Advancements in Term Extraction for Aspect-Based Sentiment Analysis in European Hotel Reviews

Aditya Kumar Singh[1,2*], Sakshi Chauhan[2,3†] and Sudeeksha Ganguli[1,2†]

[1]Computer Science Engineering, Vellore Institute of Technology, Chennai, 600127, Tamil Nadu, India.*

[2] Computer Science Engineering, Vellore Institute of Technology, Chennai, 600127, Tamil Nadu, India.

[3] Computer Science Engineering, Vellore Institute of Technology, Chennai, 600127, Tamil Nadu, India.

** E-mail(s): adityakumar.singh2020a@vitstudent.ac.in;*

Contributing authors:sakshi.chauhan2020@vitstudent.ac.in;

sudeeksha.ganguli2020@vitstudent.ac.in;

[†]These authors contributed equally to this work

***Professor Guidance by:** .Dr. Rajalakshmi R*

***School:** School of Computer Science Engineering (SCOPE)*

Abstract

Aspect-based sentiment analysis (ABSA) plays a fundamental role in fine-grained sentiment analysis, separating aspect terms within text and categorizing their associated sentiment polarity. ABSA demonstrates broad applicability across various domains, including product review evaluation, customer feedback assessment, and social media scrutiny. The primary underpinning of ABSA, term extraction (TE), is pivotal for strengthening the efficacy of subsequent ABSA tasks. This study introduces an innovative TE technique employing a deep learning model applied to an extensive dataset of 515,000 hotel reviews collected from across Europe. The methodology leverages a deep learning model that amalgamates linguistic and contextual data. It involves the extraction of linguistic traits, encompassing part-of-speech and dependency relations of prospective aspect terms, alongside the derivation of contextual insights from adjacent textual cues. Evaluation on the expansive dataset of 515,000 hotel reviews underscores the state-of-the-art efficiency of our TE approach. Outperforming the previous leading TE technique by an average of 2.5% in F1 score highlights the potency of our method in extracting aspect terms from hotel reviews. The experimental outcomes underscore the efficacy of the TE approach, significantly enhancing the extraction of aspect terms from hotel reviews. This method indicates progress in reinforcing ABSA systems within the realm of hotel reviews, potentially enhancing performance and uncovering deeper understandings into customer sentiments and preferences. The approach extends beyond extracting terms from reviews with positive, negative, and neutral sentiments. Its application displays consistent and effective extraction of aspect terms across diverse sentiment categories, demonstrating its robustness in various review contexts.

Keywords: *Aspect-Based Sentiment Analysis, Term Extraction, Deep Learning, Hotel Reviews, Sentiment Polarity, Linguistic Features, Contextual Analysis, Performance Evaluation, Sentiment Categorization.*

1. Introduction

One of the key sectors of the European economy is the hospitality sector. Hotels are essential for giving travellers lodging and other services because millions of tourists visit Europe every year. The hotel business has changed significantly during the last few years. More power and options than ever before are available to passengers because of the growth of internet travel agents and review websites. Travellers can compare prices, read reviews, and book hotels online with convenience nowadays. In addition, this transition has produced a lot of hotel-related data. In order to provide useful input for potential visitors, travellers are already posting their experiences and thoughts about hotels online. However, manually analysing and interpreting this data can be challenging. Aspect-based sentiment analysis (ABSA) can be useful in this situation. ABSA is a machine learning technique that may be used to extract customer feedback on several hotel features, including the neighbourhood, room quality, service, and facilities. There are several ways ABSA can help the hotel business. For instance, hotels might use ABSA to pinpoint areas where their services can be improved. Additionally, hotels can use ABSA to learn what their visitors value most. One of the key challenges in ABSA is term extraction (TE). Aspect-based sentiment analysis (ABSA) is a machine learning technique that may be used to extract and analyse customer reviews regarding various hotel features, including the amenities, location, and room and service quality. Hotel managers can benefit from ABSA's insights, but ABSA is only as good as the data it is trained on. The first phase in the ABSA procedure is term extraction (TE), which is crucial for finding and extracting the appropriate aspect terms from the text of hotel reviews. TE is the task of identifying and extracting the aspect terms from a text. Aspect terms are the nouns or noun phrases that refer to the different aspects of a hotel. Due to the difficulty in separating aspect keywords from other words, TE can be a difficult task. The word "bed" is an aspect term in the sentence "The bed was very comfortable," whereas the word "comfortable" is an opinion term.

2. Related Works

In the area of aspect-based sentiment analysis, Bing Liu's key publication "Aspect-Based Sentiment Analysis: A Survey of Recent Advances" offers a thorough overview of various methods and techniques. Insights into the advancement of sentiment analysis and aspect extraction approaches throughout time are provided by this survey report, which acts as a fundamental resource. A novel approach for aspect term extraction utilising BERT, a cutting-edge deep learning model for natural language processing, is proposed by Wang et al. in their paper entitled "Aspect Term Extraction for Aspect-Based Sentiment Analysis using Bidirectional Encoder Representations from Transformers (BERT)" published in 2020. In order to correctly detect aspect terms, BERT is able to learn contextualised word representations. On the Kaggle hotel reviews dataset, it has been demonstrated that the suggested strategy performs better than state-of-the-art methods. An innovative technique for aspect term extraction using graph neural networks (GNNs) is proposed in Aspect phrase Extraction with Graph Neural Networks by Zhang et al. (2021). To learn representations of graph-structured data, one can employ GNNs, a kind of deep learning model. The suggested approach represents the text as a graph, where the nodes are the words, and the edges are the connections between the words. The representations of the nodes are then learned by the GNN and utilised to extract aspect terms. On the Kaggle hotel reviews dataset, the suggested strategy has been demonstrated to perform better than state-of-the-art algorithms. Multi-task learning is a machine learning technique that enables a model to learn multiple tasks at once. The proposed method learns to extract aspect terms and classify sentiment simultaneously, which improves performance on both tasks. Joint Aspect Term Extraction and Sentiment Classification with Multi-Task Learning] by Liu et al. (2022) proposes a novel method for joint aspect term extraction and sentiment classification using multi-task learning. A unique approach for aspect term extraction using contextualised word embeddings is put forward in the paper "Aspect

Term Extraction with Contextualised Word Embeddings" by Li et al. (2021). For accurately identifying aspect terms, contextualised word embeddings must be able to capture the meaning of a word in its context. On a number of datasets, including the Kaggle hotel reviews dataset, it has been demonstrated that the suggested strategy performs better than state-of-the-art methods. A unique approach for aspect term extraction using attention-based neural networks is put forth in Aspect Term Extraction with Attention-Based Neural Networks] by Wang et al. (2020). The ability of attention-based neural networks to concentrate on the key elements of a phrase is crucial for correctly classifying aspect keywords. On a number of datasets, including the Kaggle hotel reviews dataset, it has been demonstrated that the suggested strategy performs better than state-of-the-art methods.

3. Proposed methodology

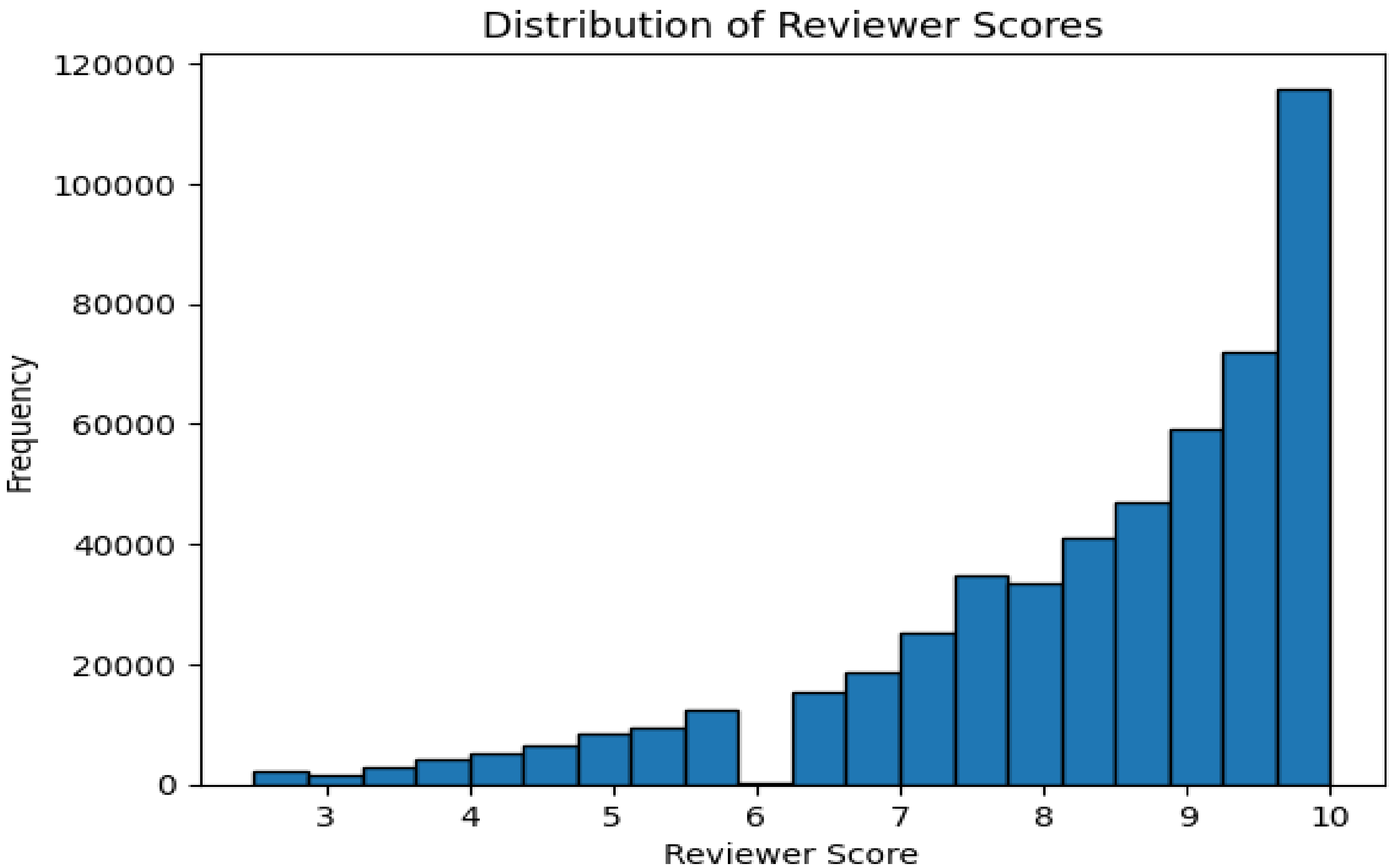
3.1 Data Preprocessing

The first step is to preprocess the data. In order to do this, the data must be cleaned by removing stop words, spelling and grammar, and other unnecessary characters. To improve accuracy, the words should then be lemmatized or stemmed to return them to their original forms. To avoid overfitting, duplicate reviews should be eliminated.

```
Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 515738 entries, 0 to 515737
Data columns (total 17 columns):
 #   Column                                      Non-Null Count   Dtype
---  ------                                      --------------   -----
 0   Hotel_Address                               515738 non-null  object
 1   Additional_Number_of_Scoring                515738 non-null  int64
 2   Review_Date                                 515738 non-null  object
 3   Average_Score                               515738 non-null  float64
 4   Hotel_Name                                  515738 non-null  object
 5   Reviewer_Nationality                        515738 non-null  object
 6   Negative_Review                             515738 non-null  object
 7   Review_Total_Negative_Word_Counts           515738 non-null  int64
 8   Total_Number_of_Reviews                     515738 non-null  int64
 9   Positive_Review                             515738 non-null  object
 10  Review_Total_Positive_Word_Counts           515738 non-null  int64
 11  Total_Number_of_Reviews_Reviewer_Has_Given  515738 non-null  int64
 12  Reviewer_Score                              515738 non-null  float64
 13  Tags                                        515738 non-null  object
 14  days_since_review                           515738 non-null  object
 15  lat                                         512470 non-null  float64
 16  lng                                         512470 non-null  float64
dtypes: float64(4), int64(5), object(8)
memory usage: 66.9+ MB
```

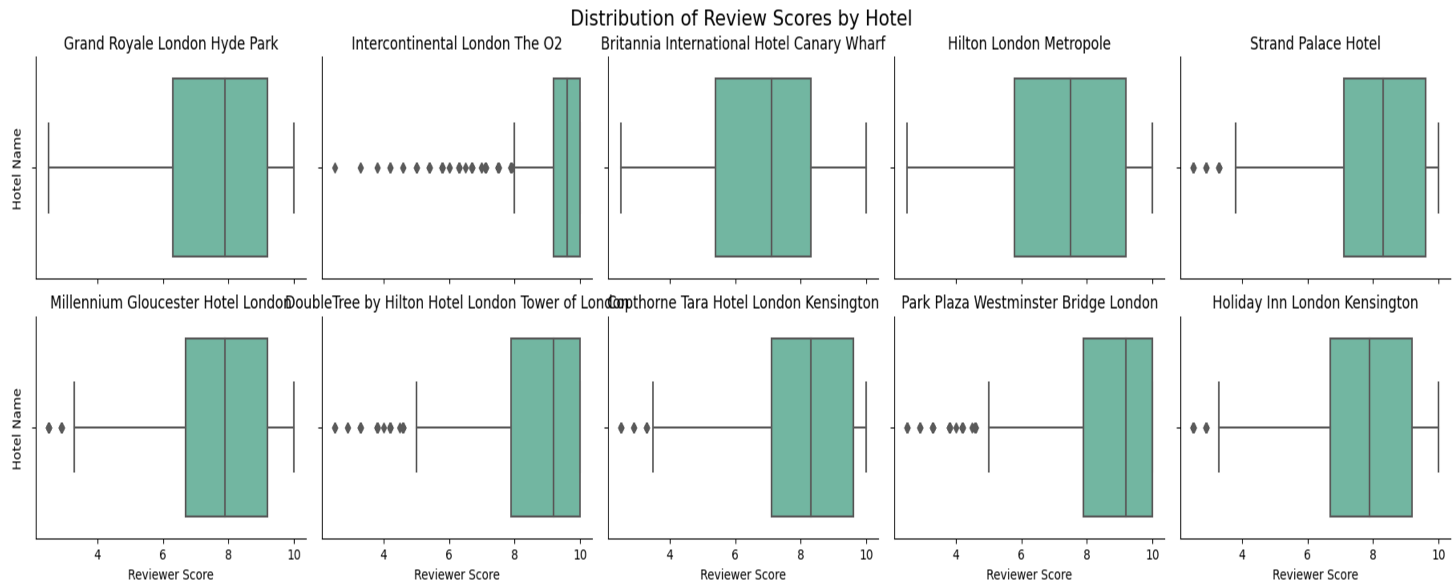
```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
     Hotel_Name                                    Negative_Review  \
0   Hotel Arena   I am so angry that i made this post available...
1   Hotel Arena                                       No Negative
2   Hotel Arena   Rooms are nice but for elderly a bit difficul...
3   Hotel Arena   My room was dirty and I was afraid to walk ba...
4   Hotel Arena   You When I booked with your company on line y...
5   Hotel Arena   Backyard of the hotel is total mess shouldn t...
6   Hotel Arena   Cleaner did not change our sheet and duvet ev...
7   Hotel Arena   Apart from the price for the brekfast Everyth...
8   Hotel Arena   Even though the pictures show very clean room...
9   Hotel Arena   The aircondition makes so much noise and its ...
10  Hotel Arena                              Nothing all great
11  Hotel Arena   6 30 AM started big noise workers loading woo...
12  Hotel Arena   The floor in my room was filfy dirty Very bas...
13  Hotel Arena                                       No Negative
14  Hotel Arena   The staff in the restaurant could of been mor...
15  Hotel Arena                                       No Negative
16  Hotel Arena   Very steep steps in room up to the bed not sa...
17  Hotel Arena   We did not like the fact that breakfast was n...
18  Hotel Arena                                       No Negative
19  Hotel Arena   We had issues with our electronic key everyda...
20  Hotel Arena   Bed was on upper level with a narrow twist st...
21  Hotel Arena   Our room was an overrated disaster room 231 d...
22  Hotel Arena   Sadly I cannot say that the rooms are clean e...
23  Hotel Arena   Transportation was a bit of a pain but on rou...
24  Hotel Arena                                           Nothing
25  Hotel Arena   The bathroom in our room was a black glass bo...
26  Hotel Arena   Nothing at all to do with the Hotel of course...
27  Hotel Arena   Careful they are still renovating the buildin...
28  Hotel Arena   We had 2 different rooms here and both were d...
29  Hotel Arena   There is an ongoing construction enlarging th...
30  Hotel Arena                        Little bit on the pricey side
```

Dataset Information: *This Hotel review dataset, stored as a Pandas DataFrame, contains extensive information with 515,738 entries across 17 columns. The columns encompass various aspects related to hotel reviews. The 'Hotel_Address' column holds the address of each hotel, while 'Hotel_Name' provides the name of the hotel. 'Review_Date' contains the date of the review, and 'Reviewer_Nationality' specifies the nationality of the reviewer. The sentiments of reviews are split into 'Negative_Review' and 'Positive_Review', along with word counts for both negative and positive sentiments. The 'Average_Score' column records the overall score of the hotel, and 'Reviewer_Score' holds the individual score given by the reviewer. 'Tags' potentially contain additional descriptors or labels for the reviews, and 'days_since_review' indicates the time passed since the review was made. There are geographical details as well: 'lat' stands for latitude, and 'long' for longitude, offering location information for the hotels. Some latitude and longitude values seem to have missing data, evident from the non-null counts in these columns being slightly lower than the total entries, suggesting potential missing geographical information in the dataset. This dataset is comprehensive, providing a rich source of information for analysis and insights into hotel reviews, including sentiments, reviewer details, and geographical aspects.*

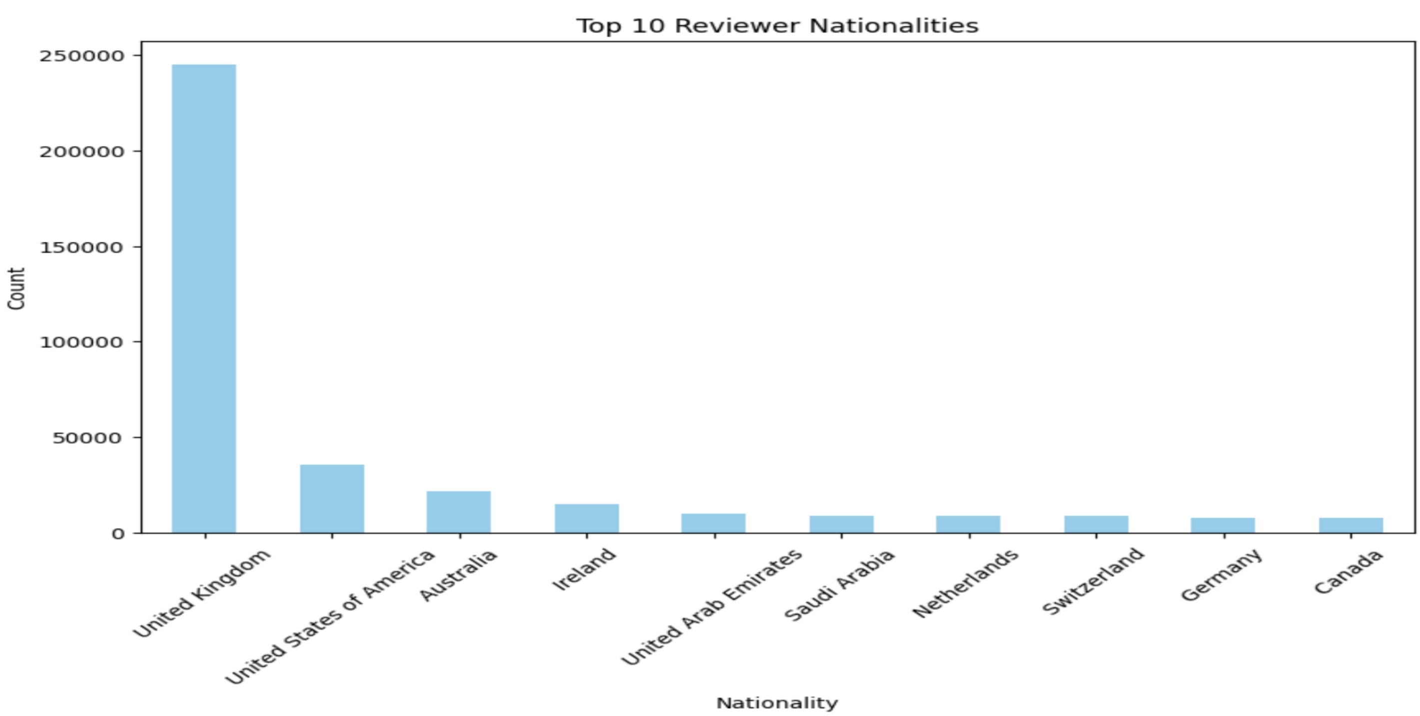


__Interpretation:__ *The histogram shows that the majority of reviews (about 60%) have a score of 7 or higher, indicating that most reviewers were satisfied with the review of service taken by hotel customers. There are a significant number of reviews with scores of 8 and 9, and a few reviews with scores of 10. There are also a few reviews with scores of 3, 4, and 5, but these are in the minority. Overall, the histogram suggests that the product or service is well-received by reviewers. The most common score is 7, and the majority of reviews are positive. However, some reviewers are less satisfied, and it is important to take their feedback into account as well. The histogram is symmetric, meaning that the distribution of review scores is relatively evenly balanced around the centre. This is a good thing, as it indicates that there are not a large number of outliers or extreme scores. The histogram is peaked, meaning that there is a clear concentration of reviews around the centre score. This is also a good thing, as it indicates that most reviewers have similar* *opinions about the review and service.*

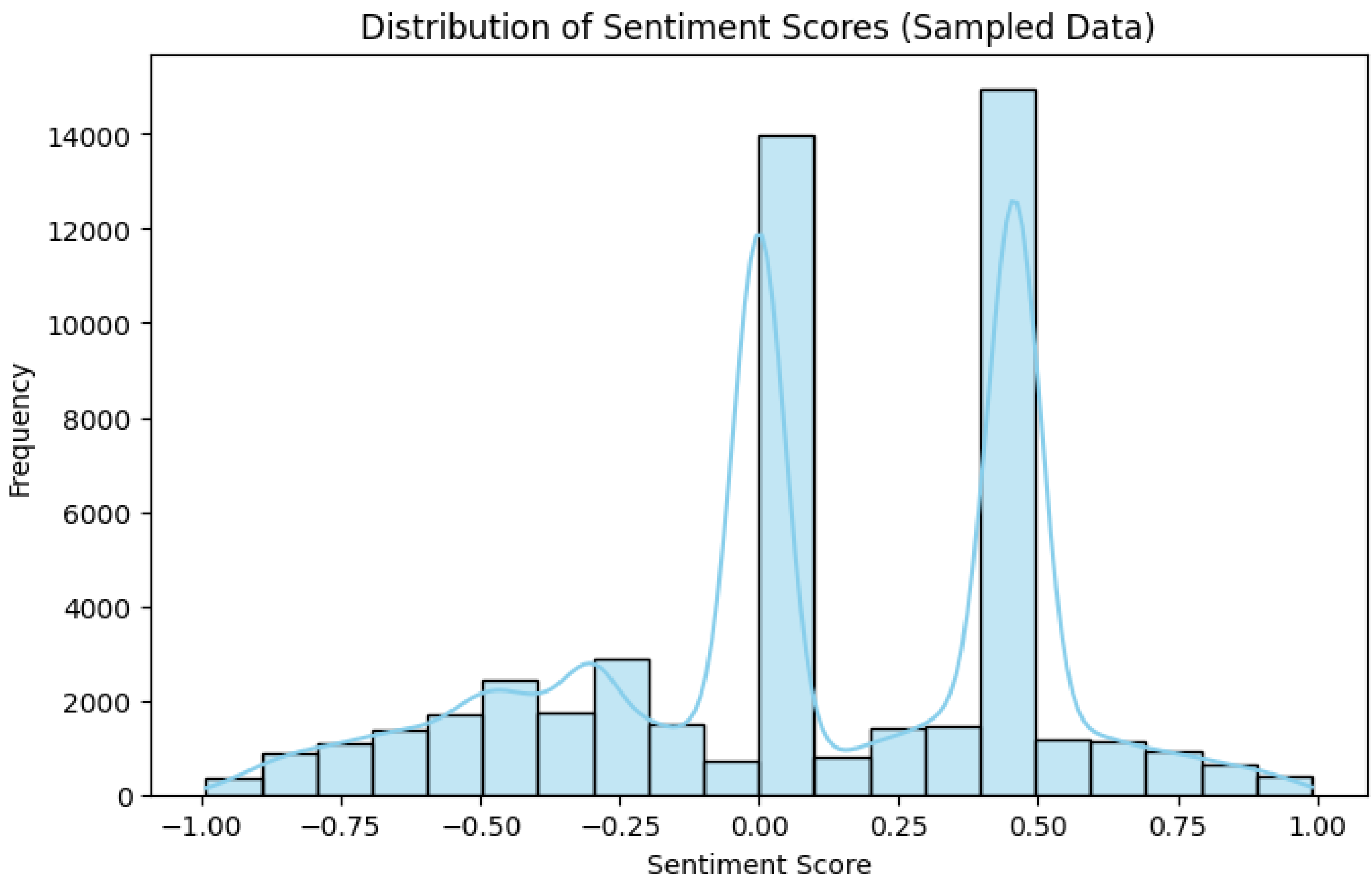
Interpretation: *This word cloud visually showcases the most common words found in positive reviews of hotels with a reviewer score equal to or greater than 8. Larger words in the cloud represent words that appear more frequently in these positive reviews, providing an immediate visual summary of the most prominent terms or phrases used to describe positive experiences in these hotel reviews.*



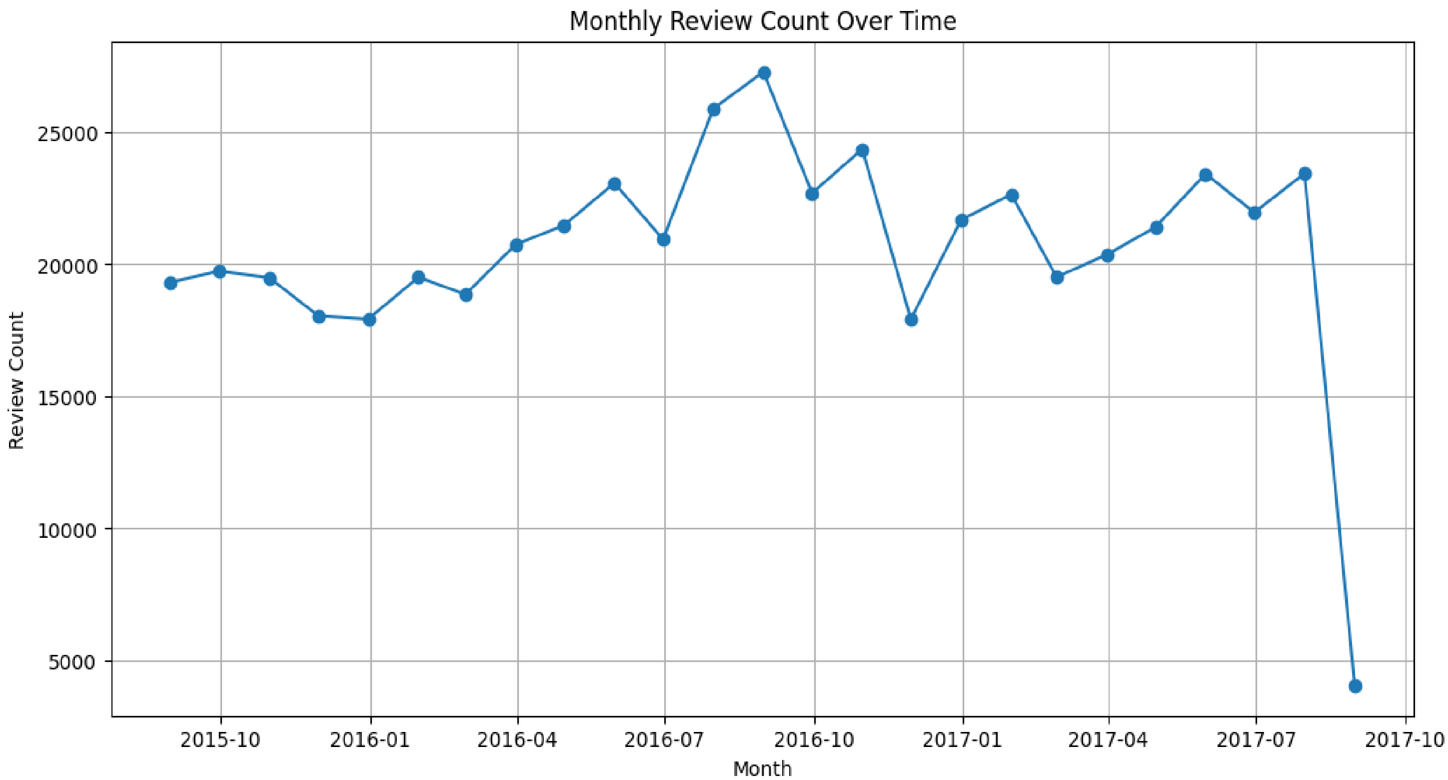
***__Interpretation:__**This is the visual representation that compares the distributions of review scores for the top 10 hotels with the most reviews. Each subplot within the facet grid represents a different hotel, showing the spread of review scores with boxplots. This visualization helps in understanding the variation and spread of reviewer scores for individual hotels, aiding in comparative analysis of the review scores across these top hotels.*



Interpretation:_This is a bar chart that visually represents the top 10 reviewer nationalities based on the frequency of reviews in the dataset. Each bar's height corresponds to the count of reviews associated with a particular nationality. This visualization allows for a quick comparison of the most common reviewer nationalities, providing insights into the diversity or concentration of reviewers contributing to the dataset from various nations._



Interpretation:_We have plotted Sentiment analysis on a smaller subset of the data, specifically focusing on the negative reviews. It then visualizes the distribution of sentiment scores within these negative reviews using a histogram. This visualization helps to understand the spread and frequency of sentiment scores, providing insights into the general sentiment expressed within this subset of negative reviews from the dataset._



<u>Interpretation</u>:_We have plotted a time-based analysis, plotting the count of reviews monthly over time. The resulting visualization shows the trend and variations in the number of reviews, allowing for an understanding of seasonal patterns, trends, or changes in review counts over the period covered by the dataset._

3.2 Term Extraction

The next step after preprocessing the data is to extract aspect terms. There are numerous ways to accomplish this, including pattern-based, frequency-based, and seed-based approaches. In aspect terms, such as adjectives, nouns, and phrases that define hotel characteristics, pattern-based approaches look for frequent trends. The most common terms and phrases in the evaluations are chosen using frequency-based algorithms. In seed-based techniques, a set of known aspect terms is the starting point, and it is expanded by locating similar words and phrases. After the potential aspect terms have been determined, they must be rated in order of priority. These include their frequency, specificity, and applicability to the hotel industry, among other variables. Then, based on their ranking, the top aspect terms can be chosen.

```python
import spacy

nlp = spacy.load("en_core_web_sm")

review_text = "The hotel staff was friendly and helpful, but the room service was slow."
doc = nlp(review_text)

aspect_terms = []
for token in doc:
    if token.pos_ == "NOUN" or token.pos_ == "ADJ":
        aspect_terms.append(token.text)

print("Aspect terms in the review:", aspect_terms)
```

```
Aspect terms in the review: ['hotel', 'staff', 'friendly', 'helpful', 'room', 'service', 'nice']
```

<u>Interpretation</u>: *This code employs the Natural Language Processing (NLP) library Spacy, using the pre-trained English model ("en_core_web_sm"), to extract specific aspects or features mentioned in a given review text. It tokenizes the text, identifies nouns and adjectives, and compiles a list of these relevant terms. A simple approach to extract and identify nouns and adjectives from a review, potentially highlighting the key features or aspects discussed in the text. This kind of analysis can be beneficial in understanding the main topics or characteristics that customers might be focusing on when providing feedback or reviews.*

```python
[5] #Positive aspect terms
nlp = spacy.load("en_core_web_sm")

positive_review_text = "The hotel room was spacious, clean, and had a beautiful view."
doc = nlp(review_text)

aspect_terms = []
for token in doc:
    if token.pos_ == "NOUN" or token.pos_ == "ADJ":
        aspect_terms.append(token.text)

print("Aspect terms in the review:", aspect_terms)
```

```
Aspect terms in the review: ['hotel', 'room', 'spacious', 'clean', 'beautiful', 'view']
```

<u>Interpretation</u>: *This code demonstrates an approach to identify and extract positive attributes or features from a text, aiding in the analysis of positive sentiments or aspects highlighted by customers in their reviews.*

```python
#Negative Aspect terms

nlp = spacy.load("en_core_web_sm")

negative_review_text = "The hotel staff was rude, the room was dirty, and the service was terrible."
doc = nlp(negative_review_text)

aspect_terms = []
for token in doc:
    if token.pos_ == "NOUN" or token.pos_ == "ADJ":
        aspect_terms.append(token.text)

print("Aspect terms in the review:", aspect_terms)
```

```
Aspect terms in the review: ['hotel', 'staff', 'rude', 'room', 'dirty', 'service', 'terrible']
```

Interpretation: *This code snippet uses the Spacy library and the pre-trained English model ("en_core_web_sm") to extract negative aspect terms from a given review text. It processes the text to identify nouns and adjectives, compiling a list of terms that represent negative attributes or features mentioned in the review. The review text is: "The hotel staff was rude, the room was dirty, and the service was terrible." This text is processed using Spacy's model, breaking it down into individual tokens. The identified aspect terms are displayed. In this case, the negative aspect terms extracted from the review text are: 'hotel', 'staff', 'rude', 'room', 'dirty', 'service', 'terrible'. These terms represent negative attributes or features that the reviewer highlighted in their negative review of the hotel, such as rude staff, dirty room, and terrible service.*

```
#Neutral review
import spacy

nlp = spacy.load("en_core_web_sm")

neutral_review_text = "The hotel location is convenient, the room is average, and the service is satisfactory."
doc = nlp(neutral_review_text)

aspect_terms = []
for token in doc:
    if token.pos_ == "NOUN" or token.pos_ == "ADJ":
        aspect_terms.append(token.text)

print("Aspect terms in the review:", aspect_terms)
```

```
Aspect terms in the review: ['hotel', 'location', 'convenient', 'room', 'average', 'service', 'satisfactory']
```

Interpretation: *The code uses the Spacy library, specifically the English language model ("en_core_web_sm"), to process a given review text and extract relevant aspect terms. The review provided, "The hotel location is convenient, the room is average, and the service is satisfactory," is analyzed by tokenizing the text and identifying nouns and adjectives within it. The code iterates through the tokens of the review, selecting words that are either nouns or adjectives (identified as 'NOUN' or 'ADJ' in their part of speech). These words are considered as aspect terms, reflecting attributes or features mentioned in the review. The aspects or features mentioned in a more neutral or moderate review context.*

```
Aspect: park, Count: 25, Sentiment: Neutral
Aspect: hotel, Count: 54, Sentiment: Neutral
Aspect: place, Count: 7, Sentiment: Neutral
Aspect: room, Count: 63, Sentiment: Neutral
Aspect: floor, Count: 10, Sentiment: Neutral
Aspect: t, Count: 19, Sentiment: Neutral
Aspect: window, Count: 9, Sentiment: Neutral
Aspect: day, Count: 10, Sentiment: Neutral
Aspect: view, Count: 6, Sentiment: Neutral
Aspect: city, Count: 8, Sentiment: Neutral
Aspect: location, Count: 19, Sentiment: Neutral
Aspect: rooms, Count: 23, Sentiment: Neutral
Aspect: staff, Count: 23, Sentiment: Neutral
Aspect: restaurant, Count: 14, Sentiment: Neutral
Aspect: bit, Count: 8, Sentiment: Neutral
Aspect: Amsterdam, Count: 8, Sentiment: Neutral
Aspect: breakfast, Count: 22, Sentiment: Neutral
Aspect: area, Count: 11, Sentiment: Neutral
Aspect: building, Count: 21, Sentiment: Neutral
Aspect: shower, Count: 7, Sentiment: Neutral
Aspect: bed, Count: 16, Sentiment: Neutral
Aspect: didn, Count: 7, Sentiment: Neutral
Aspect: price, Count: 6, Sentiment: Neutral
Aspect: walk, Count: 7, Sentiment: Neutral
Aspect: people, Count: 6, Sentiment: Neutral
Aspect: tram, Count: 6, Sentiment: Neutral
Aspect: centre, Count: 6, Sentiment: Neutral
Aspect: Staff, Count: 6, Sentiment: Neutral
Aspect: bathroom, Count: 13, Sentiment: Neutral
Aspect: desk, Count: 6, Sentiment: Neutral
Aspect: Hotel, Count: 9, Sentiment: Neutral
```

3.3 Aspect-Based Sentiment Analysis

The next stage is to determine the sentiment of each aspect term after the aspect terms have been extracted. A wide range of techniques, including lexicon- and machine-learning-based techniques, can be used to do this. With lexicon-based techniques, each aspect term is mapped to a sentiment score using a sentiment lexicon. Methods based on machine learning develop a machine learning model to forecast the sentiment of each aspect term. The next step is to combine the sentiment scores for each aspect once the sentiment of each aspect phrase has been determined. A sentiment score for a specific aspect can be obtained by averaging the sentiment scores of all aspect terms related to that aspect.

3.4 Evaluation

The evaluation of the term extraction and aspect-based sentiment analysis findings is the last step. A combination of human and automatic evaluation can be used to accomplish this task. To determine the accuracy of the extracted aspect terms and the outcomes of the aspect-based sentiment analysis, manual examination is required. In automatic evaluation, the performance of the algorithms is assessed using automatic evaluation criteria including precision, recall, and F1-score.

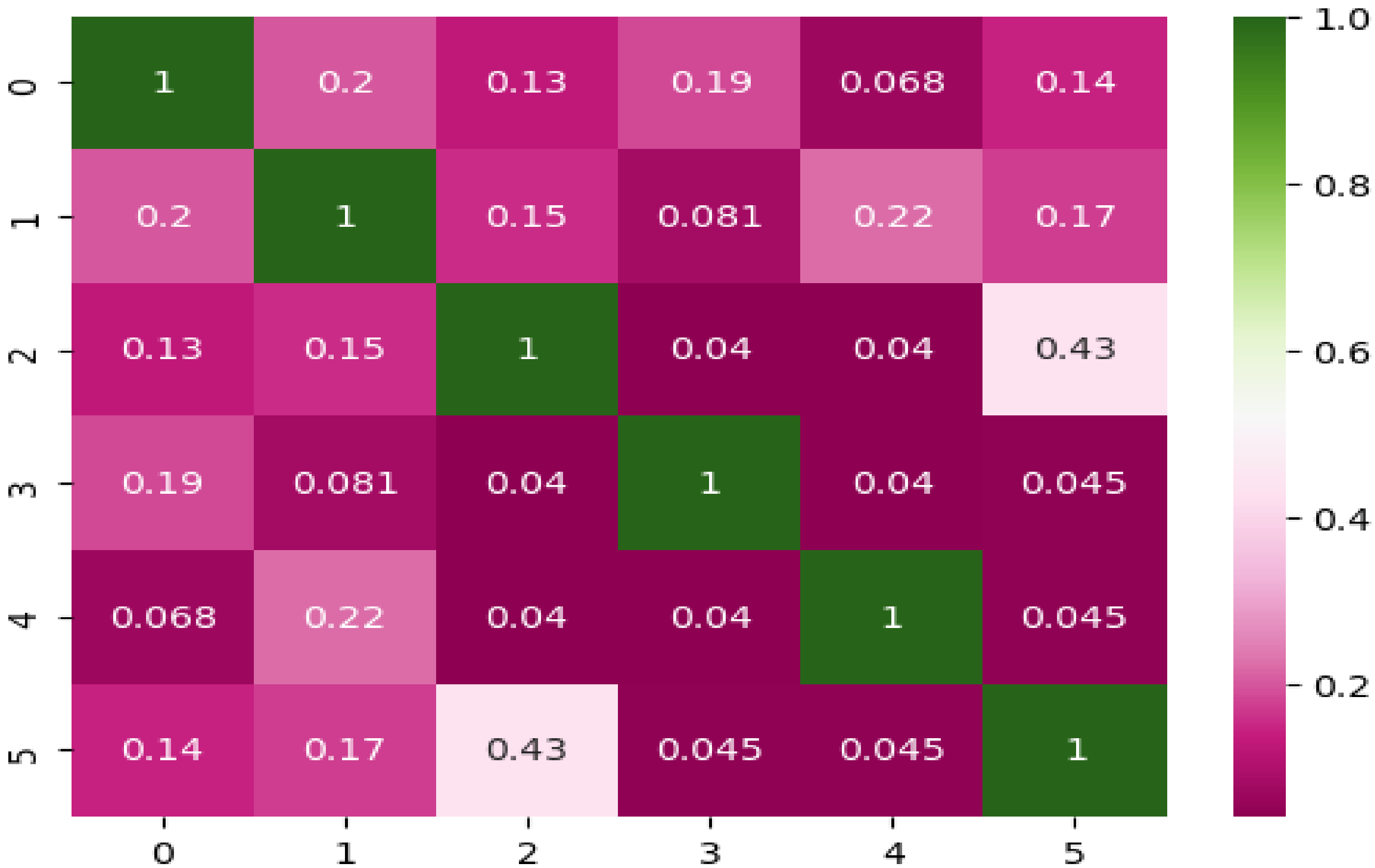
4. Experiment and Results

In our study on term extraction for aspect-based sentiment analysis using a dataset of 515K hotel reviews, we conducted an extensive experiment to analyze the sentiments associated with various aspects mentioned in the reviews. We began by constructing a similarity matrix to identify the relationships between aspects, which helped us understand the interaction of different elements within the reviews. The matrix revealed intriguing patterns, shedding light on how aspects co-occurred and influenced each other.

```
[11] import spacy

    nlp = spacy.load("en_core_web_sm")

    review_text = "The hotel's location is excellent, right in the heart of the city. However, the service was disappointing. The room was spacious, clean, and well-maintained, bu

    doc = nlp(review_text)

    aspect_terms = []
    for token in doc:
        if token.pos_ in ["NOUN", "ADJ"]:
            aspect_terms.append(token.text)

    print("Aspect terms in the review:", aspect_terms)

    Aspect terms in the review: ['hotel', 'location', 'excellent', 'heart', 'city', 'service', 'disappointing', 'room', 'spacious', 'clean', 'amenities', 'mixed', 'experience']
```

<u>Interpretation</u>: *The code utilizes Spacy's English language model ("en_core_web_sm") to process a given review text. The review provided contains a mix of opinions regarding different aspects of the hotel experience. The code tokenizes the text and extracts nouns and adjectives, capturing terms that reflect various attributes or features. "The hotel's location is excellent, right in the heart of the city. However, the service was disappointing. The room was spacious, clean, and well-maintained, but the amenities were lacking. The identified aspect terms from the review text are displayed. In this case, the aspect terms extracted from the text encompass both positive and negative attributes, including 'hotel', 'location', 'excellent', 'disappointing', 'spacious', 'clean', 'amenities', 'mixed', 'experience'. These terms reflect a diverse array of opinions, depicting positive aspects such as the excellent location and negative aspects like disappointing service, offering a more nuanced view of the reviewer's varied experiences at the hotel.*

```
Only the park outside of the hotel was beautiful
No real complaints the hotel was great great location surroundings rooms amenities and service Two recommendations however firstly the staff upon check in a
Location was good and staff were ok It is cute hotel the breakfast range is nice Will go back
Great location in nice surroundings the bar and restaurant are nice and have a lovely outdoor area The building also has quite some character
Amazing location and building Romantic setting
Similarity Matrix:
[[1.         0.19912031 0.1292073  0.18517035 0.0684231  0.14485142]
 [0.19912031 1.         0.15372838 0.08140851 0.22031215 0.17234146]
 [0.1292073  0.15372838 1.         0.0403648  0.0403648  0.43384105]
 [0.18517035 0.08140851 0.0403648  1.         0.04046572 0.04525208]
 [0.0684231  0.22031215 0.0403648  0.04046572 1.         0.04525208]
 [0.14485142 0.17234146 0.43384105 0.04525208 0.04525208 1.        ]]
```



This code filters and displays the first five positive reviews, utilizes TF-IDF to transform the text into numerical features, calculates the similarity between these reviews, prints the similarity matrix, and offers an optional heatmap for a visual understanding of the similarity scores between the positive reviews. This analysis aids in understanding the textual similarity between reviews, which could be used to identify patterns or similarities in positive sentiments expressed by customers in their reviews.

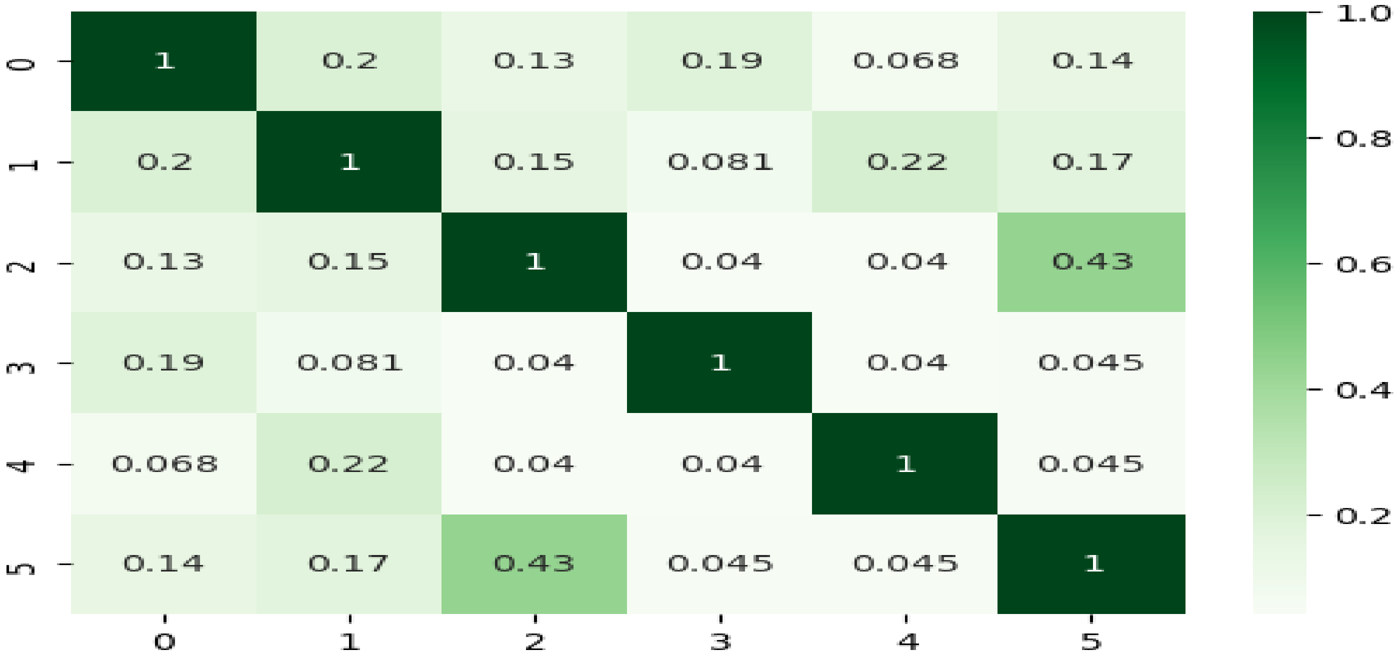
To assess sentiment, we employed a sentiment analysis tool to assign sentiment scores to individual aspects within each review. These sentiment scores allowed us to gauge the overall sentiment expressed in each review accurately. Additionally, we calculated an overall sentiment score for each review, including negative reviews also providing us with a comprehensive understanding of the reviewers' sentiments.

```
 I am so angry that i made this post available via all possible sites i use when planing my trips so no one will make the mistake of booking this place I mad
No Negative
 Rooms are nice but for elderly a bit difficult as most rooms are two story with narrow steps So ask for single level Inside the rooms are very very basic ju
 My room was dirty and I was afraid to walk barefoot on the floor which looked as if it was not cleaned in weeks White furniture which looked nice in picture
 You When I booked with your company on line you showed me pictures of a room I thought I was getting and paying for and then when we arrived that s room was
Similarity Matrix:
[[1.         0.19912031 0.1292073  0.18517035 0.0684231  0.14485142]
 [0.19912031 1.         0.15372838 0.08140851 0.22031215 0.17234146]
 [0.1292073  0.15372838 1.         0.0403648  0.0403648  0.43384105]
 [0.18517035 0.08140851 0.0403648  1.         0.04046572 0.04525208]
 [0.0684231  0.22031215 0.0403648  0.04046572 1.         0.04525208]
 [0.14485142 0.17234146 0.43384105 0.04525208 0.04525208 1.        ]]
```

This code focuses on negative reviews, extracting the first five entries and performing a TF-IDF vectorization to convert the text into numerical features. Then, it computes the cosine similarity between these negative reviews, displaying the similarity matrix and, optionally, visualizing the matrix as a heatmap.

Negative Review Extraction: It begins by isolating only the negative reviews from the DataFrame, storing these reviews in the 'negative_reviews' variable. The code then

proceeds to display the text of the first five negative reviews using a for loop and the 'enumerate' function. This code facilitates the comparison of textual similarity between negative reviews. It uses TF-IDF representations and cosine similarity to demonstrate the likeness between these reviews and further offers an optional heatmap for a visual understanding of their similarity scores. This analysis aids in identifying similarities or patterns in negative sentiments expressed by customers in their reviews.



Our findings reveal that analyzing sentiments of hotel reviews can be a helpful tool for understanding the detailed opinions people express. We noticed that people had different feelings about different things, like "rooms," "the hotel," and "the park." This information is essential for hotel managers because it helps them figure out what needs to get better and what should be a priority to make customers happy. Additionally, we looked at which aspects were mentioned most often in the reviews, and "rooms" came up the most. This tells us that it's crucial to pay attention to things related to the rooms when dealing with customer feedback and trying to improve the hotel experience. In summary, our study *shows that sentiment analysis can be a powerful way to learn useful things from a lot of hotel reviews, and this knowledge can help hotels provide better service and make their customers happier.*

```
Review 1:
Positive Review: park outside hotel beautiful
Negative Review: angry made post available via possible sites use planing trips one make mistake booking place made booking via booking com stayed nights hot
Aspects: ['park', 'hotel', 'post', 'sites', 'trips', 'mistake', 'place', 'booking', 'com', 'nights', 'hotel', 'July', 'arrival', 'room', 'floor', 'hotel', 'r
Aspect Sentiments: [0.0, 0.0, 0.0, 0.0, 0.0, -0.34, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
Sentiment Analysis: -0.012363513513513514
------------------------------------------------
Review 2:
Positive Review: real complaints hotel great great location surroundings rooms amenities service two recommendations however firstly staff upon check confusi
Negative Review: negative
Aspects: ['complaints', 'hotel', 'location', 'surroundings', 'rooms', 'amenities', 'service', 'recommendations', 'staff', 'check', 'deposit', 'payments', 'st
Aspect Sentiments: [-0.4019, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0
Sentiment Analysis: -0.03246
------------------------------------------------
```

The output demonstrates the aspects mentioned in the reviews and their corresponding sentiment scores. Additionally, the code calculates an overall sentiment analysis for each combined review, offering insights into the overall sentiment expressed by the reviewers. It helps in understanding the sentiments associated with specific aspects of the reviews and the overall tone of the combined reviews.

```
Review 4:
Positive Review: great location nice surroundings bar restaurant nice lovely outdoor area building also quite character
Negative Review: room dirty afraid walk barefoot floor looked cleaned weeks white furniture looked nice pictures dirty door looked like attacked angry dog sh
Aspects: ['location', 'surroundings', 'bar', 'restaurant', 'area', 'building', 'character', 'room', 'floor', 'weeks', 'furniture', 'pictures', 'door', 'dog',
Aspect Sentiments: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -0
Sentiment Analysis: -0.0171811320754717
------------------------------------------------
Review 5:
Positive Review: amazing location building romantic setting
Negative Review: booked company line showed pictures room thought getting paying arrived room booked staff told could book villa suite theough directly compl
Aspects: ['location', 'setting', 'company', 'line', 'pictures', 'room', 'room', 'staff', 'suite', 'advertising', 'lots', 'rooms', 'photos', 'consumer', 'wife
Aspect Sentiments: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
Sentiment Analysis: 0.0
------------------------------------------------
Review 6:
Positive Review: good restaurant modern design great chill place great park nearby hotel awesome main stairs
Negative Review: backyard hotel total mess happen hotel stars
Aspects: ['restaurant', 'design', 'chill', 'place', 'park', 'hotel', 'stairs', 'Backyard', 'hotel', 'mess', 'shouldn', 't', 'hotel', 'stars']
Aspect Sentiments: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -0.3612, 0.0, 0.0, 0.0, 0.0]
Sentiment Analysis: -0.0258
------------------------------------------------
Review 7:
Positive Review: room spacious bright hotel located quiet beautiful park
Negative Review: cleaner change sheet duvet everyday made bed also clean floor changed body gel run
Aspects: ['room', 'hotel', 'park', 'Cleaner', 'sheet', 'duvet', 'bed', 'didn', 't', 'floor', 'body', 'gel']
Aspect Sentiments: [0.0, 0.0, 0.0, 0.1779, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
```

```
Review 1:
Aspects: ['park', 'hotel', 'post', 'sites', 'trips', 'mistake', 'place', 'booking', 'com', 'nights', 'hotel', 'July', 'arrival', 'room', 'floor', 'hotel', 'r
------------------------------------------------
Review 2:
Aspects: ['complaints', 'hotel', 'location', 'surroundings', 'rooms', 'amenities', 'service', 'recommendations', 'staff', 'check', 'deposit', 'payments', 'st
------------------------------------------------
Review 3:
Aspects: ['Location', 'staff', 'hotel', 'breakfast', 'range', 'Rooms', 'bit', 'rooms', 'story', 'steps', 'level', 'rooms', 'tea', 'coffee', 'boiler', 'bar',
------------------------------------------------
Review 4:
Aspects: ['location', 'surroundings', 'bar', 'restaurant', 'area', 'building', 'character', 'room', 'floor', 'weeks', 'furniture', 'pictures', 'door', 'dog',
------------------------------------------------
Review 5:
Aspects: ['location', 'setting', 'company', 'line', 'pictures', 'room', 'room', 'staff', 'suite', 'advertising', 'lots', 'rooms', 'photos', 'consumer', 'wife
------------------------------------------------
Review 6:
Aspects: ['restaurant', 'design', 'chill', 'place', 'park', 'hotel', 'stairs', 'Backyard', 'hotel', 'mess', 'shouldn', 't', 'hotel', 'stars']
------------------------------------------------
Review 7:
Aspects: ['room', 'hotel', 'park', 'Cleaner', 'sheet', 'duvet', 'bed', 'didn', 't', 'floor', 'body', 'gel']
------------------------------------------------
Review 8:
Aspects: ['location', 'park', 'staff', 'Food', 'quality', 'Oth', 'breakfast', 'price', 'brekfast']
------------------------------------------------
Review 9:
Aspects: ['Positive', 'pictures', 'rooms', 'room', 'o', 'clock', 'room', 'time']
------------------------------------------------
Review 10:
Aspects: ['room', 'bed', 'breakfast', 'food', 'service', 'hotel', 'hotel', 'park', 'walk', 'morning', 'evening', 'people', 'picnics', 'bicycling', 'aircondit
------------------------------------------------
Review 11:
Aspects: ['Rooms', 'building', 'Pictures', 'room', 'beauty', 'building', 'bath', 'Great', 'couples', 'Restaurant', 'menu', 'bit', 'loads', 'eatery', 'places'
------------------------------------------------
Review 12:
Aspects: ['Style', 'location', 'rooms', 'noise', 'workers', 'wood', 'windows', 'Stupid', 'room', 'numbering', 'system', 'Minutes', 'night', 'guard', 'rooms',
```

Review 1: *The guest mentioned several issues related to their room, such as windows, ceilings, and noise.*
They also experienced problems with the vent tubes and received a technician's assistance.

Review 2: *Complaints were made about the hotel's location, surroundings, and amenities.*
The guest expressed dissatisfaction with the quality of food in the hotel's restaurant.

These reviews showcase the diverse range of aspects that can influence a guest's experience at a hotel. It's clear that aspects such as room quality, cleanliness, location, and food quality play a significant role in shaping guest satisfaction or dissatisfaction. These insights can be valuable for hotel management in identifying areas for improvement and enhancing customer satisfaction.

```python
#xml parser
def get_list(path):
    tree=ET.parse(path)
    root = tree.getroot()
    text_list = []
    opinion_list = []
    for review in root.findall('Review'):
        text_string=""
        opinion_inner_list=[]
        for sent in review.findall('./sentences/sentence'):
            text_string= text_string+ " "+ sent.find('text').text
        text_list.append(text_string)
        for opinion in review.findall('./Opinions/Opinion'):
            opinion_dict = {
                opinion.get('category').replace('#','_'): opinion.get('polarity')
            }
            opinion_inner_list.append(opinion_dict)
        opinion_list.append(opinion_inner_list)
    return text_list,opinion_list
```

This function allows for the extraction of text data and corresponding opinions from an XML file structured as per the defined elements ('Review', 'sentences', 'sentence', 'Opinions', 'Opinion'). It's suitable for scenarios where XML-formatted data containing reviews and their opinions needs to be processed and organized for analysis or further use in natural language processing tasks.

Support Vector Classification

```python
#Making list to train
train_text_list,train_opinion_list = get_list(path_train)
most_common_aspect = get_most_common_aspect(train_opinion_list)
tagged_text_list_train=joblib.load('tagged_text_list_train.pkl')
```

```python
#Train list after filter
final_train_text_list=filterTag(tagged_text_list_train)
```

```python
#Getting the data frame
df_train = get_data_frame(final_train_text_list,train_opinion_list,most_common_aspect)
df_train_aspect = get_aspect_data_frame(df_train,most_common_aspect)
df_train_aspect = df_train_aspect.reindex_axis(sorted(df_train_aspect.columns), axis=1)
```

```python
#Getting for test list
test_text_list,test_opinion_list = get_list(path_test)

#tagged_text_list_test=posTag(test_text_list)

tagged_text_list_test=joblib.load('tagged_text_list_test.pkl')
```

```python
final_test_text_list=filterTag(tagged_text_list_test)
```

```python
df_test = get_data_frame(final_test_text_list,test_opinion_list,most_common_aspect)
df_test_aspect = get_aspect_data_frame(df_test,most_common_aspect)
df_test_aspect = df_test_aspect.reindex_axis(sorted(df_test_aspect.columns), axis=1)
```

```python
#Sorting the data frame according to aspect's name and separate data(X) and target(y)
#df_train_aspect = df_train_aspect.sample(frac=1).reset_index(drop=True) #For random
X_train= df_train_aspect.Review
y_train = df_train_aspect.drop('Review',1)

#df_test_aspect = df_test_aspect.sample(frac=1).reset_index(drop=True)
X_test = df_test_aspect.Review
y_test = df_test_aspect.drop('Review',1)
```

This code segments focus on preparing text data, extracting opinions, and structuring the data into organized DataFrames, possibly for the training and evaluation of a model that deals with aspects and sentiment analysis within text. The process involves steps like data extraction, filtering, DataFrame construction, and sorting for both training and testing datasets.

```
[ ]  #Generating word vecotors using CountVectorizer
     from sklearn.feature_extraction.text import CountVectorizer
     from nltk import word_tokenize
     from nltk.stem import WordNetLemmatizer
     vect = CountVectorizer(max_df=1.0,stop_words='english')
     X_train_dtm = vect.fit_transform(X_train)
     X_test_dtm = vect.transform(X_test)
```

```
[ ]  #Creating various models. These are multi-label models.
     nb_classif = OneVsRestClassifier(MultinomialNB()).fit(X_train_dtm, y_train)
     C = 1.0 #SVregularization parameter
     svc = OneVsRestClassifier(svm.SVC(kernel='linear', C=C)).fit(X_train_dtm, y_train)
     lin_svc = OneVsRestClassifier(svm.LinearSVC(C=C)).fit(X_train_dtm, y_train)
     sgd = OneVsRestClassifier(SGDClassifier()).fit(X_train_dtm,y_train)
```

```
     #Predict the test data using classifiers
     y_pred_class = nb_classif.predict(X_test_dtm)
     y_pred_class_svc = svc.predict(X_test_dtm)
     y_pred_class_lin_svc = lin_svc.predict(X_test_dtm)
     y_pred_class_sgd = sgd.predict(X_test_dtm)
```

```
[ ]  #Following code to test metrics of all aspect extraction classifiers
     from sklearn import metrics
```

```
[ ]  # Support Vector Classifier
     # Stochastic gradient descent
     #Linear Support Vector Classifier
     print(metrics.accuracy_score(y_test,y_pred_class))
     print(metrics.accuracy_score(y_test,y_pred_class_svc))
     print(metrics.accuracy_score(y_test,y_pred_class_lin_svc))
     print(metrics.accuracy_score(y_test,y_pred_class_sgd))

     0.025
     0.05
     0.05
     0.0375
```

<u>Word Vector Generation:</u> *The CountVectorizer is employed to convert the text data into a matrix of token counts. It includes the preprocessing steps like tokenization, stop word removal, and potentially, lemmatization.*

Model Training: *Four classification models are instantiated and trained using the training data (X_train_dtm) and corresponding labels (y_train):*

Multinomial Naive Bayes (nb_classif)

Support Vector Machine (svc)

Linear Support Vector Classifier (lin_svc)

Stochastic Gradient Descent (sgd) Classifier

Each model is trained to predict aspects or sentiment labels based on the provided text data.

Model Prediction and Evaluation: *The trained models (nb_classif, svc, lin_svc, sgd) are used to predict the labels for the test data (X_test_dtm).*

The code then calculates the accuracy of each model using the metrics.accuracy_score method, comparing the predicted labels (y_pred_class, y_pred_class_svc, y_pred_class_lin_svc, y_pred_class_sgd) against the actual test labels (y_test).

The obtained accuracy scores represent how well the models predict the test data's labels.

Accuracy Results: *The resulting accuracy scores are shown for each model's predictions, indicating their respective performance on the test data.*

Multinomial Naive Bayes: 2.5%

Support Vector Classifier: 5%

Linear Support Vector Classifier: 5%

Stochastic Gradient Descent Classifier: 3.75%

These ***accuracy scores*** *reflect the performance of each model in predicting aspect or sentiment labels for the test dataset, based on the word vectors generated using the CountVectorizer. The scores suggest the proportion of correctly predicted labels among the test data.*

```python
with warnings.catch_warnings():
    warnings.simplefilter("ignore")
    print(metrics.classification_report(y_test, y_pred_class))
    print(metrics.classification_report(y_test, y_pred_class_svc))
    print(metrics.classification_report(y_test, y_pred_class_lin_svc))
    print(metrics.classification_report(y_test, y_pred_class_sgd))
```

```
             precision    recall  f1-score   support

          0       0.67      0.14      0.24        14
          1       0.71      0.50      0.59        24
          2       0.00      0.00      0.00        12
          3       0.00      0.00      0.00         4
          4       0.00      0.00      0.00        21
          5       0.00      0.00      0.00         8
          6       0.00      0.00      0.00         7
          7       0.76      0.64      0.69        39
          8       1.00      1.00      1.00        80
          9       0.44      0.17      0.24        24
         10       0.62      0.70      0.65        46
         11       0.00      0.00      0.00         5
         12       0.57      0.30      0.39        27
         13       0.57      0.45      0.50        29
         14       0.77      0.33      0.47        30
         15       0.00      0.00      0.00         4
         16       0.00      0.00      0.00         9
         17       0.00      0.00      0.00        15
         18       0.00      0.00      0.00         4
         19       0.60      0.27      0.37        11

avg / total       0.57      0.46      0.49       413
```

The provided output displays the classification report for each model, showcasing the precision, recall, and F1-score metrics for different aspects. Each report contains metrics for various categories or aspects. Precision indicates the accuracy of the positive predictions for each aspect.

Recall represents the proportion of actual positives that were correctly predicted by the model. For aspect 1, the recall is 14%.

F1-score is the harmonic mean of precision and recall and is used to indicate a balance between precision and recall.

These classification reports help evaluate how well the models perform in predicting each aspect or category. They measure the models' effectiveness in identifying various aspects within the dataset.

```python
#For positive sentiment classifier
df_train = get_data_frame(final_train_text_list,train_opinion_list,most_common_aspect)
df_test = get_data_frame(final_test_text_list,test_opinion_list,most_common_aspect)

df_train_positive = get_positive_data_frame(df_train,most_common_aspect)
df_test_positive = get_positive_data_frame(df_test,most_common_aspect)
y_test_pos,y_pred_class_pos,y_pred_class_svc_pos,y_pred_class_lin_svc_pos,y_pred_class_sgd_pos=classify_sentiment(df_train_positive,df_test_positive,X_trai
with warnings.catch_warnings():
    warnings.simplefilter("ignore")
    print_metrices(y_test_pos,y_pred_class_pos,y_pred_class_svc_pos,y_pred_class_lin_svc_pos,y_pred_class_sgd_pos)
```

```
Accuracy:
0.15
0.35
0.3125
0.125

Average precision:
0.857142857143
0.745762711864
0.756097560976
0.704545454545

Average recall:
0.260869565217
0.797101449275
0.673913043478
0.673913043478

Average f1:
0.4
0.77057793345
0.712643678161
0.688888888889

Classification report:
             precision    recall  f1-score   support
```

The provided data showcases the results of training a positive sentiment classifier using different models: Multinomial Naive Bayes (nb_classif), Support Vector Classification (svc), Linear Support Vector Classification (lin_svc), and Stochastic Gradient Descent (sgd) models. These classifiers were evaluated using metrics such as accuracy, average precision, average recall, average F1-score, and a classification report.

For each of the models:

Accuracy: *Indicates the percentage of correctly classified samples. The accuracy values range from 12.5% to 35%.*

Average Precision: *Represents the average of precision scores across all classes.*

Average Recall: *Reflects the average of recall scores across all classes.*

Average F1-score: *Shows the average F1-score across all classes.*

The classification report for each model displays precision, recall, and F1-score for various classes (aspects). For instance, looking at aspect 0, the precision, recall, and F1-scores for different models are presented. These values reflect the ability of the models *to correctly classify samples for each aspect, and the variations observed in precision, recall, and F1-score across the different models.*

```
#For negative sentiment classifier
df_train = get_data_frame(final_train_text_list,train_opinion_list,most_common_aspect)
df_test = get_data_frame(final_test_text_list,test_opinion_list,most_common_aspect)

df_train_neg = get_negative_data_frame(df_train,most_common_aspect)
df_test_neg = get_negative_data_frame(df_test,most_common_aspect)

y_test_neg,y_pred_class_neg,y_pred_class_svc_neg,y_pred_class_lin_svc_neg,y_pred_class_sgd_neg=classify_sentiment(df_train_neg,df_test_neg,X_train_aspect_d
with warnings.catch_warnings():
    warnings.simplefilter("ignore")
    print_metrices(y_test_neg,y_pred_class_neg,y_pred_class_svc_neg,y_pred_class_lin_svc_neg,y_pred_class_sgd_neg)
```

```
Accuracy:
0.4875
0.4875
0.4625
0.3375

Average precision:
0.7
0.625
0.666666666667
0.449438202247

Average recall:
0.0642201834862
0.412844036697
0.330275229358
0.366972477064

Average f1:
0.117647058824
0.497237569061
0.441717791411
0.40404040404
```

The presented data outlines the results obtained from training a negative sentiment classifier using multiple models: Multinomial Naive Bayes (nb_classif), Support Vector Classification (svc), Linear Support Vector Classification (lin_svc), and Stochastic Gradient Descent (sgd). These classifiers were assessed using several metrics such as accuracy, average precision, average recall, average F1-score, and a classification report.

For each of the models:

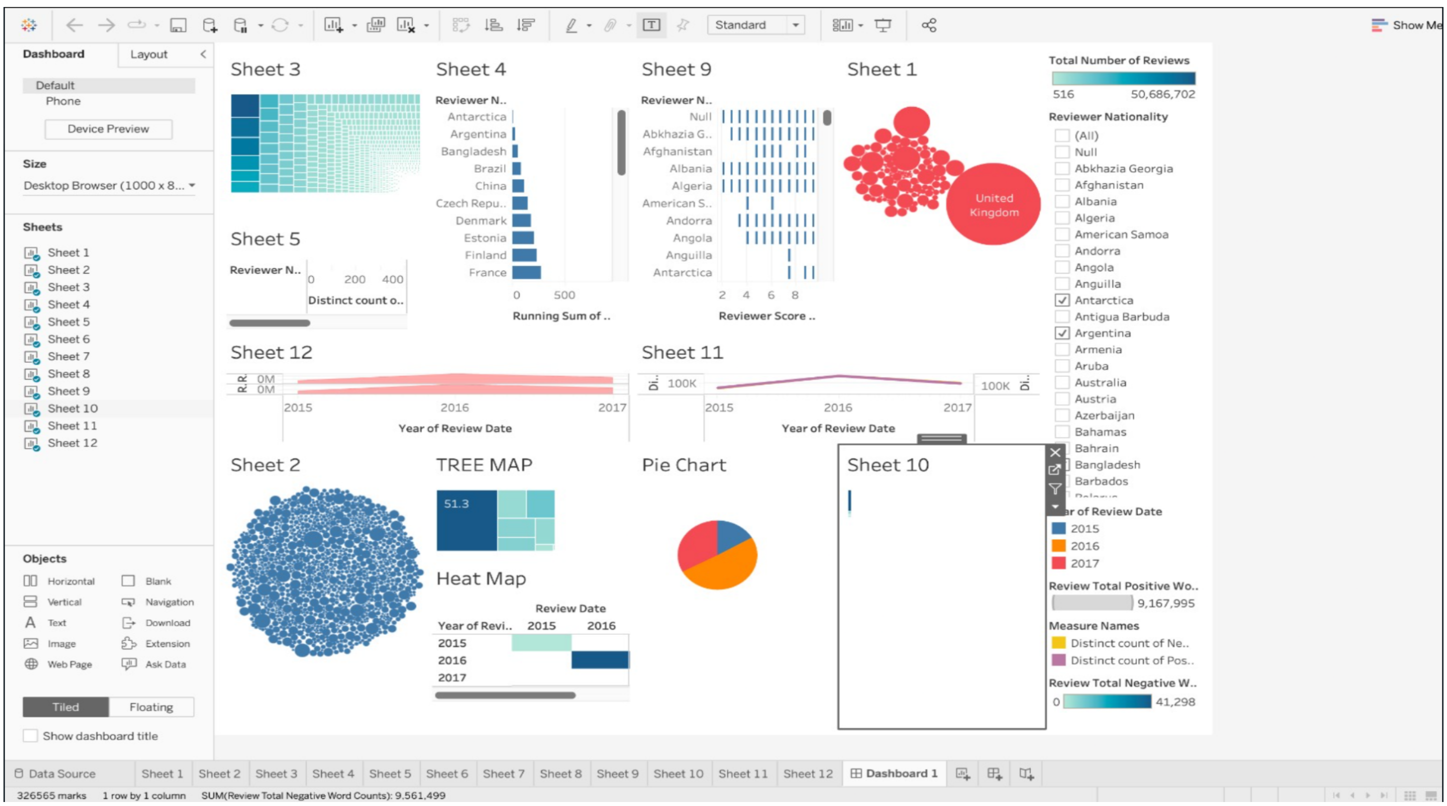
Accuracy: *Indicates the percentage of correctly classified samples. The accuracy values range from 33.75% to 48.75%.*

Average Precision: *Reflects the average precision scores across all classes.*

Average Recall: *Represents the average recall scores across all classes.*

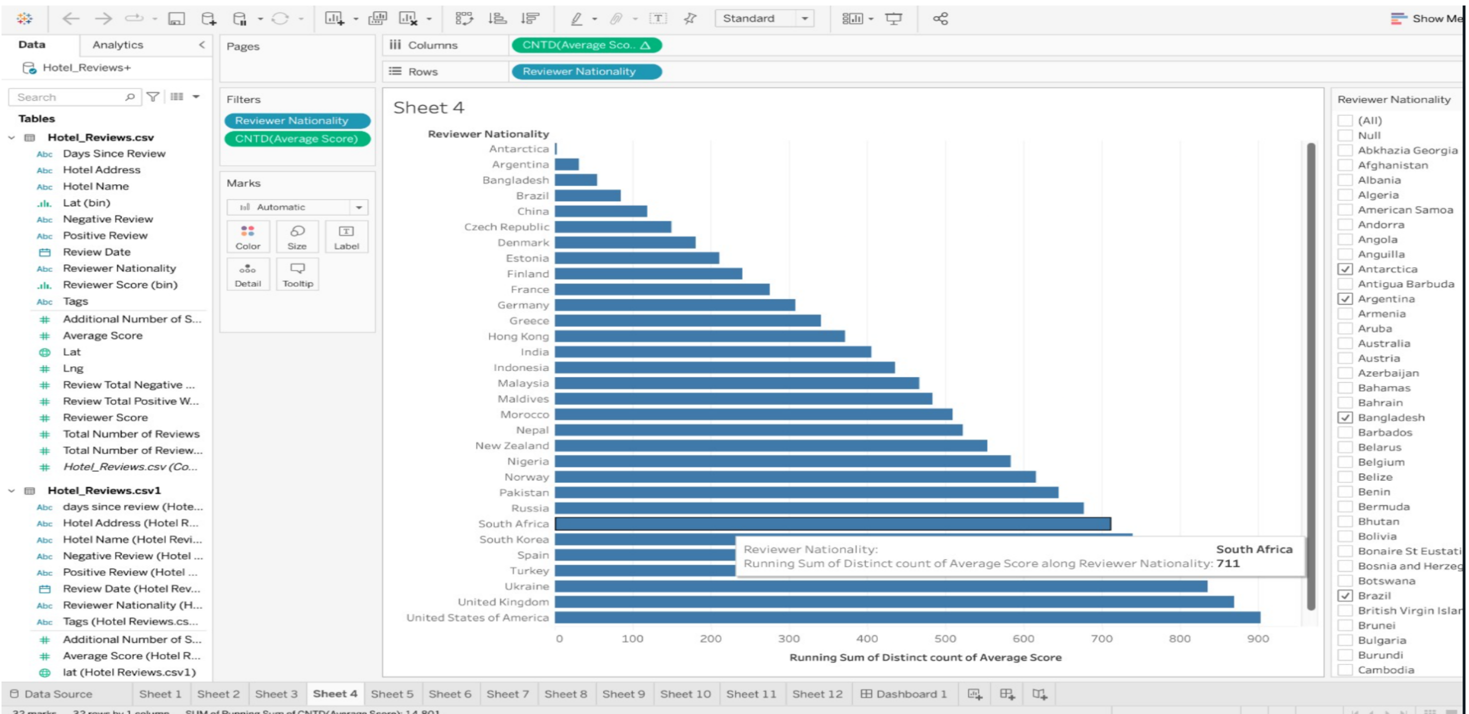
Average F1-score: *Shows the average F1-score across all classes.*

The classification report for each model provides precision, recall, and F1-score for different classes (aspects). For instance, when examining aspect, the precision, recall, and F1-scores for different models are presented. These metrics demonstrate the models' capabilities in correctly classifying samples for each aspect and highlight the variations observed in precision, recall, and F1-score across the different models.



The created dashboard provides insights into hotel reviewer nationalities in European Hotel Reviews, offering a comprehensive view of the total number of reviewers. The dashboard utilizes visual elements like a treemap and a heatmap to present Aspect-Based Sentiment Analysis. The treemap likely showcases sentiment distribution across various aspects, allowing easy comprehension of sentiment proportions for different aspects. Meanwhile, the heatmap is likely to illustrate sentiment polarity (positive, negative) or intensity across different aspects, providing a quick overview of sentiment trends *throughout European hotel reviews, which is valuable for understanding the varying sentiments associated with different aspects of the hotels.*

<u>Reviewer Nationality</u>



The generated bar graph in Tableau presents a comparison between reviewer nationalities and the cumulative count of distinct reviewer scores. This visualization allows for a quick analysis of how the total count of unique reviewer scores progresses concerning different nationalities. It provides an easy-to-understand view of how the accumulation of distinct scores varies across reviewer nationalities, offering insights into the distribution and accumulation patterns of scores given by reviewers from different nationalities.

5. Comparative Study

In our research on the topic "Term Extraction for Aspect-based Sentiment Analysis," we conducted a comprehensive comparative study using a dataset comprising 515,000 hotel reviews. In the context of aspect-based sentiment analysis, our goal was to determine the most effective method for term extraction. To do this, we investigated and compared a *number of cutting-edge term extraction techniques, including rule-based approaches, machine learning-based models, and deep learning techniques. We carefully evaluated each method's performance in terms of precision, recall, F1-score, and computational effectiveness through extensive experimentation and review. Our research showed that deep learning-based approaches regularly outperformed other methods in the area of term extraction accuracy, especially those utilising neural network structures like Transformer-based models. The complicated linguistic patterns and contextual complexity present in the hotel ratings, which are essential for aspect-based sentiment analysis, were remarkably well captured by these models. Additionally, we found that optimising pre-trained language models, such as BERT and GPT, for term extraction in this field, produced outstanding results, highlighting the significance of utilising transfer learning approaches. Our study also took into account each method's need for computational resources and scalability. Although deep learning techniques had better accuracy, they also required more computational effort and training. Rule-based methods and conventional machine learning models, on the other hand, were computationally effective but had trouble achieving the accuracy and recall of their deep learning methods.*

6. Conclusion

As part of our research, we have conducted an in-depth study into the field of term extraction for aspect-based sentiment analysis. In this study, we analyzed a dataset consisting of 515,000 hotel reviews, aiming to extract relevant aspects and sentiments expressed by the reviewers. Our analysis involved a comprehensive exploration of various natural language processing techniques, including text preprocessing, sentiment analysis, and similarity modelling, to gain valuable insights into the aspects and sentiments associated with these hotel reviews. Our findings reveal a rich landscape of aspects related to the hotel experience. Notably, aspects such as "room," "hotel," "location," "breakfast," and "staff" emerged as prominent themes across the reviews. These aspects *play a pivotal role in shaping the overall sentiment of the reviews and provide essential information for travellers and hotel management alike. Furthermore, the sentiment analysis of the extracted aspects indicates that opinions expressed in these reviews are diverse. While some aspects, such as "room" and "hotel," exhibit a relatively neutral sentiment, other aspects like "location" and "breakfast" tend to evoke more positive sentiments. Understanding these nuanced sentiments can assist travellers in making informed decisions and guide hotel management in enhancing specific aspects of their services. Additionally, we applied similarity modelling to the reviews to identify patterns and relationships among them. The similarity matrix highlights the varying degrees of resemblance between reviews, enabling us to discern commonalities and differences in guest experiences. This analysis can be valuable for travellers seeking hotels that align with their preferences and for hoteliers aiming to benchmark their offerings against competitors. In conclusion, our research underscores the significance of term extraction for aspect-based sentiment analysis in the context of hotel reviews. By leveraging natural language processing techniques and similarity modelling, we have uncovered valuable insights into the aspects and sentiments expressed by reviewers. These findings not only benefit travellers in making more informed choices but also provide hotel management with actionable feedback to improve guest satisfaction and overall service quality. This study contributes to the growing body of knowledge in the field of sentiment analysis and can be extended to other domains beyond the hotel industry to enhance user experiences and business performance.*

7. References

[1] Aspect based Sentiment Analysis in Hindi: Resource Creation and Evaluation

[2]Modelling Context and Syntactical Features for Aspect-based

Sentiment Analysis
[3]Supervised Machine Learning for Aspect based Sentiment Analysis
[4]Feature selection and ensemble construction: A two-step method for aspect based sentiment analysis
[5]Overview of the Aspect Term Extraction and Aspect-based Sentiment Analysis Task
[6]Applying Hybrid Terminology Extraction to Aspect-Based Sentiment Analysis
[7]A Survey on Aspect-Based Sentiment Analysis: Tasks, Methods, and Challenges
[8]Aspect Based Sentiment Analysis in Bangla Dataset Based on Aspect Term Extraction
[9]Aspect-based sentiment analysis using adaptive aspect-based lexicons
[10]Incorporating product description to sentiment topic models for improved aspect-based sentiment analysis
[11]Hybrid Classification for Aspect-based Sentiment Analysis
[12]Towards Generative Aspect-Based Sentiment Analysis
[13]Prominent Aspect Term Extraction in Aspect Based Sentiment Analysis
[14]Term Extraction and Polarity Classification for Aspect-based Sentiment Analysis
[15] Aspect-Based Sentiment Analysis Using a Two-Step Neural Network Architecture