Analysis of Hotel Trip Reviews Dataset

Paige Berrigan
Student Number: 1290283

Fanshawe College School of Information Technology

INFO 6148 | Natural Language Processing

October 14, 2024

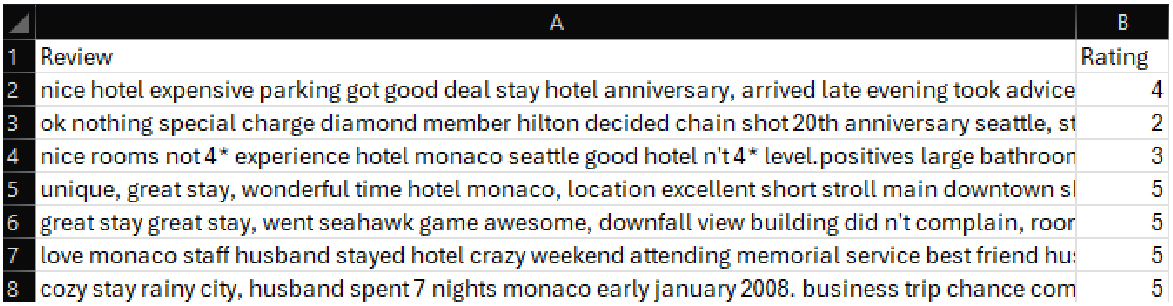
Contents

1. Introduction

This project involves the analysis of the "Hotel Trip Reviews" dataset sourced from Kaggle [1] The main objective is to apply various Natural Language Processing (NLP) techniques from the SpaCy library to clean, visualize, and analyze the reviews, extracting meaningful insights. Additionally, the project will compare different word embedding methods and evaluate their effectiveness in analyzing the textual data.

2. Definition of Dataset

The dataset utilized in this project is the “Tripadvisor Hotel Reviews” dataset, sourced from Kaggle. It consists of 20,494 rows and two columns. The "Review" column contains textual feedback from users regarding various hotels listed on TripAdvisor, while the "Rating" column provides a corresponding score on a 1-5 scale. For the purposes of this project, only the "Review" column is utilized for text cleaning, visualization, and analysis, while the "Rating" column is used for a basic logistic regression classification task, serving as the assigned labels.



	A	B
1	Review	Rating
2	nice hotel expensive parking got good deal stay hotel anniversary, arrived late evening took advice	4
3	ok nothing special charge diamond member hilton decided chain shot 20th anniversary seattle, st	2
4	nice rooms not 4* experience hotel monaco seattle good hotel n't 4* level.positives large bathroon	3
5	unique, great stay, wonderful time hotel monaco, location excellent short stroll main downtown sl	5
6	great stay great stay, went seahawk game awesome, downfall view building did n't complain, roor	5
7	love monaco staff husband stayed hotel crazy weekend attending memorial service best friend hu	5
8	cozy stay rainy city, husband spent 7 nights monaco early january 2008. business trip chance com	5

FIGURE 1 : RAW TABLE DISPLAYING A SAMPLE OF HOTEL REVIEWS AND CORRESPONDING RATINGS

[1] Kaggle. (2024). *Hotel reviews dataset*. Retrieved from https://www.kaggle.com/datasets/joebeachcapital/hotel-reviews/discussion?sort=hotness

To improve efficiency, only 5,000 reviews from the original dataset were processed. The code loads both the full dataset, which includes both the "Review" and "Rating" columns as *full_df* for potential future sentiment analysis or other applications, and a *reviews_df*, which contains only the "Review" column.

3. Visualization of Dataset Features

To analyze and visually represent the features in this dataset, I used the cleaned data (refer to **Part 4** for more details) and created three visualizations that show key features of the dataset. These visualizations were selected because they provide different perspectives on the data to best understand the patterns and terms found within the hotel reviews.

A. Word Cloud Visualization

This visualization shows the lemmas that occur most frequently throughout the 5000 hotel reviews searched. The larger a word appears, the more frequently it appears in the dataset. This is a great visualization as at a glace an onlooker can quickly get an understanding of the general themes of the reviews.

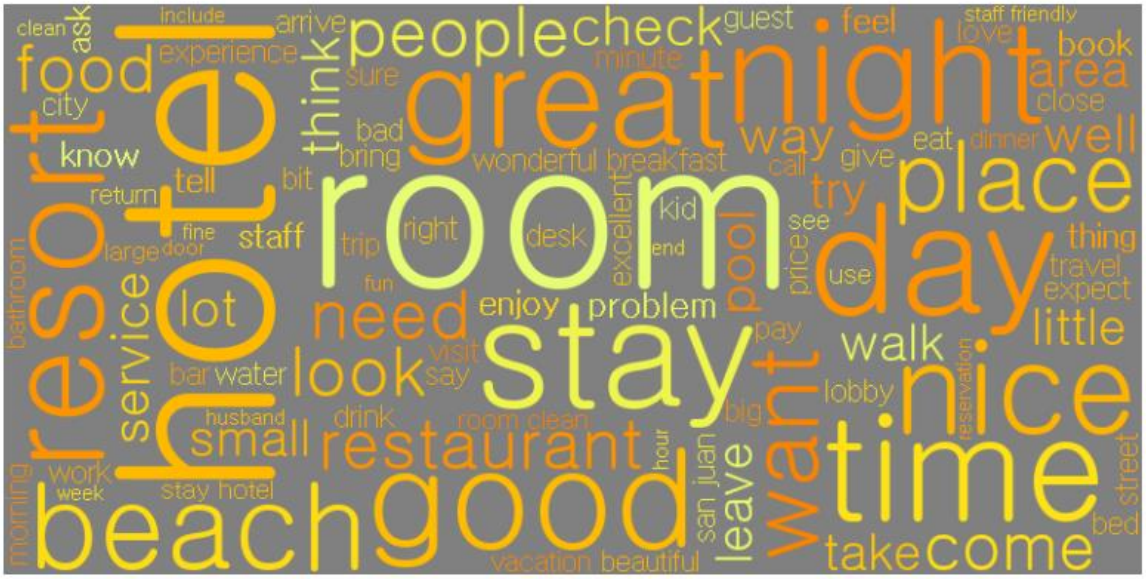


FIGURE 2: WORD CLOUD OF MOST FREQUENTLY USED LEMMAS IN TRIPADVISOR REVIEWS

From **Figure 2** above, it is unsurprising that some of the most frequently occurring words include "hotel," "room," and "stay." However, the word cloud also reveals a mixture of words with both negative and positive connotations. While words like "small" and "bad" suggest negative experiences, there are far more positive terms such as "good," "nice," "beautiful," and "friendly." It is important to note that these positive words could be used in negative contexts, such as in phrases like "not beautiful" or "not friendly," which would alter their sentiment. Despite this possibility, the overall impression from the dataset suggests that most reviews lean toward the positive side. This work could have been expanded on by creating word clouds for only 5 star reviews, and comparing them to words found in 1 star reviews.

B. Bar Plot of Most Frequently used Words

Taking a more quantitative approach to the information displayed in the word cloud, a bar chart was created to show the most frequently used words in a more structured and manner. The top 25 words were selected for this visual representation, as this number provides a balanced overview of the most common terms while highlighting words with significant frequency. By displaying the exact counts of each word, this chart allows for clearer comparisons and helps in identifying key themes within the reviews, reinforcing insights about the dataset's content and focus. This information is shown in **Figure 3.**

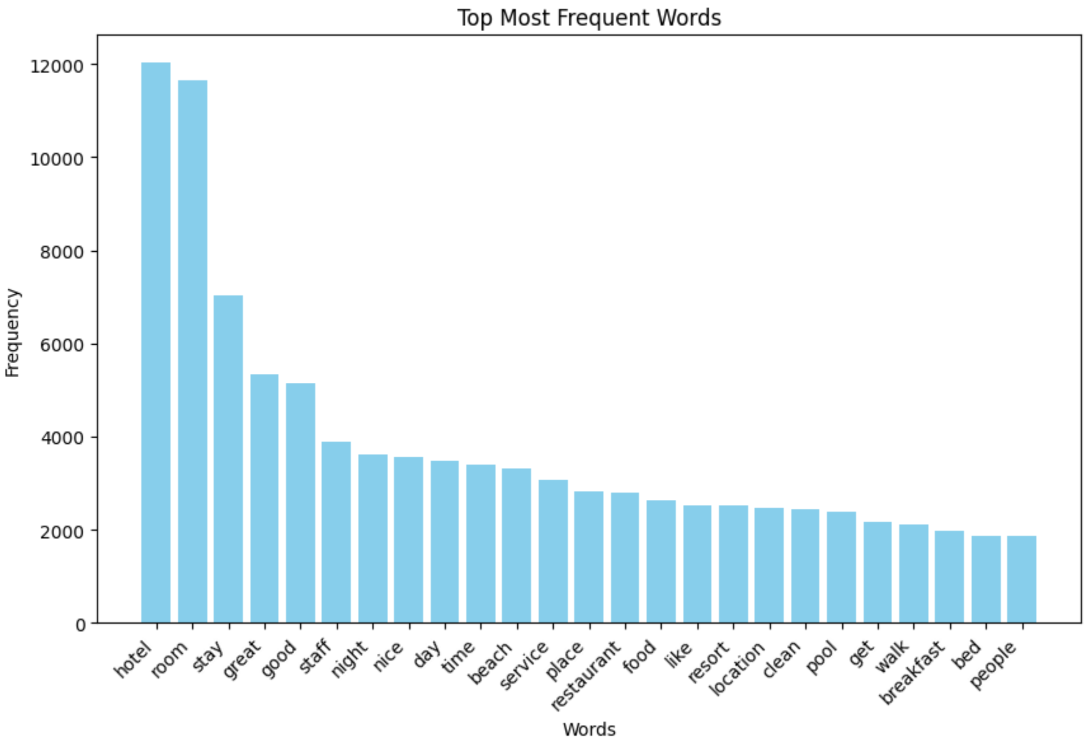


FIGURE 3: MOST FREQUENTLY USED WORDS IN 5000 TRIPADVISOR REVIEWS

What is interesting about this plot is that it offers qualitative insight into word frequency across the dataset. Despite analyzing only 5,000 reviews, certain words appear over 5,000 times, for example, the top word "hotel" is mentioned approximately 12,000 times within these 5,000 reviews. This shows that, on average, the word "hotel" is used more than once per review on TripAdvisor.

C. Distribution Pie Chart

Finally, A pie chart was created that visualized the top 5 most frequently used word. The pie chart illustrates the distribution of reviews based on how many of the top 5 words they contain. The largest portion, 34.5%, represents reviews that contain 4 of the top 5 words, followed by 30.1% of reviews containing 3 of these words. Reviews with 2 words make up 14.5%, while 16.5% of reviews contain all 5 words. A smaller percentage of reviews contain only 1 word (3.6%), and just 0.7% contain none of the top 5 words. This distribution highlights that the majority of reviews incorporate at least 3 of the top 5 words.

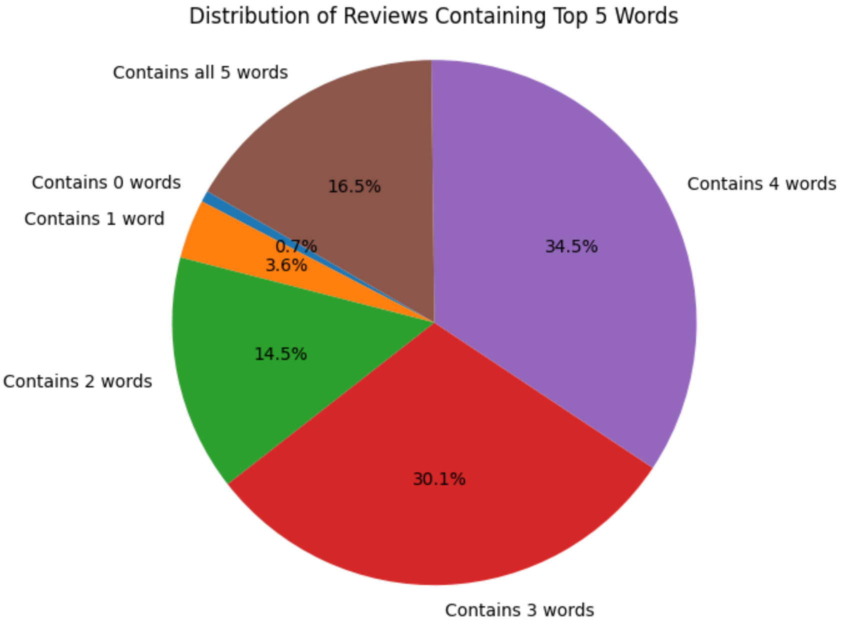


FIGURE 4: PIE CHART OF DISTRIBUTION OF REVIEWS CONTAINING TOP 5 WORDS

This chart illustrates how frequently the top 5 words—"hotel," "room," "stay," "great," and "good"—appear in hotel reviews, revealing their distribution across the dataset. The majority of reviews contain 3 or 4 of these words, highlighting common themes related to the stay, room quality, and overall positive sentiments. The fact that 64.6% of reviews mention at least 3 or 4 of these words suggests these terms are central to user experiences or how people discuss their experiences. Though unsurprising, having "hotel" as the number one word is interesting. Despite it often being clear from the context that the person is talking about a hotel, guests frequently mention the word "hotel" multiple times for clarity.

4. Cleaning of Text

The “clean_text” function was created to preprocess the dataset of hotel reviews. This function utilizes SpaCy’s NLP pipeline to process the reviews efficiently in batches, and several undertakes several steps, including: tokenization, lowercasing, removal of stop words and punctuation, lemmatization, and filtering of short tokens.

The order of these steps is crucial for maintaining the accuracy and quality of the processed text. Tokenization is performed first, breaking down each review into individual words, which allows subsequent operations to be applied at the word level. Lowercasing comes next, ensuring consistency in the text by treating tokens with different capitalizations (such as "Hotel" and "hotel") are counted as the same word to be for different metrics. After that, removing stop words and punctuation reduces noise by excluding commonly used words (e.g., "and," "the") and non-informative characters. This step is performed before lemmatization because stop words and punctuation do not need to be lemmatized, improving efficiency.

Lemmatization is then applied to convert each word to its base form (e.g., "staying" becomes "stay"), ensuring that different forms of the same word are treated the same and counted towards the same word for final metrics. Finally, filtering short words ensures that only words with more than two characters remain, removing non-informative tokens such as single-letter words that are unlikely to contribute to meaningful analysis. This was an unconventional approach to cleaning the text, but when creating the word cloud and bar chart, removing common connecting words like "I" and "A" significantly reduced the noise in these visualizations.

This systematic process enhances the quality of the cleaned text by ensuring consistency and reducing noise. By using the same pipeline across all visualizations and later text embedding methods, we can better understand the consistent dataset and the intricacies of the techniques applied. After the cleaning process, the cleaned reviews were stored in a new column, "cleaned_review," in the dataset. The *".head()"* function was then used to print the first few cleaned reviews to confirm the accuracy of the process. This approach ensures that the text is standardized and properly prepared for data visualization and further analysis.

5. Comparison of Word Embedding Techniques

In this analysis, three distinct word embedding techniques were applied to convert the textual hotel reviews into numerical features to train a classification model to predict review ratings. Each of these techniques were used to show different characteristics of the text, providing slightly different representations of the identical cleaned and processed dataset.

The first word embedding technique used was Bag of Words (BoW). A matrix of cleaned tokens was made in which each token represented a single feature, and the frequency of the word was recorded. In BoW, the contextual relationship of the words was not analyzed between the individual tokens. Next, Term Frequency-Inverse Document Frequency (TF-IDF) was used to show a relation like the BoW representation, but that importance was then adjusted over the entire dataset. Words that appear often throughout the corpus (such as our top 5 words found in the last section) were weighed with less importance, whereas words that were unique were given more importance. Finally, the last word embedding technique was Word2Vec, which represents the reviews as vectors to fully show the relationship between words. This differs from the first two techniques since instead of focusing on one token at a time, Word2Vec takes the entirety of the review.

Each of these embedding techniques was used to transform the data, which was then put through a simple logistic regression classifier to predict the star ratings (1-5) associated with each review. The performance of the classifier was evaluated using accuracy and classification metrics as shown in the tables below.

Bag of Words Classification Report				
Ranking	**Precision**	**Recall**	**F1-score**	**Support**
1	0.56	0.53	0.54	80
2	0.31	0.30	0.30	93
3	0.32	0.30	0.31	108
4	0.53	0.50	0.52	311
5	0.69	0.75	0.72	408
Accuracy			0.56	1000
Macro Avg.	0.48	0.47	0.48	1000
Weighted Avg.	0.56	0.56	0.56	1000

Table 1: Bag of Words Report

TF-IDF Classification Report				
Ranking	**Precision**	**Recall**	**F1-score**	**Support**
1	0.67	0.50	0.57	80
2	0.35	0.25	0.29	93
3	0.39	0.11	0.17	108
4	0.51	0.50	0.51	311
5	0.63	0.83	0.72	408
Accuracy			0.57	1000
Macro Avg.	0.51	0.44	0.45	1000
Weighted Avg.	0.54	0.57	0.54	1000

Table 2: TF-IDF Report

Word2Vec Classification Report				
Ranking	**Precision**	**Recall**	**F1-score**	**Support**
1	0.53	0.50	0.51	80
2	0.29	0.26	0.27	93
3	0.44	0.11	0.18	108
4	0.49	0.44	0.46	311
5	0.59	0.78	0.67	408
Accuracy			0.53	1000
Macro Avg.	0.47	0.42	0.42	1000
Weighted Avg.	0.51	0.53	0.50	1000

Table 3: Word2Vec Report

The results show that TF-IDF achieved the highest accuracy at 57.1%, slightly higher than Bag of Words (56.3%) and Word2Vec (53%). This is to be expected, as the intention was to use an identical cleaned dataset, and the cleaned dataset I had included removing stop words and words shorter than 2 characters. The removal of stop words is not a crucial note for BoW and TF-IDF, but in Word2Vec, the technique relies on capturing semantic similarity throughout the entire review. By removing stop words the sentiment could have been greatly altered. If further work was to be done on this dataset exclusively using Word2Vec, the SpaCy pipeline would need to be changed to accommodate this difference of processing.

Comparing the rankings over the classes, across all models, class 5 (positive reviews) was predicted most accurately, with the highest precision and F1-scores. I predict this is due to the larger number of samples in this class. Further, the models struggled with mid-range ratings (classes 2 and 3), showing significantly lower scores across all categories which suggests difficulty in distinguishing neutral reviews. It is predicted that due to not only there being fewer reviews with these neutral tones, some reviews contain both positive and negative indication words which the model has a difficult time deciphering, for example if “the food was great but the room was bad” is a 2 or a 3.

Overall, the model does not classify the data particularly well, and a more complex model would undoubtedly improve performance. However, for the purposes of this exercise, exploring patterns and trends in various word embedding techniques, even achieving 50-60% accuracy still revealed valuable insights.

6. Conclusion

This project applied NLP techniques to the "Tripadvisor Hotel Reviews" dataset using the SpaCy text cleaning pipeline to gather information on the set of reviews based on content found in the reviews. This pipeline was used to crate informative visualizations of the dataset including a word cloud, bar chart, and a pie chart. Further, Three word embedding methods—Bag of Words (BoW), TF-IDF, and Word2Vec to transform textual data into numerical features that were used for rudimentary classification. The goal was to evaluate the effectiveness of these methods in predicting star ratings (1-5) using a basic logistic regression model.

The results showed that that TF-IDF produced the highest accuracy at 57.1%, sligtly outperforming Bag of Words at 56.3% and Word2Vec at 53%. TF-IDF excelled due to its ability to weigh words based on their importance across the dataset, effectively capturing the distinctive features of the reviews. In contrast, Word2Vec underperformed, likely due to the text cleaning process that removed stop words, which is a key part of capturing the context needed by Word2Vec to understand semantic relationships. The removal of these words through the SpaCy pipeline hurt Word2Vec's ability to process each review and had a large impact on the techniques accuracy, especially considering that this technique is considered more advanced.

Another insight gathered was that the simple logistic regression model was best able to classify strongly positive reviews (class 5) more accurately than other ratings. This is likleu due to the larger number of reviews for class 5, allowing the models to better recognize patterns associated with positive feedback. Class 1 (low reviews) also performed similarly to class 4, and class 4 has significantly more reviews than class 1, showing that both class 1 and 5, were the easiest to classify due to their non-neutral nature. Along the same lines, the models struggled with mid-range ratings (classes 2 and 3), showing significantly lower performance. These ratings typically contain mixed feelings, where reviews express both positive and negative elements, making them harder to classify. This ultimately led to lower scores for these neutral reviews.

While the overall model accuracies (50-60%) may seem very modest, this simple model was able to effectively highlight differences between embedding techniques. TF-IDF proved to be the most suitable method for this dataset, given the text cleaning process. If further work were to prioritize Word2Vec, retaining stop words in the preprocessing step could potentially improve its performance by maintaining critical context within the reviews.

7. Bibliography

Explosion AI. (n.d.). *spaCy API Documentation*. Retrieved from https://spacy.io/api/doc/

Kaggle. (2024). Hotel reviews dataset. Retrieved from https://www.kaggle.com/datasets/joebeachcapital/hotel-reviews/discussion?sort=hotness

Řehůřek, R. (n.d.). *Gensim: Word2Vec model*. Retrieved from https://radimrehurek.com/gensim/models/word2vec.html