and Sentiment Analysis in Text-Based," Lehigh University, 2017.

[7] M. S. F. d. J. Ayoub Bagheri, "An Unsupervised Aspect Detection Model," Springer, Berlin, Heidelberg, 2013.

[8] F. Pascual, "A Comprehensive Guide to Aspect-based Sentiment Analysis," monkeylearn, 2019. [Online]. Available: https://monkeylearn.com/blog/aspect-based-sentiment-analysis/. [Accessed 2020].

[9] "Infosearch," Infosearch, [Online]. Available: http://www.infosearch.co.th/. [Accessed 2020].

[10] "Datafiniti Hotel Reviews," [Online]. Available: https://www.kaggle.com/datafiniti/hotel-reviews. [Accessed 2020].

[11] "spaCy," [Online]. Available: https://spacy.io/. [Accessed 2020].

[12] Standford, "Standford NLP - Stemming and Lemmatization," [Online]. Available: https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html. [Accessed 2020].

[13] "NLTK," [Online]. Available: https://www.nltk.org/. [Accessed 2020].

[14] Wikipedia, "Word embeddings," [Online]. Available: https://en.wikipedia.org/wiki/Word_embedding. [Accessed 2020].

[15] R. S. C. D. M. Jeffrey Pennington, "GloVe: Global Vectors for Word Representation," Stanford, [Online]. Available: https://nlp.stanford.edu/projects/glove/. [Accessed 2020].

[16] Scikit-learn, "sklearn.metrics.classification_report," scikit-learn, [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html. [Accessed 2020].

[17] Scikit-learn, "sklearn.metrics.hamming_loss," scikit-learn, [Online]. Available: scikit-learn.org/stable/modules/generated/sklearn.metrics.hamming_loss.html. [Accessed 2020].

[18] W. Poch, "Aspect-based sentiment analysis," Github, 2020. [Online]. Available: https://github.com/WilliamPoch/AspectBasedSentimentAnalysis.