Mining hotel review data to discover the typical trends positive and negative reviews have in common

Jason A. Combs
Department of Computer Sciences
Simon Fraser University
British Columbia, Canada
jcombs@sfu.ca
301352433

CMPT459 – Introduction to Data Mining Final Project
Jian Pei

Abstract

Hotel reviews are not only great ways for the general public to learn more about a hotel and about the experiences people have had at this hotel but can also provide critical information for hotel owners and hotel managers about their hotel. This information once understood can be taken into careful consideration to improve the quality of a hotel. However, there are usually hundreds or thousands of different reviews online and being able to process this mass of information manually may prove to be a very time-consuming and tedious task. In this paper, I analyse two datasets of hotel reviews from hotels in the United States and Italy and data mine on each dataset to find the typical words that appear together in positive and negative reviews. In other words, we find the typical words characteristic of negative and positive reviews to summarize the things people mention when writing a negative or positive review.

1 Introduction

As a hotel owner or manager, it is important to listen to and understand feedback from your customers to maintain and improve customer service and guest experiences. Being able to understand the negative aspects of a hotel can lead to changes in an effort accommodate customers better while also being able to understand the positive aspects of a hotel can help ensure that any of the changes made either do not affect the things that customers like about your hotel or at least minimally affect these things. This knowledge about a hotel can lead to an increase in positive reviews. Positive reviews tend to quickly spread which in turn would result in more customers being more encouraged to choose your hotel over the hundreds of other hotels available. Thus, it is critical to the success of a hotel company to understand the reviews. This could be done by manually looking into each review. However, this method might take far too long depending on how long each review is as well as number of reviews to read. Because of this, a hotel owner may only read some reviews.

However, this introduces a new problem. Understanding each review is different than understanding all the reviews in the data as a whole. For example, a hotel owner observes that 5 people report in their reviews that the rooms on the top floor of the hotel are far too cold at night. The hotel owner then decides to adjust the temperatures to be a bit warmer to make these guests as well as future guests more satisfied with their stay. On the other hand, there are 76 people who reported in their reviews that the rooms on the top floor of the hotel are too warm at night. Since the hotel owner did not have a full (or better) understanding of the entire dataset, the hotel owner did the opposite of what the majority of the reviewers wanted to change, inconveniencing these people. This is an example on how a hotel owner or manager may interpret a small subset of reviews and conclude the wrong solution to accommodating customers better. With this problem in mind, I propose that my program is a useful tool to solve this problem.

My program uses the words used in the reviews, parses and tokenizes them, applies frequent pattern mining to find the most frequent patterns of words appearing in the data, and then creates visual representations on how often each pattern appeared via WordCloud and bar charts. This is done twice for each dataset, once for the negative reviews and a second time for the positive reviews. The results can provide hotel owners with a clearer look at what the majority of reviewers mention when it comes to giving a positive or negative hotel review to support more informed decision making.

2 Datasets

To conduct this program, I took my data composed of real reviews from the *kaggle* website. My first dataset is a combination of two datasets of hotel reviews in the US from Datafiniti's Business Database. The website splits the data into two subsets by date, so I combine these into one set in my program. The first subset contains 10,000 hotel reviews containing about 2,000 different hotels that were updated between January 2018 and September 2018. The second subset of data contains another 10,000 reviews containing about 1,400 different hotels that were updated between December 2018 and May 2019 on different hotels. Note that these two datasets are subsets of a much larger dataset that I will not be using which is available on the *Datafiniti*[1] website. So, my combined dataset has a list of 20,000 hotel reviews within the time ranges of January 2018 – September 2018 and December 2018 – May 2019. In this dataset, I did not use all the attributes of information available on each review. The attributes I did use were:

- "review.date", "reviews.rating", and "reviews.text";
- "city", "latitude", and "longitude".

I renamed each attribute which is shown in *Table 1*.

Review	Hotel
date	city
rating	country
text	latitude
	longitude
	name
	province

Table 1: Table of attributes used in the United States dataset

I noticed that the review ratings were out of 5 instead of 10 like my second data set is, so I doubled each rating.

My second dataset is a list of 515,000 hotel reviews containing about 1,493 hotels across Europe. This data was updated between August 2015 and August 2017. As 515K reviews is too large of a dataset for the purposes of my program, I filtered the data to show only hotel reviews from Italy. This narrowed down the number of reviews of my dataset to about 37,000. Once again, this dataset has more attributes of information on each review than I needed. The attributes I did use were:

- "Review_Date", "Negative_Review", and "Positive_Review";
- "lat" for latitude, and "lng" for longitude.

I renamed each attribute which is shown in *Table 2*.

Review	Hotel
date	address
negative_text	name
positive_text	latitude
rating	longitude

Table 2: Table of attributes used in the Italy dataset

For each dataset, I classified the "positive" reviews and "negative" reviews differently. For the first dataset, I classified the "positive" reviews as any review with a rating greater than 5 whereas any review that was less than or equal to 5 as a "negative" review. As for the second dataset, two attributes were already provided split-up into "Negative_Review" and "Positive_Review" attributes. I simply used this as a way to classify the data for my second dataset into positive and negative reviews.

3 Data Cleaning

Lots of data cleaning on both datasets was required to obtain the appropriate data for my program. The first step was removing an extra null column called "reviews.dateAdded" found in the Europe dataset. I then combined both subsets of the US dataset and ensured that the resulting dataset as well as the Europe dataset had no duplicate reviews. Next, I dropped many unnecessary attributes from each dataset and renamed each attribute to keep each dataset somewhat consistent with other. I then filtered the Europe dataset to contain only hotel reviews from Italy. Next, for both datasets I dropped any rows containing an empty or 'null' cell. I then doubled all the ratings in the US dataset and then filtered out any ratings less than 0 or greater than 10 for both datasets. Then, for both datasets I filtered out any reviews not in their respective countries using latitude and longitude values. Then, for the US dataset, I added a new attribute called "bad0_good1" to

[1] https://datafiniti.co/

keep track of which review is a negative or positive review by giving the review a 0 or 1, respectively. There are only about 2000 negative reviews in the US dataset, so I extracted 2000 positive and negative reviews, and 5000 reviews from the Europe dataset. I then reused the tokenization and parsing code used in Assignment 1 of CMPT459 to clean the review text in each dataset. In the Europe dataset, there are some cells in the positive and negative text review attributes that contain strings "No Negative" or "No Positive"; I removed any rows containing both strings in "negative_text" and "positive_text" attributes respectively. After tokenization, these strings became "negative" and "positive". I made sure to remove these strings from the data as they may affect my program's results. Lastly, I made sure I used the exact same amount of positive and negative reviews in the US dataset to ensure consistent results.

4 Frequent Pattern Mining

The frequent pattern data mining technique is a form of association rule learning and is the foundation for many data mining tasks[2]. It is useful for mining associations, correlations, interesting patterns, and most frequent patterns in large datasets. The FP-growth algorithm and the Apriori algorithm are examples of algorithms that are used to complete these tasks but, in this project, I use the FP-growth algorithm.

How does the FP-growth algorithm work? Well firstly, a support threshold is set. This value determines the minimum amount of times an item or item pattern in a dataset must occur to be chosen. After this, the algorithm computes the frequency of all elements in the dataset and only chooses those with a frequency greater than or equal to the minimum support value. Next, an FP-tree is constructed. The algorithm iterates through the itemsets and add a node to the tree. If a subset of an itemset is already in the tree, then the frequency count is increased. If the subset of the itemset is not in the tree, create a new node and set its frequency to 1 since this would be the first occurrence of this pattern in the tree. Next, a list of all the paths from the root of the tree that lead to the chosen itemsets is created. [10] Next, for each of the selected items, a set of elements common across the paths created listed in the previous step is created. Lastly, the frequent patterns are generated by grouping items in the tree with their respective items.

I chose to use frequent pattern mining because the information provides a good outlook on the many different objects used in the pattern mining. In this case, it makes relationships between different words occurring in negative and positive reviews more obvious and allows for a quick summarization of a large sets of data, instead of alternatively learning these relationships manually.

In my program, I use the find_frequent_patterns(…) function from built-in package *pyfpgrowth*, to compute the most frequent patterns. I use this function two times on each dataset, the first time using the negative review words and the second using the positive review words. By then end of this, I have 4 dictionaries showing the patterns along with the associated frequencies sorted in descending order. These dictionaries are converted into data frames and then saved to the current directory of the user as a CSV file.

5 Results

I managed to create four data frames containing the most frequent patterns of positive and negative reviews for both US and Italy datasets. From these data sets I created a WordCloud for each and placed the WordClouds for the negative and positive reviews for a country side-by-side. This can be seen for the US dataset in *Figure 1* and the Italy dataset in *Figure 2*. The WordClouds are made by making the most frequent pattern show up in a larger font than the rest of the patterns. This makes words like "room" and "hotel" pop up in a large font. But if you look closer at the positive reviews, you can see patterns like "friendly, staff", "great, location", "clean, hotel", and "good, hotel, room". These could be indications that any hotel with these qualities could make a hotel more successful. Looking at the negative reviews, we see patterns like "experience, hotel", "front, desk", "bed, room", and "hotel, service". This is summarized information on what people typically talk about when they write a negative hotel review.

[2] Slide 10 of Lecture 6 – *Finding useful patterns and rules*

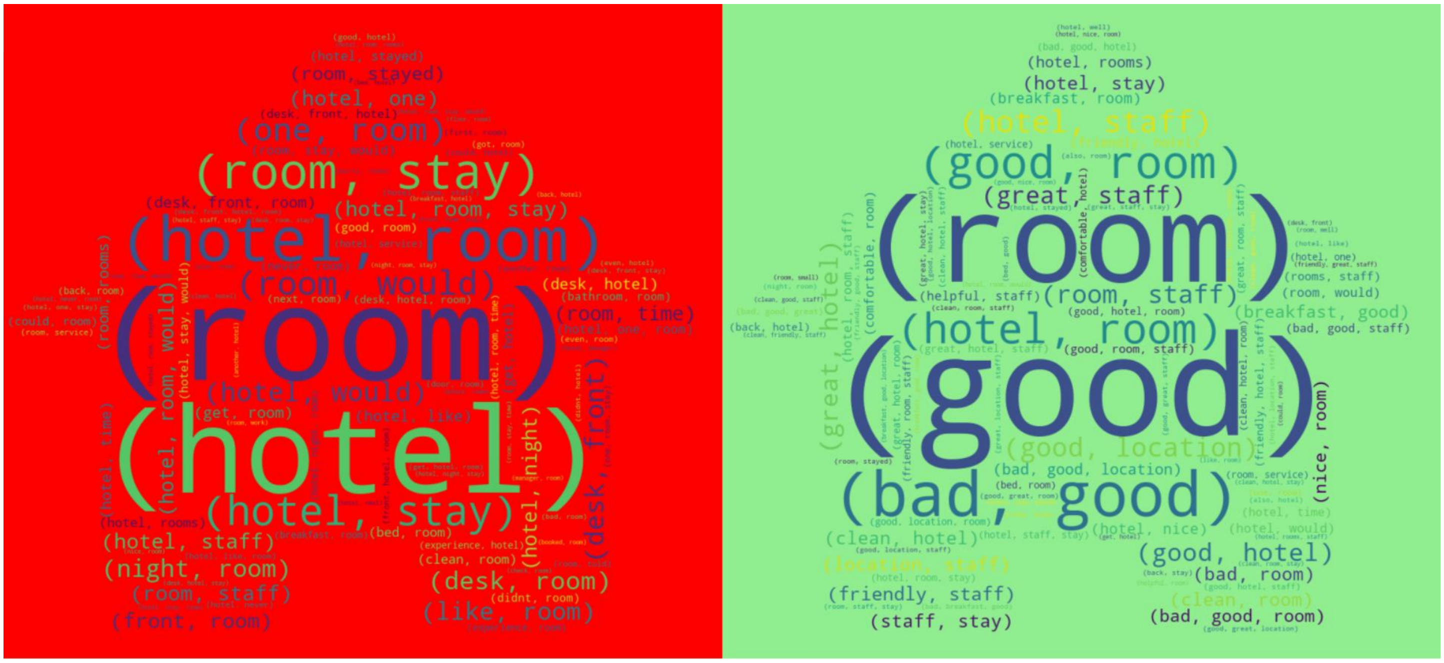

Figure 1: WordClouds of most frequent patterns of negative reviews (left in red) vs positive reviews (right in green) in the USA dataset.

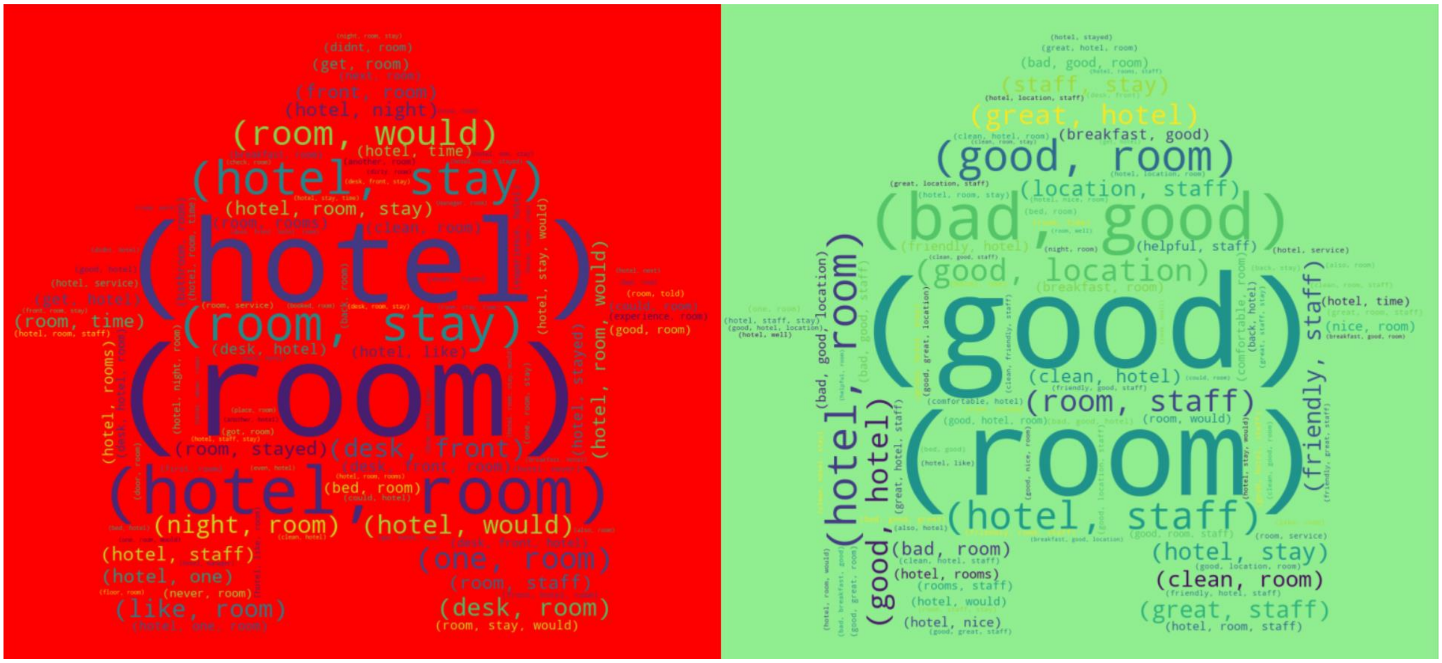

Figure 2: WordClouds of most frequent patterns of negative reviews (left in red) vs positive reviews (right in green) in the Italy dataset.

However, I do not find WordCloud to be the best way to represent my data since WordCloud emphasizes the most frequent patterns more than the other frequent patterns. Because of this, I also created bar charts to help visualize the data. For my program, I took only the top 10 most frequent word patterns to better represent which word patterns occurred often. This can be changed in my program to include more than just the top 10. I made the bars *red* to represent *negative reviews* and *green* to represent *positive reviews*. These can be seen in *Figure 3*, *Figure 4*, *Figure 5*, and *Figure 6*.

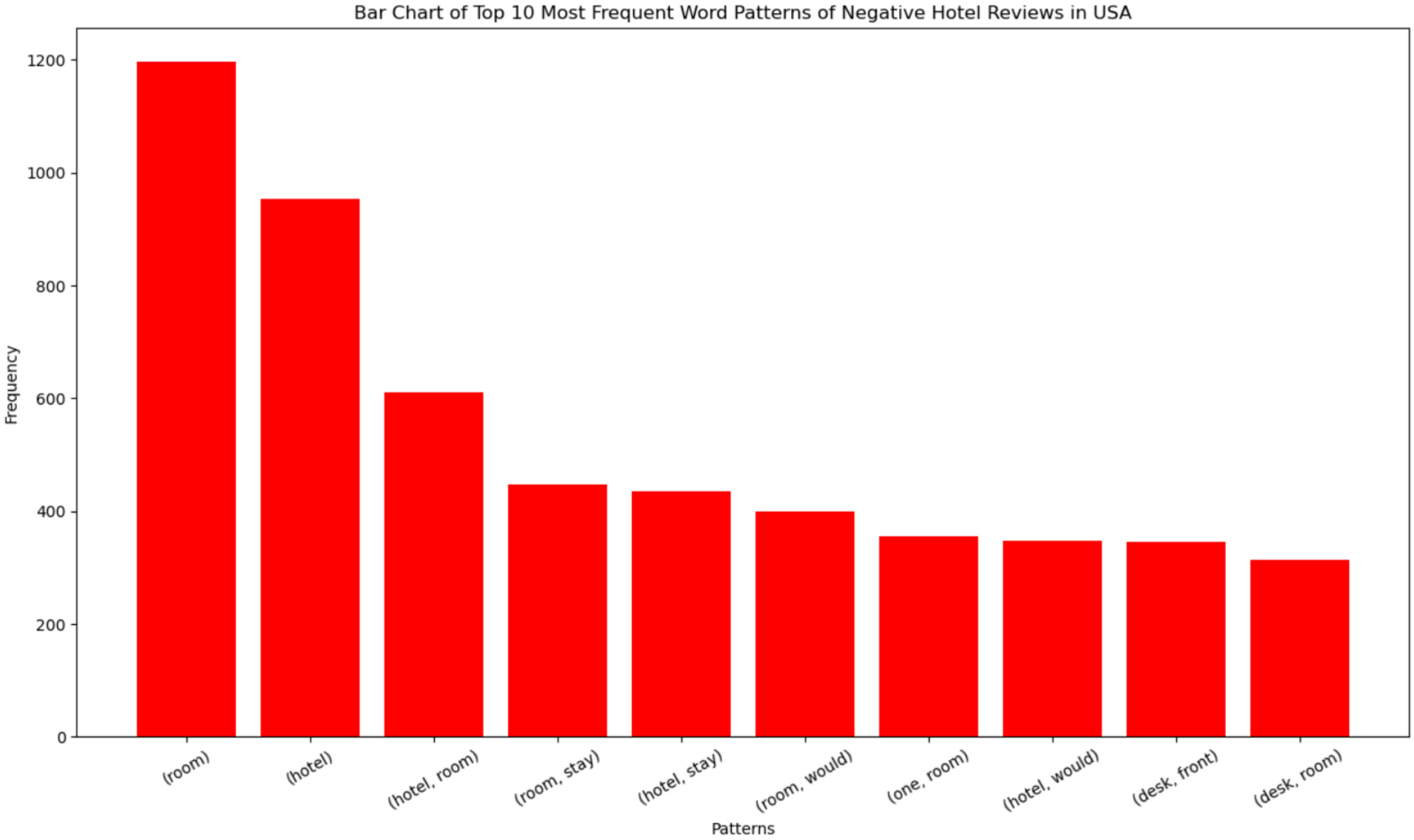

Figure 3: Bar chart of the top 10 most frequent word patterns of the negative hotel reviews in the US.

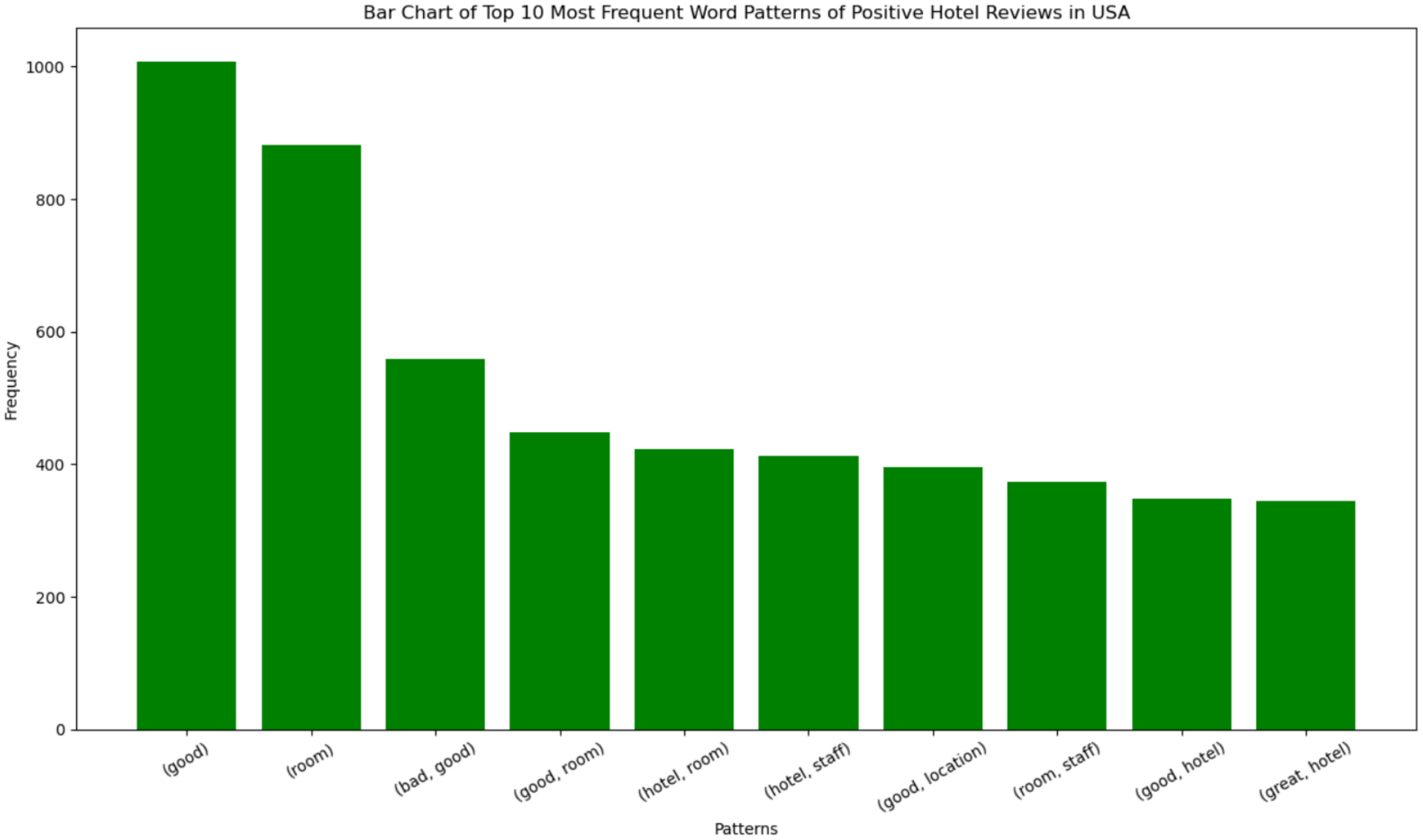

Figure 4: Bar chart of the top 10 most frequent word patterns of the positive hotel reviews in the US.

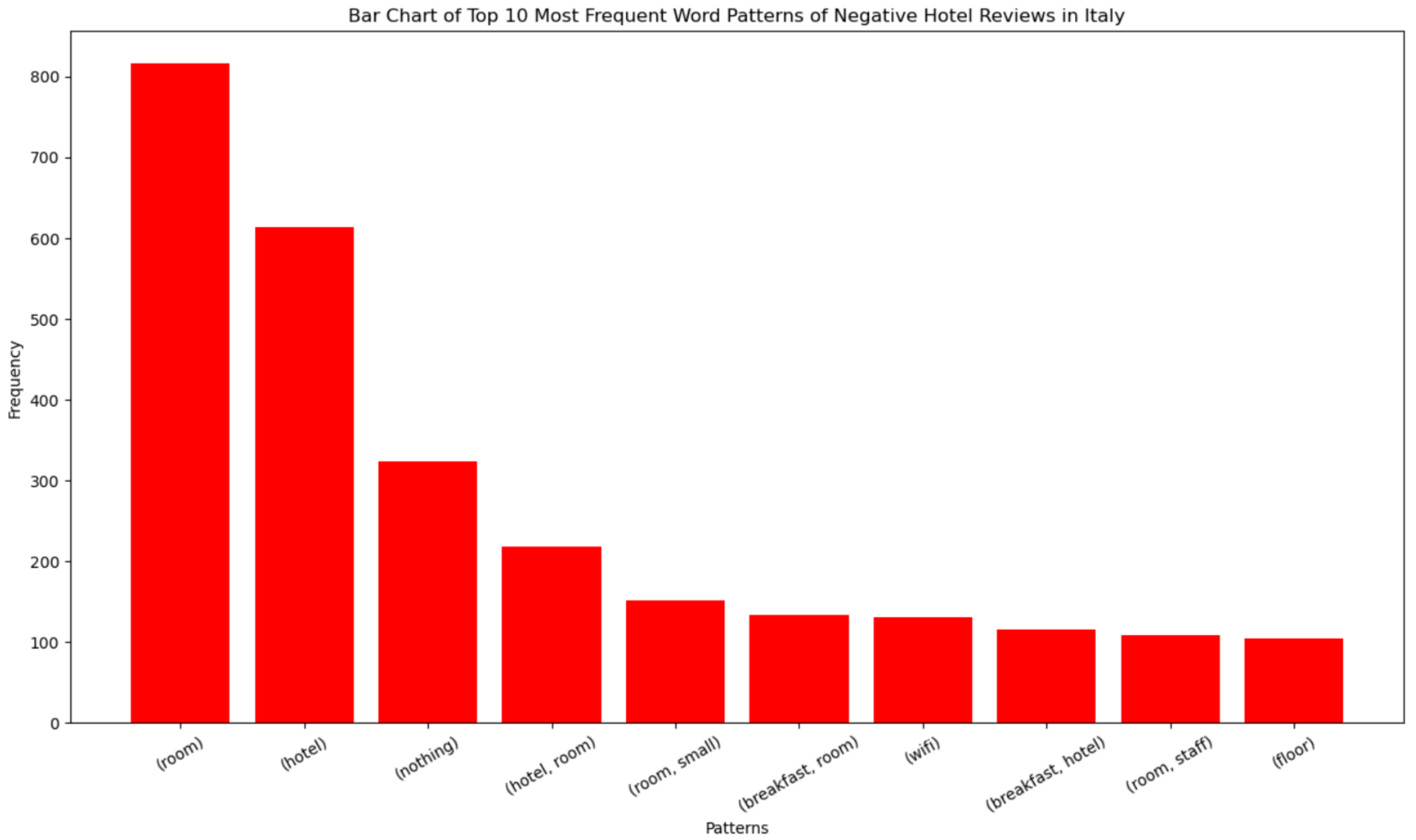

Figure 5: Bar chart of the top 10 most frequent word patterns of the negative hotel reviews in Italy.

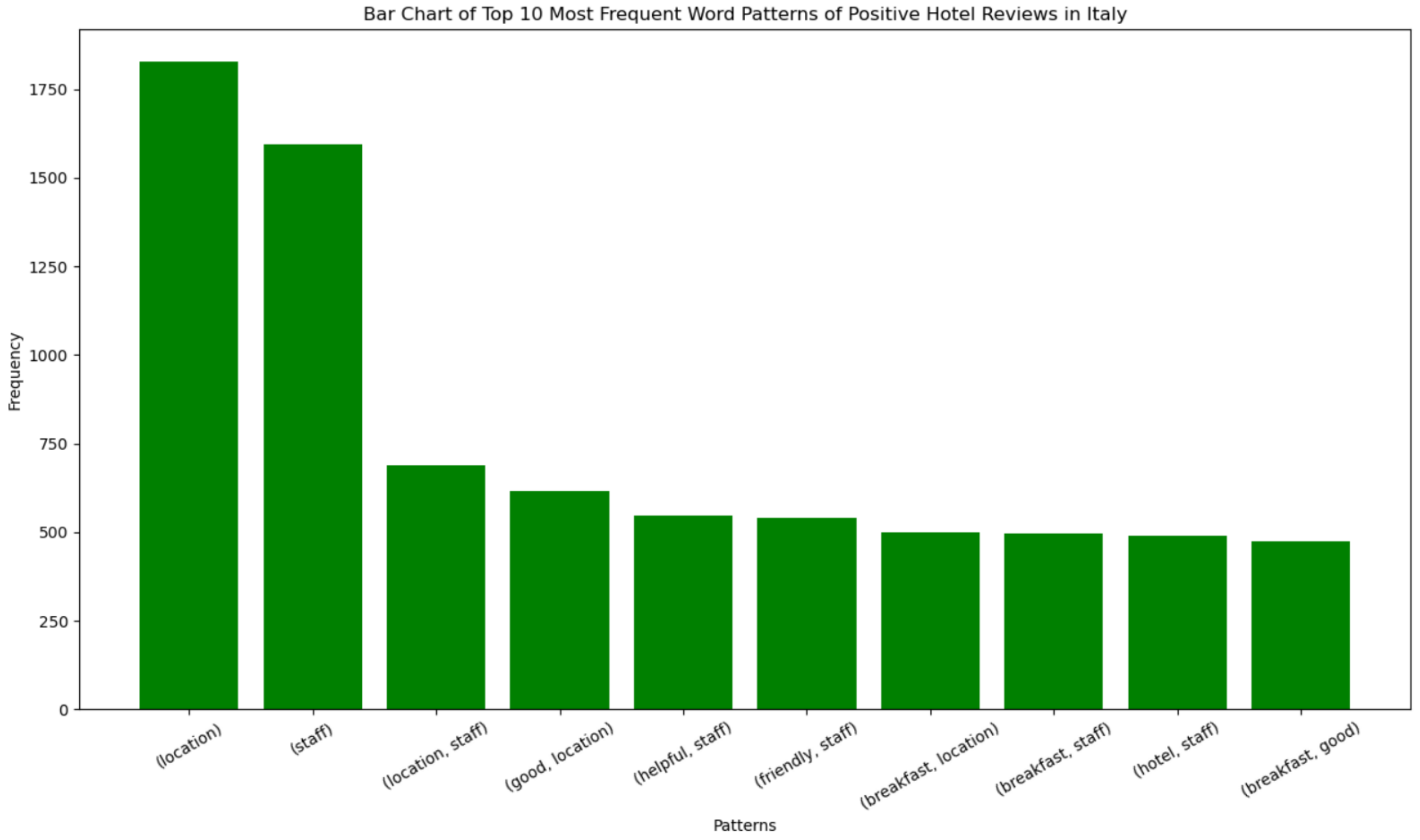

Figure 6: Bar chart of the top 10 most frequent word patterns of the positive hotel reviews in Italy.

In *Figure 3*, *Figure 4* and *Figure 5*, it is easy to see that "hotel" and "room" comes up very often in positive and negative reviews. This makes sense since these words are likely to occur in any review. So, a further extension to improve my program would likely be to either ignore words that can appear in both negative and positive reviews or find a way to interpret these words into the program while making it obvious what is being mentioned. This can likely be done by including other words used in the same sentences as "hotel" or "room" to provide more knowledge on what these words mean.

Besides that, we can see from *Figure 4* that "room, good" appears often for US hotels which could be a sign that the hotel managers or hotel owner has properly accommodated guests in this way. According to this bar chart, the hotel staff appear to have a positive impression on the hotel reviewer's as well that the hotels are in a good location. In *Figure 5*, we can see that breakfast offered at Italian hotels may not be ideal as well as the Wi-Fi should likely be improved. In *Figure 6*, we can see the location is a big factor to reviewers enjoying their stay at Italian hotels. Besides that, the staff apparently do an amazing job since the bar chart mentions that they are helpful and friendly.

However, another problem occurring in *Figure 5* and *Figure 6* is that the first bar chart mentions "breakfast, room" as a negative part of their hotel, while the second bar chart mentions "breakfast, location", "breakfast, staff", and "breakfast, good". These contradict each other. So, a further extension of this program could take into account of contradicting information like this and remove the data for breakfast in *Figure 5* since the *Figure 6* has "breakfast" occurring much more in the positive dataset rather than the negative dataset.

6 Conclusions

We focused on hotel reviews to investigate the general trends in the text of positive and negative reviewer text. The results we obtained are good because now we have an enhanced idea on what do people generally mention when they had negative or positive experiences at a hotel. I feel like this program could be used on much larger datasets to gather more information on what hotels do right or wrong.

My program is using data extracted from many different hotels and therefore, the general trends we are mining for are for multiple hotels. But not all hotels have the same problems as other hotels. These results show only the qualities of the hotels recorded in the US and Italy in general. So, a hotel manager or a hotel owner could use only data from their own hotel, so that the results they obtain from my program would be more personalized to the negative and positive parts of their hotel.

Mentioning one of the problems discussed before, changes should be made to take into consideration the words that generally appear in both positive and negative reviews as this has made an obvious effect in the results shown in the WordClouds and bar charts. In addition to that, the second problem I discussed was how the information as shown in the bar charts contradicted itself. This should also be taken into consideration for future applications of this program. Another possible problem is how I split my data up into "positive" and "negative" reviews. First off, is defining a bad review as any review with a rating 5 or less a good idea? This may need more consideration. Additionally, I used two different ways to split my data up, one was based on the ratings and the other was based on how the data was already ordered into negative and positive reviews. This difference I originally thought would give more of a variety in results, but it is possible that this difference creates too much of a deviation between the results of the datasets. Once again, this is another thing to give more consideration to.

Nevertheless, this approach can help both hotel owners and managers as well as the general public. The information can be shared publicly to provide people with a better awareness on what other people think about a hotel. This may be nice for the person reading a review since they are likely to read only a subset of reviews about a hotel instead of all the reviews about a hotel.

This method of data summarization may also be used in different fields too including car rental or car mechanical services, restaurants, nature trails or hikes, to cite a few. Small adjustments to the program would have to made but as we are using association rule mining, this technique can be easily used in other ways not used or mentioned here.

7 References

[1] "Hotel Reviews", kaggle, 2019. US dataset 1 accessed from: https://www.kaggle.com/datafiniti/hotel-reviews?select=Datafiniti_Hotel_Reviews.csv US dataset 2 accessed from: https://www.kaggle.com/datafiniti/hotel-reviews?select=Datafiniti_Hotel_Reviews_Jun19.csv [Accessed: 3-August-2020].

[2] “515K Hotel Reviews Data in Europe”, kaggle, 2017. Europe dataset accessed from: https://www.kaggle.com/jiashenliu/515k-hotel-reviews-data-in-europe [Accessed: 3-August-2020].

[3] “Mining implicit data association from *Tripadvisor* hotel reviews”. Vittoria Cozza, Marinella Petrocchi, Angelo Spognardi. Accessed from: http://ceur-ws.org/Vol-2083/paper-09.pdf [Accessed: 4-August-2020].

[4] “Predicting Hotel Rating from Reviews”, kaggle, 2017. Luke Puglisi. Accessed from: https://www.kaggle.com/lpuglisi/predicting-hotel-rating-from-reviews [Accessed: 4-August-2020].

[5] “Fine-grained Sentiment Analysis in Python (Part 1)”, towards data science, 2019. Prashanth Rao. Accessed from: https://towardsdatascience.com/fine-grained-sentiment-analysis-in-python-part-1-2697bb111ed4 [Accessed: 3-August-2020].

[6] “Pandas : Get unique values in columns of a Dataframe in Python”, thispointer.com, 2019. Accessed from: https://thispointer.com/pandas-get-unique-values-in-single-or-multiple-columns-of-a-dataframe-in-python/ [Accessed: 3-August-2020].

[7] “Understand and Build FP-Growth Algorithm in Python”, towards data science, 2020. Andrewngai. Accessed from: https://towardsdatascience.com/understand-and-build-fp-growth-algorithm-in-python-d8b989bab342 [Accessed: 4-August-2020].

[8] “mini project2-sentiment analysis”, kaggle, 2020. Hannah Wu. Accessed from: https://www.kaggle.com/hannahwendanwu/mini-project2-sentiment-analysis [Accessed: 5-August-2020].

[9] “FP-Growth”, pyfpgrowth 1.0, 2016. Accessed from: https://pypi.org/project/pyfpgrowth/ [Accessed: 6-August-2020].

[10] “FP-GROWTH: An Ultimate Guide To Pattern Mining”, Machine Learning Py, 2020. Priya Ghetia. Accessed from: https://machinelearningpy.com/machine-learning/fp-growth-ultimate-guide/#Advantages_of_FP-Growth [Accessed: 7-August-2020].

[11] “Implementing FP-Growth in python”, medium.com, 2018. Pushkhalla Chandramoulli. Accessed from: https://medium.com/@pcm1312/implementing-fp-growth-in-python-170f3dc64d78 [Accessed: 7-August-2020].

[12] “Generate word cloud from single-column Pandas dataframe”, stack overflow, 2017. Accessed from: https://stackoverflow.com/questions/43606339/generate-word-cloud-from-single-column-pandas-dataframe [Accessed: 7-August-2020].

[13] “Populating a dictionary using for loops (python) [duplicate]”, stack overflow, 2015. Accessed from: https://stackoverflow.com/questions/30280856/populating-a-dictionary-using-for-loops-python [Accessed: 7-August-2020].

[14] “Pandas DataFrame.sum()”, java t point, 2018. Accessed from: https://www.javatpoint.com/pandas-sum#:~:text=sum()-,Pandas%20DataFrame.,the%20values%20in%20each%20column. [Accessed: 7-August-2020].

[15] “Home logo”, kissclipart. Accessed from: https://www.kissclipart.com/house-animation-png-clipart-house-home-building-v0kcma/ [Accessed: 7-August-2020].

[16] “Generating Word Cloud Python”, GeeksforGeeks, 2020. Accessed from: https://www.geeksforgeeks.org/generating-word-cloud-python/ [Accessed: 8-August-2020].

[17] “CMPT 459 (Data Mining) PatternMining (Part 3 FP-growth)” , YouTube, 2020. Jian Pei. Accessed from: https://www.youtube.com/watch?v=ygPtvRTLEIQ&feature=youtu.be [Accessed: 5-August-2020].

[18] “Twitter ETL Tutorial”, 2020. arjuun09. Accessed from: https://www.youtube.com/watch?v=Cz2hSCEDJqs&feature=youtu.be [Accessed: 5-August-2020].

[19] “Twitter Streaming Tutorial”, 2020. arjuun09. Accessed from: https://www.youtube.com/watch?v=WSTob0j3I3E&feature=youtu.be [Accessed: 5-August-2020].

[20] Slide 10 of “Lecture 6 – Finding useful patterns and rules”, CMPT459 – Introduction to Data Mining, 2020. Jian Pei. [Accessed: 26-July-2020].