Google Maps

ASPECT-BASED SENTIMENT ANALYSIS FOR RESTAURANT REVIEWS

By: Tran Thi Kieu Trinh

28/06/2023

COMMITMENTS

I conducted this project independently from June 20, 2023, to June 28, 2023.

Author

Tran Thi Kieu Trinh

Contents

I. Introduction

In this section, I will provide a general introduction to my project and discuss the motivation behind it. There are several reasons why I chose this particular topic. Firstly, among the given options, I found that gathering reviews from Google Maps appeared to be the easiest process for me. However, the more important reason is that I noticed that the tags shown above the review section on Google Maps are quite useful in quickly providing an overview of the restaurant.

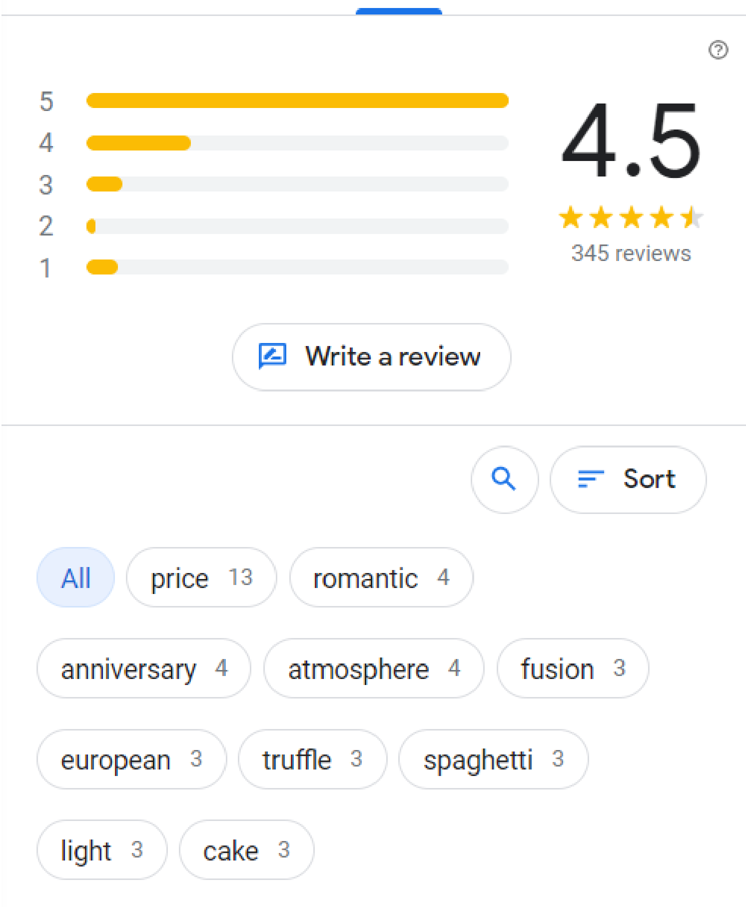

Figure 1: The tags above the review area

However, these tags only display the frequency of the keywords mentioned in the reviews, regardless of whether they are positive or negative. This means that if customers are interested in a particular tag, they have to read through all the reviews to gather more information.

Based on this observation, I aim to create some tags that can summarize customer reviews into sentiment score and the scores will then be attached to some special aspects that restaurant seeker often pay more attention to. In other words, the sentiment sore instead of frequency of each term should be displayed.

To do this, I will apply Machine Learning techniques in sentiment analysis and aspect extraction.

II. Methodology

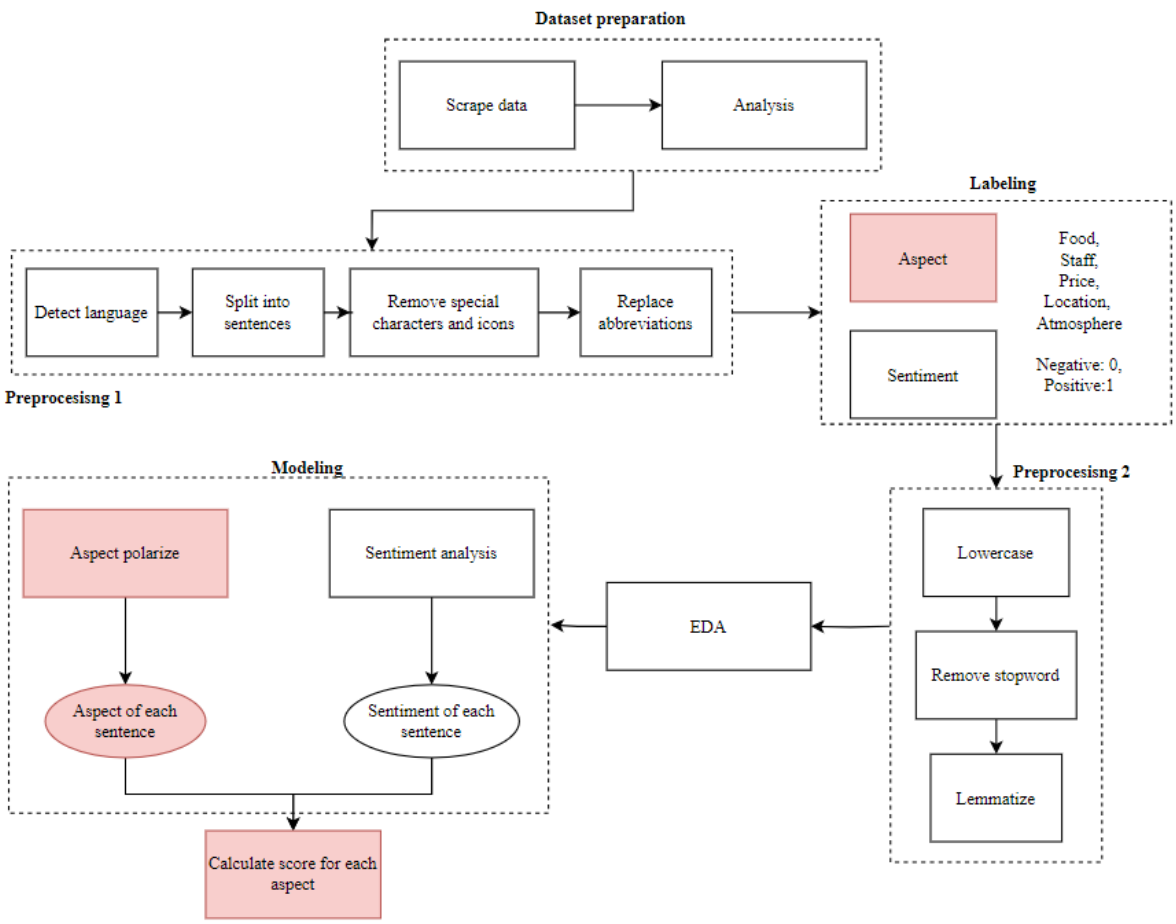

Figure 2: Process Diagram

**Due to a lack of time, I cannot complete all the steps. The steps marked in red is which I have not conducted yet.*

1. Data preparation

1.1. Scrape Data

To scrape data from Google Maps, I utilized Selenium and BeautifulSoup and the corresponding code is stored in the file named Scrape.ipynb. The scraping process involved several steps. Firstly, I searched for a specific district using a chosen list of districts including 'Tân Bình', 'Phú Nhuận', 'Bình Thạnh', 'Quận 5', and 'Quận 4' to focus on a particular area. Then, I used the keyword 'Nhà hàng' to search for restaurants in that area.

Subsequently, I carried out several steps to scrape restaurant information, including their address, ratings, rating counts, and reviews. These steps were carefully documented within the coding file.

1.2. General analysis

After completing the process of scraping data, the dataset comprises 120 restaurants with 11,208 (with 9308 non-null text) reviews. I conducted several analysis steps to gain an overview of the dataset, ***with the aim of identifying an appropriate labeling approach and determining any necessary preprocessing steps.***

- First, to find out aspects that suit this dataset I plotted a wordcloud from original data, and find that some aspects that are taken into account include Food, Staff, Facility, Price and Location. This step helps in defining aspect list for labeling.

Figure 3: Wordcloud from the original dataset

- Besides, I printed the frequency of words and found out that high-frequent words containing important information so that just removing stopwords and no need to use TF-IDF removal.

```
.: 10966
,: 9768
ăn: 3591
and: 2488
the: 2473
ngon: 2341
is: 1745
food: 1606
không: 1575
món: 1517
vụ: 1463
có: 1345
a: 1344
to: 1342
!: 1319
và: 1211
good: 1182
phục: 1077
for: 1052
rất: 1047
Good: 1021
viên: 1016
I: 962
hàng: 933
The: 901
...
was: 887
là: 883
thì: 803
gian: 791
```

-

Figure 4: Top high-frequency words

- Another important point to consider is that the dataset contains both Vietnamese and English language data. Therefore, it is essential to separate them before preprocessing the data to ensure appropriate handling of each language.

2. Detect language

To separate Vietnamese and English reviews, I used GoogleTrans package to detect language of each review. The code with detailed explaination for this step is presented in the ***Preprocessing.ipynb file***.

```
en       4769
vi       4516
ar          4
lb          2
mr          2
ms          2
ja          1
la          1
mi          1
hi          1
jw          1
it          1
haw         1
es          1
nl          1
zh-CN       1
de          1
om          1
lv          1
Name: language, dtype: int64
```

Figure 5: Language and the number of reviews

After labeling review with its language, I separate the reviews into 2 files, one is Vietnamese and the other is English and ignore reviews in other languages because they account for a small percentage. This step is necessary to use the appropriate tools in preprocessing.

It is surprising that the number of reviews in English is higher than that in Vietnamese.

3. Cleaning data

The preprocessing steps are divided into 2 parts because lowercase and stopword removal steps will affect significantly the labeling process using VADER, which will be discussed in more detail in the next section.

This section involves cleaning the review text by removing special characters and icons, and converting all text to lowercase. Following this, the NLTK library is utilized to eliminate all stopwords, including commonly used words such as "the" and "a".

Following that, the text undergoes a lemmatization process to convert words to their base form. Subsequently, any abbreviated text is replaced with a dictionary of defined terms, which is sourced from Kaggle (life2short, n.d.) and updated to include abbreviations found in the current dataset.

4. EDA

4.1. Restaurant level

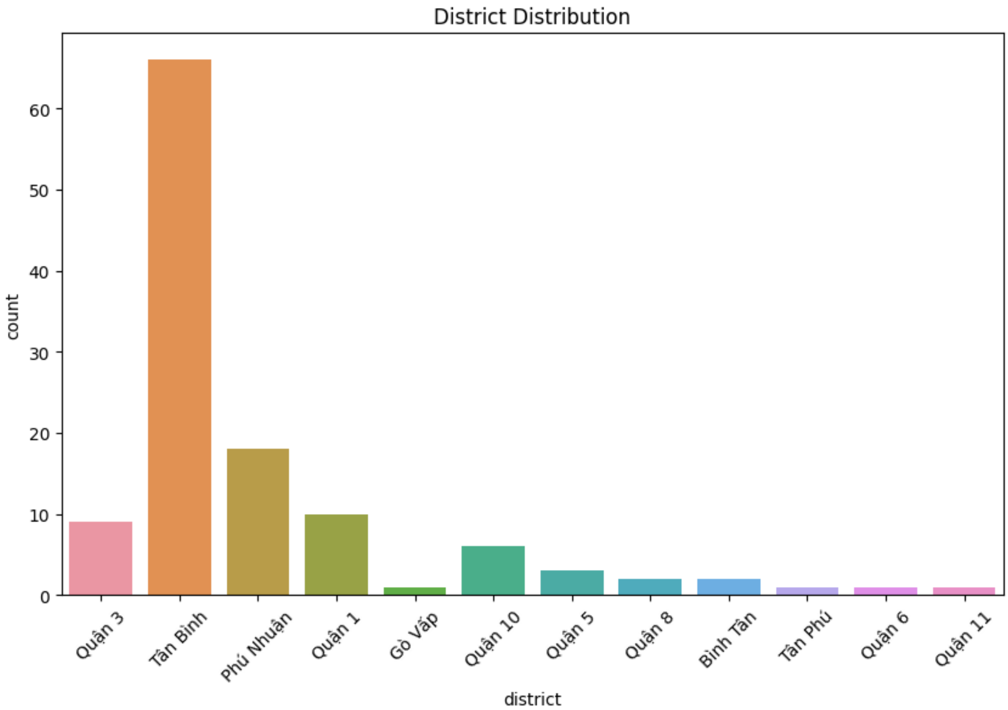

Figure 5: Distribution of district

A count plot was created to illustrate the distribution of restaurants across districts. It can be seen that most of restaurants in the dataset are located in Tan Binh district, and there is an imbalance among district. The reason may be of the way search district in scrape step. The extreme imbalance makes it irrational to compare among districts. However, this imbalance is unlikely to significantly affect the final result.

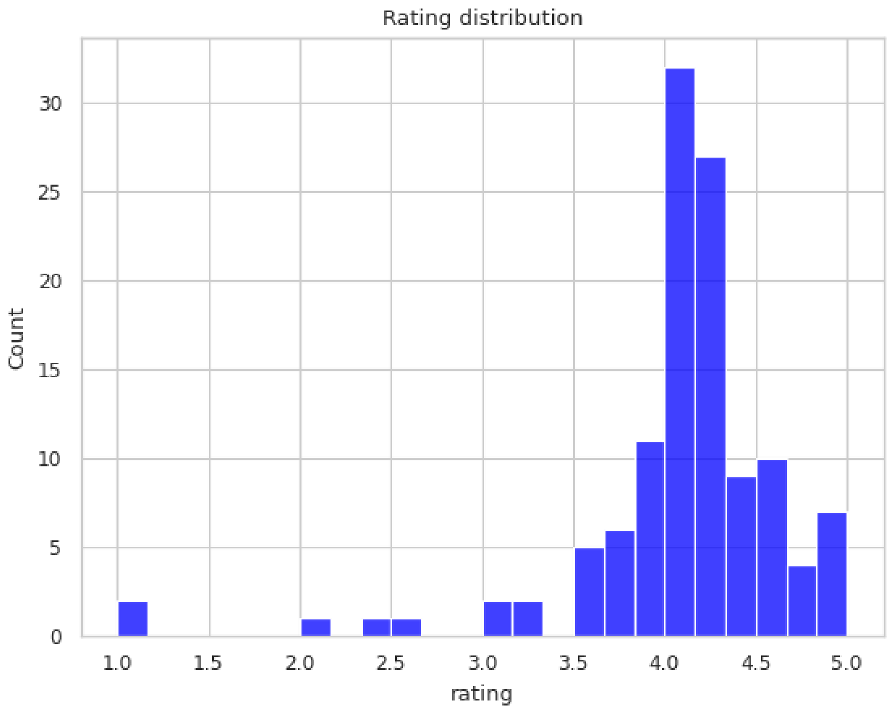

Figure 5: Distribution of average rating of each restaurant

The plot above provides information on the distribution of the average rating of restaurants. The majority of listed restaurants have high ratings, over 3.5. However, some restaurants have negative ratings, ranging from 1 to 2.5. It is important to consider the number of ratings for these restaurants.

4.2. Review level (Using the data prior to the removal of stopwords)

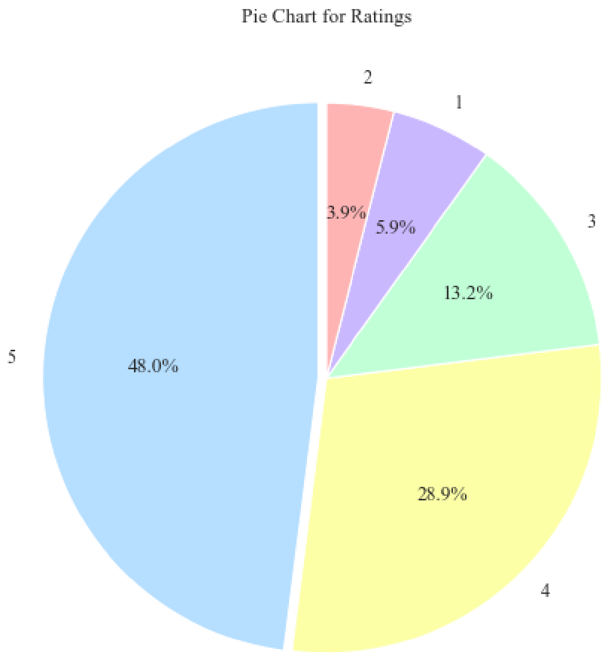

Figure 6: Percentage of rating

Reviews with 5 stars account for a huge proportion, with nearly hafl of the total. Besides, 4-star ratings also take up a lage percentage. Both 5-star and 4-star reviews are consider positive, which will cause imbalanced dataset. Therefore, it is necessary to conduct some techniques to resample such as oversampling to avoid bias phenomenon.

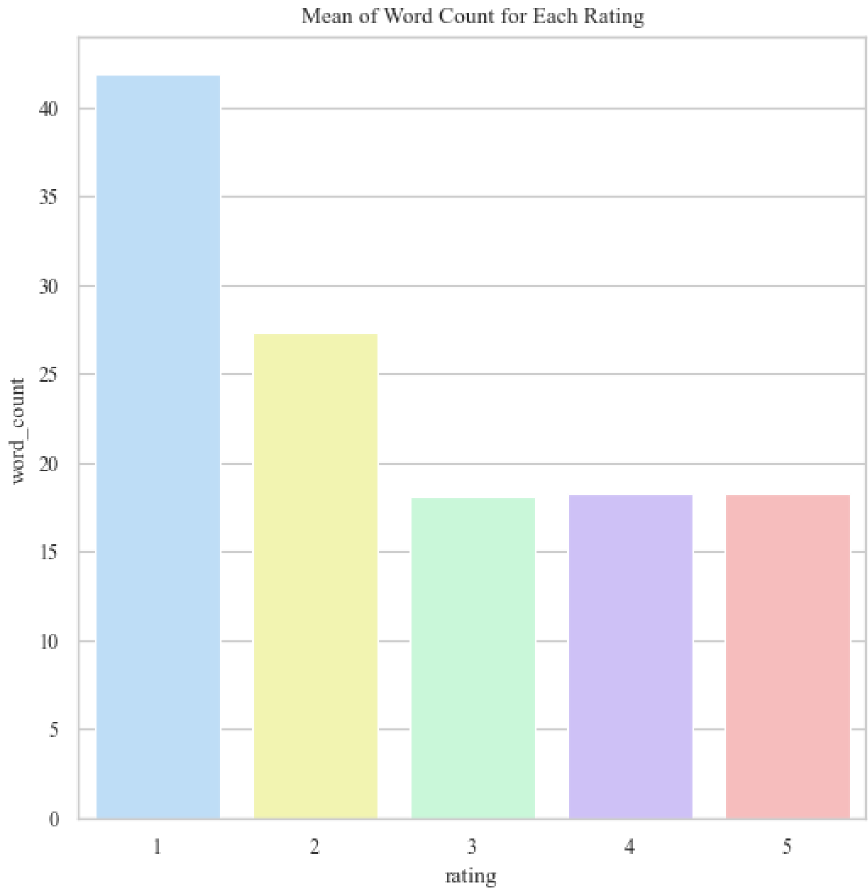

Figure 7: Average number of word per review

It seem that people tend to use more word in negative reviews than positive. It can be seen from Figure 5 that reviews with 1 star contain over 40 word on average.

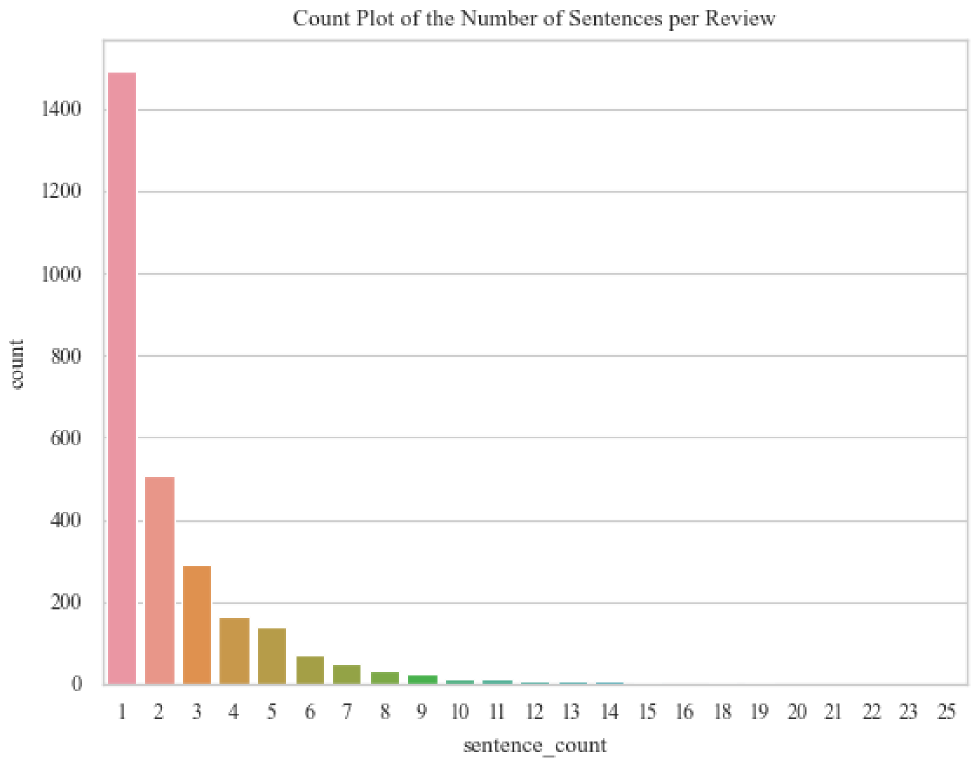

Figure 8: Distribution of average rating of each restaurant

Most of reviews have 1 sentence and the longest review has 25 sentences.

4.3. Sentence level (Using the data after the removal of stopwords

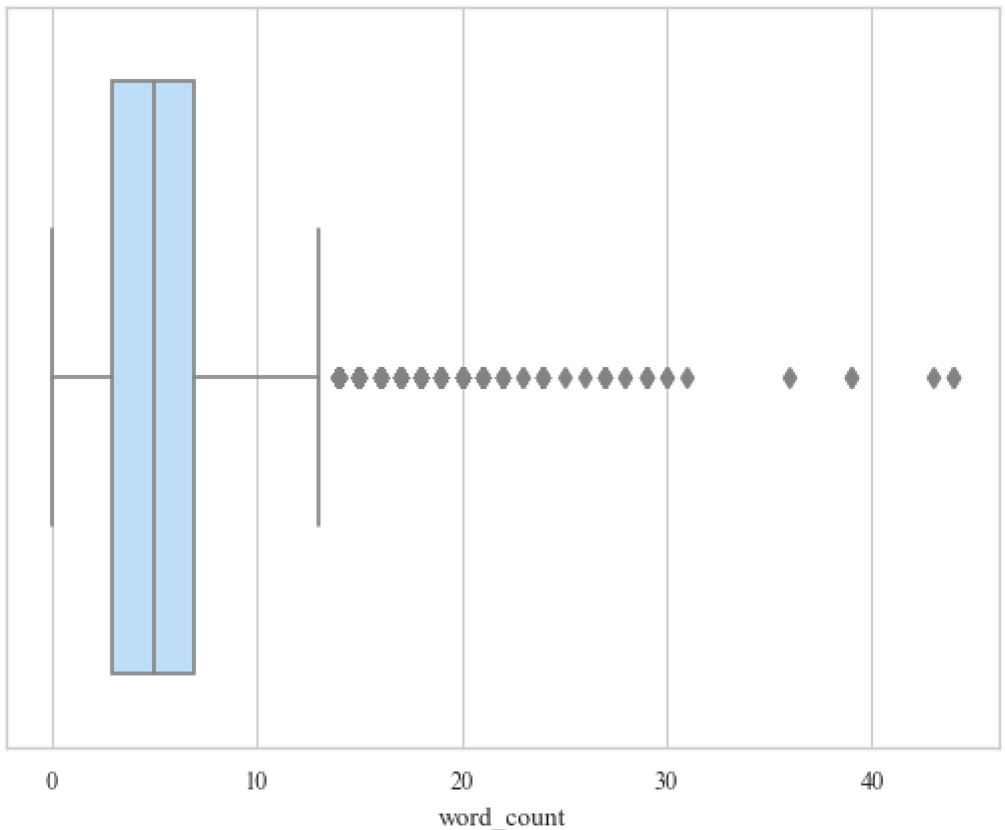

Figure 9: Distribution of average rating of each restaurant

Most of sentences, which have been cleaned and removed stopwords, contain less than 15 words. This information is useful for modeling process.

Figure 10: WordCloud

5. Labeling

Two commonly used sentiment analysis Python libraries, VADER and TextBlob, are used for labeling. These libraries use a lexicon-based method, which maps words to sentiment, and the overall sentiment of a sentence is determined by aggregating the sentiment of each word (Hutto, n.d.; Loria, 2018). These libraries are a faster alternative to manual labeling, which can be time-consuming.

Assigning a rating to each sentence based on the rating of the entire review is a quick labeling approach, but it may not be appropriate for reviews with a rating of 2-4, as they often contain both positive and negative sentiments. For example, consider a review from the dataset that praises the restaurant's drinks and service but criticizes the quality of the food and the restroom: "*This new, international-level restaurant near the city center is a great place to have some drinks (cocktails are really good). The atmosphere is nice, and the service is just impeccable. But the food does not match the other qualities of the venue. Some dishes are westernized to strange result (like fried rice with vegetables, which turned to be something once can expect in a cheap European cafe). Pomelo salad was nicely served and tasty, though. Also, note, that the venue is usually full, so reservation would be a wise idea. Ah, yeh - the WC does not meet the restaurant level at all.*". Although the review has an overall rating of 4, it would not be reasonable to rate the food and service as 4-star. Therefore, this approach is not chosen.

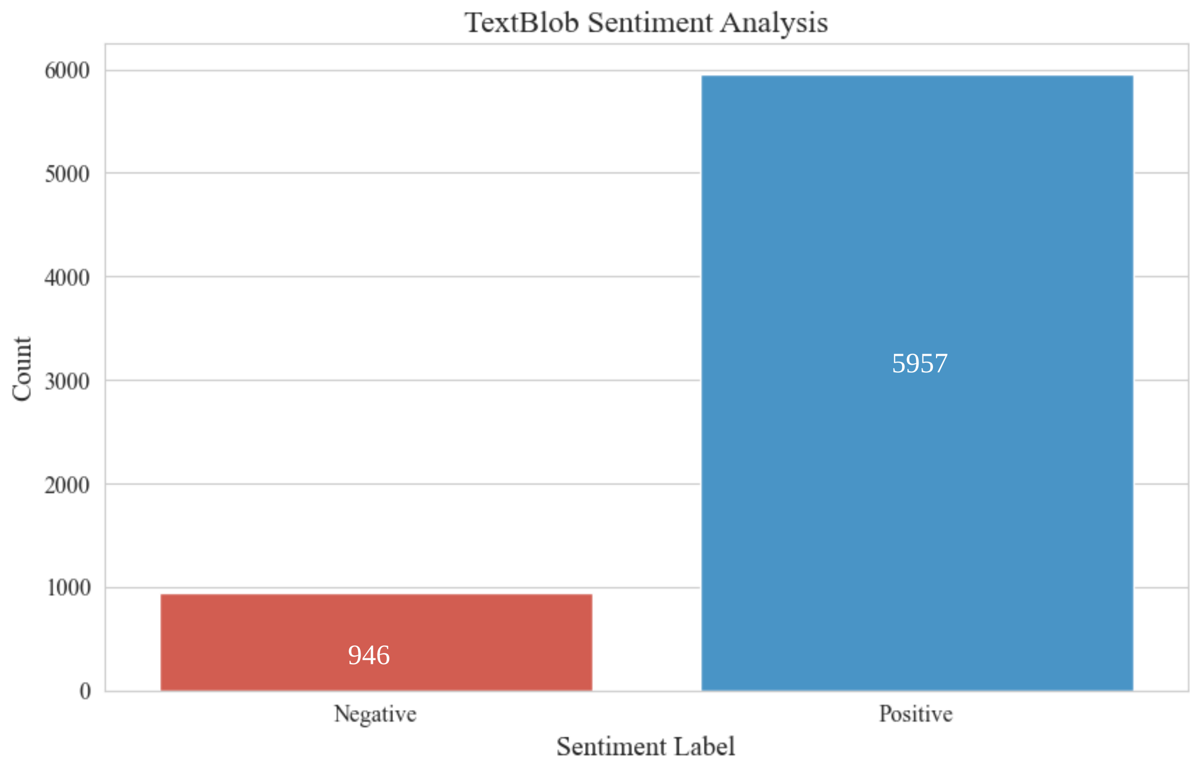

Figure 11: Labels by TextBolb

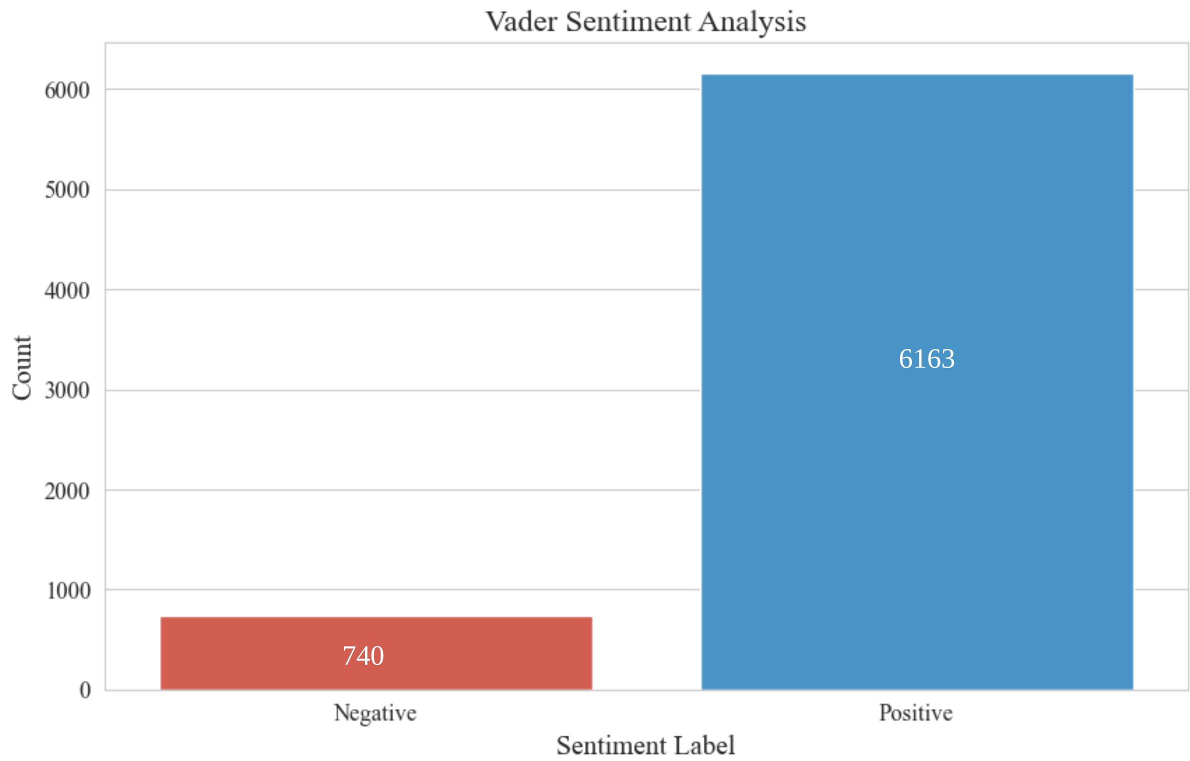

Figure 11: Labels by VADER

There is a huge imbalance between Negative and Positive labels. Thus, it is essential to oversampling data.

6. Modeling

6.1. LSTM

2 models for sentiment analysis using LSTM with the VADER and TextBlob lexicon are built by using Keras. Here's a description of each layer in the model:

- Input layer: This layer defines the input shape for the model.
- Embedding layer: This layer takes the input sequences and maps them to a high-dimensional space where similar words have similar representations. The size of the embedding space is defined by the EMBEDDING_DIM parameter, and the vocabulary size is defined by the MAX_NB_WORDS parameter.
- LSTM layer: This layer contains 100 LSTM units and has a dropout rate of 0.2. The return_sequences parameter is set to True, which means that the output of this layer will be a sequence of hidden states rather than a single hidden state.
- Dropout layer: This layer randomly drops 20% of the inputs to the previous layer, which helps prevent overfitting.
- LSTM layer: This layer contains 50 LSTM units and has a dropout rate of 0.2. The return_sequences parameter is set to True.
- Dropout layer: This layer randomly drops 20% of the inputs to the previous layer.
- Flatten layer: This layer flattens the output of the previous layer into a one-dimensional vector.
- Dense layer: This layer contains a single neuron with a softmax activation function, which produces a probability distribution over the two classes (positive and negative).

The compile method is used to configure the model for training, with the binary cross-entropy loss function and the Adam optimizer. The model is trained to optimize accuracy.

However, the predicted result from these models are unexpected because their poor performance in negative class.

Table 1: LSTM-VADER Classification Report

	Precision	**Recall**	**F1-Score**	**Support**
0	0.00	0.00	0.00	226
1	0.89	1.00	0.94	1845
Accuracy			0.89	2071
Macro Avg	0.45	0.50	0.47	2071
Weighted Avg	0.79	0.89	0.84	2071

Table 2: LSTM-TextBlob Classification Report

	Precision	**Recall**	**F1-Score**	**Support**
0	0.00	0.00	0.00	299
1	0.86	1.00	0.92	1772

Accuracy			0.86	2071
Macro Avg	0.43	0.50	0.46	2071
Weighted Avg	0.73	0.86	0.79	2071

Despite trying various resampling techniques, I was unable to improve the performance of the model for the negative class. This is likely due to the limited number of negative labels, with only 740 labeled as negative using VADER and 946 using TextBlob. As a result, I plan to collect more reviews and retrain the model to obtain better results.

6.2. SVM

The SVM model outperforms the LSTM model slightly due to its lower bias. For data labeled with VADER, the result is shown in the confusion matrix and classification report below.

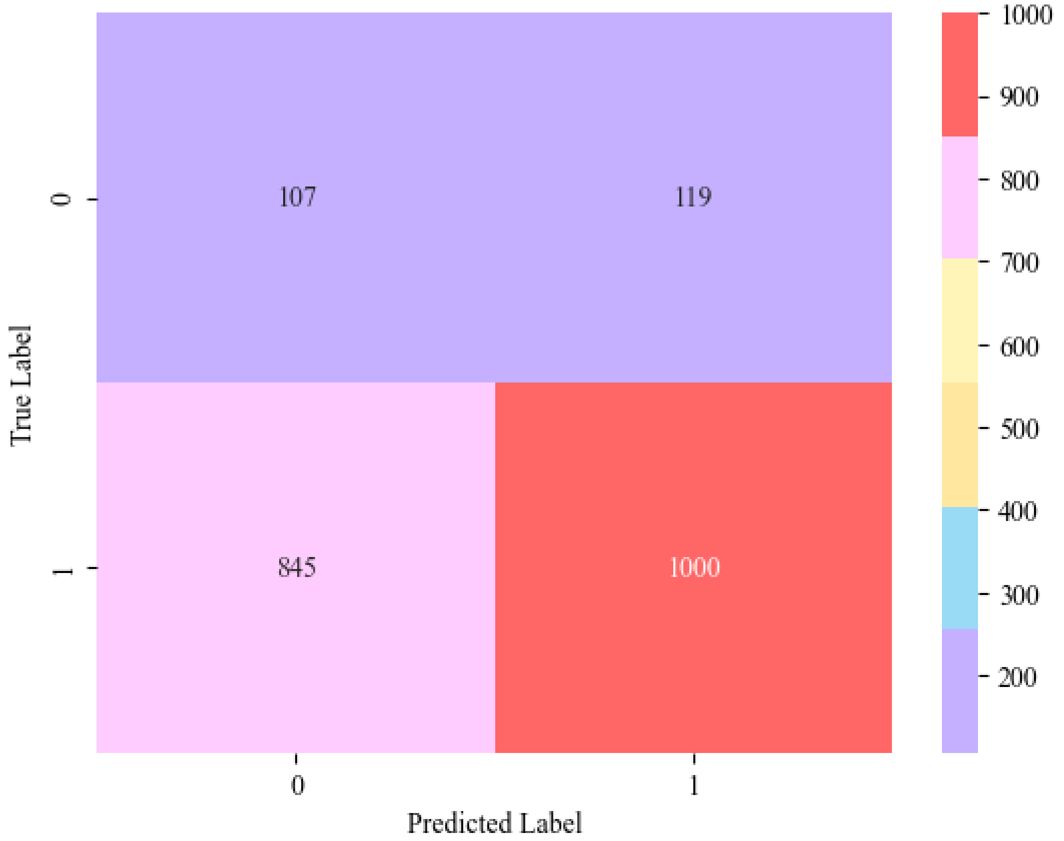

Figure 12: SVM-VADER Confusion matrix

Table 3: SVM-VADER Classification Report

Class	**Precision**	**Recall**	**F1-Score**	**Support**
0	0.11	0.47	0.18	226
1	0.89	0.54	0.67	1845
Accuracy			0.53	2071
Macro Avg	0.50	0.51	0.43	2071
Weighted Avg	0.81	0.53	0.62	2071

The TextBlob dataset give a better result than that labeled by VADER with the confusion matrix and classification report shown in the Figure 13 and Table 4.

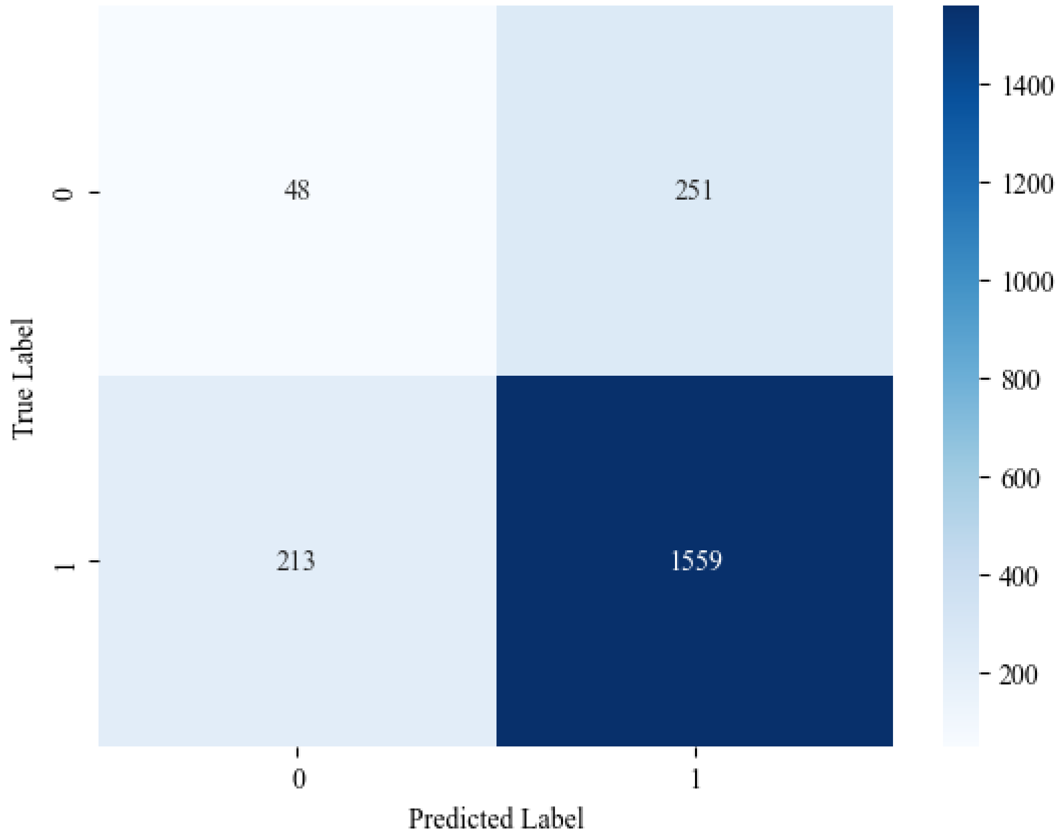

Figure 13: SVM-TextBlob Confusion matrix

Table 3: SVM-TextBlob Classification Report

	Precision	Recall	F1-Score	Support
0	0.18	0.16	0.17	299
1	0.86	0.88	0.87	1772
Accuracy			0.78	2071
Macro Avg	0.52	0.52	0.52	2071
Weighted Avg	0.76	0.78	0.77	2071

III. Conclusion

This project has been conducted through several processes, including scraping review from Google Maps, Preprocessing, EDA and finally Modeling. However, the result from models are not as good as expected. The reason is possibly of the small-size, imbalance dataset and the simple architecture model.

There are many remaining steps, which are marked in red in Figure 2, as well as dealing with Vietnamese dataset. To obtain better performance, it is essential to scrape more data. Regarding Vietnamese dataset, I will try some packages for preprocessing and labeling such as VnCoreNLP and VietSentiWordnet.

REFERENCE

Hutto, C. J. (n.d.). VADER Sentiment Analysis. GitHub. https://github.com/cjhutto/vaderSentiment

life2short. (n.d.). Data Processing: Replace Abbreviation of Word. Kaggle. Retrieved June 28, 2023, from https://www.kaggle.com/code/life2short/data-processing-replace-abbreviation-of-word

Loria, S. (2018). textblob Documentation. Release 0.15, 2.