Richa Meherwal- CS 410
Course Project Documentation

An overview of the function of the code

Code can be used to do an aspect based sentiment analysis. As seen in the code, we first tokenise all reviews. Then extract bigrams NN-ADJ pairs to form a word cloud and visualise the features that stand out the most. We also extract unigrams that are NN as the aspects to use for sentiment analysis.
Note that in order to train our classifier , we use the Ratings column from the Hotel dataset and then label our aspects with a pos,neu,neg sentiment. We then visualise the aspects and the associated sentiments using a bar plot. We the repeat this process over Airbnb reviews dataset and we use the sentiment classifier trained before to classify the sentiments of the aspects extracted. Note that Airbnb does not have ratings so it wouldn't be possible for us to retrain the classifier.

Software Implementation and Usage

1. Create a Python3 virtual environment
   python3 -m venv py3-env-final-proj

2. Activate virtual environment source-
   py3-env-final-proj/bin/activate

3. pip install -r requirements.txt

4. Install ipykernel in this environment-
   python3 -m ipykernel install --user —name=final-proj

   (final-proj will be used as env in jupter notebook)

5. Start Jupiter-notebook from shell using command :
   jupyter notebook

6. Download repository and open the *Test.ipynb* file

7. Change the file path to where the preprocessed files are i.e. under folder *data* and Run all the cells in the notebook

1. Read Pre-Processed Data

Change the file name here preprocessed_airbnb.csv/preprocessed_hotel.csv

```
df = pd.read_csv("preprocessed_airbnb.csv", encoding = "ISO-8859-1")
df[:1]
```

	Unnamed: 0	bigram_list	unigram_noun_list
0	86	clean_cozy ideal_place single_stay couple_single	mais place maintenance management reply answer visit room cozy comfy bed place stay couple traveler plan visit vancouver car return place

Note for testers

There is a joblib file that you can use to test the sentiment classifier. The classifier has been trained on HotelReviews dataset. Check Step 5.c under Hotel Review Analysis in Final_Project V4.ipynb or 4.b in Test.ipynb

Entire Code including the preprocessing and training sentiment classifier can be found in **- Final_Project V4.ipynb.**
You can also view this ipynb file using nbviewer - https://nbviewer.jupyter.org/github/richameher/CourseProject/blob/main/code/Final_proj%20V4.html
Modified Code for Testers can be found in- **Test.ipynb**

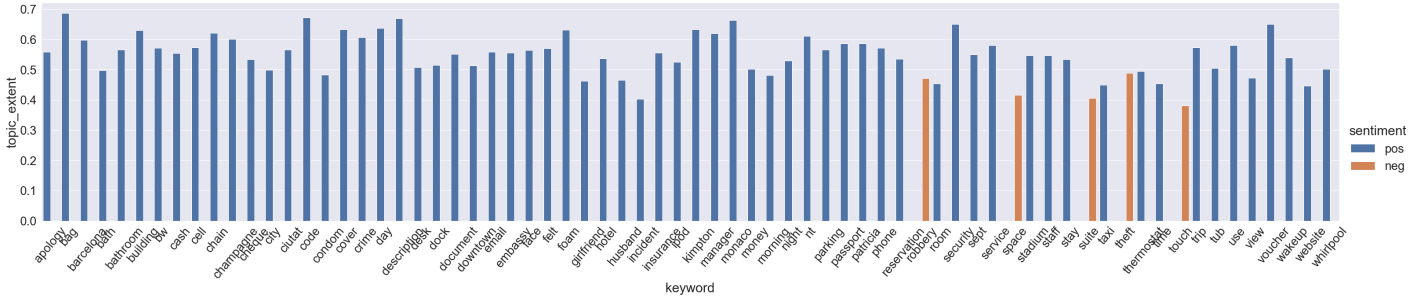
Final Results

Understanding Plots and Graphs

[Hotel Review WordCloud]

Use WordCloud to visualise bi-grams. NN-ADJ pairs are extracted from reviews and TF-IDF is used to retrieve top n bigrams.

There will also be a bar-plot associating the sentiment with every unigram NN keyword extracted with the probability of the sentiment.



[Hotel Review Aspect Sentiment Graph]

In the Wordcloud we can observe that people tend to talk about the quality of rooms. Features like safety is usually associated with the hotels than Airbnbs. Also Hotels have their own website , so people also talk about the online booking system. As for the bar plot, we can see that hotels have “theft”, “suite” aspects that have been associated with negative sentiment. Also the highest positive sentiment is observed among aspects like “room”, “view” and “manager”

What is completed and what could be better?

I have successfully been able to analyze the aspects that drives people to chose Airbnb over Hotels and vice-versa.

With Airbnb, like we can see in the graphs, the motivation is driven by finding an “affordable” yet comfortable stay. People care about “location” and “worth for money”. With hotels, people expect “luxury” in terms of “big rooms”, “spacious bathrooms”, “views” and so on.

However, the sentiment classifier could have been better if we had an available labeled dataset for airbnb reviews. I also realised that using LDA for bi-grams does not work well but specialised algorithms for Bi-gram topic extraction can be used in future. LDA for unigrams also did not group the categories very well , but top weighted words could have been considered. Therefore, I used TF-IDF to find the key aspects and only used nouns to do so.