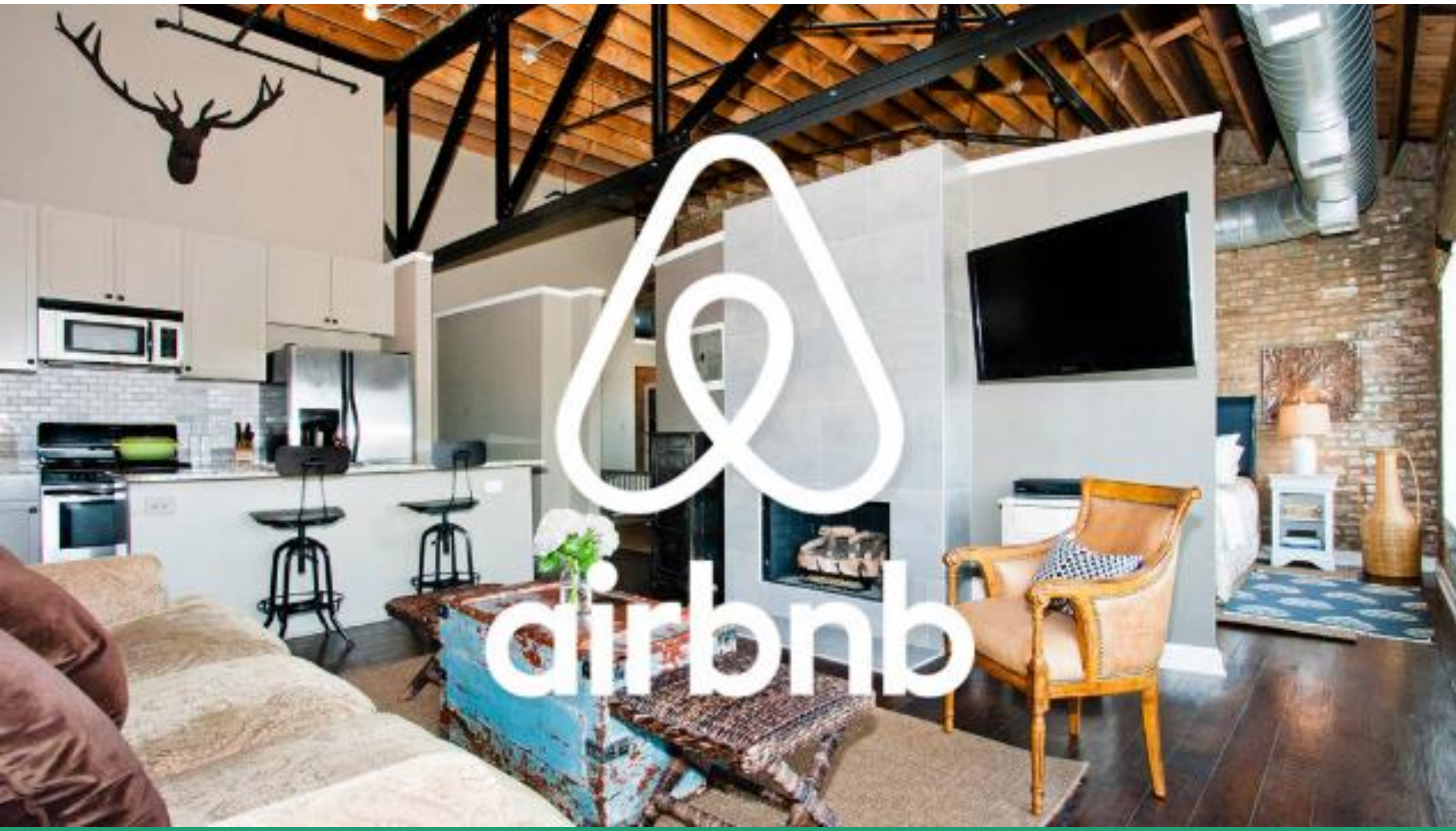
SMU
SINGAPORE MANAGEMENT UNIVERSITY



IS450- Text Mining and Language Processing

Final Report

Submitted by:

Ahmad Saifullah Bin Mustaffa, Gan Min Xuan Darren, Lu Yun Jing Ada, Muskaan Gupta, Shubham Periwal, Vikram Sanghi

Submitted to:

Professor Swapna Gottipati

A. Introduction

Airbnb is a popular platform for travellers to book their accommodation when they go overseas. Currently, Airbnb has the following aspects for reviews, along with the overall rating on a scale from 1 to 5 stars: Accuracy of listing, Communication with hosts, Cleanliness, Location, Check-in, Value. These metrics, while useful, are relatively general and are only shown on an overall scale. Customer review data (in text format) contains hidden information that cannot be shown on a ranking scale.

The aim of our project is to use language processing techniques to extract relevant and useful information from the unstructured textual data and to analyse it to aid decision making for potential travellers. This analysis, in turn, will also help house-owners make better managerial decisions. For the project, we will mainly be focusing on Los Angeles, USA as our primary location.

The data is sourced from the Inside Airbnb website: http://insideairbnb.com/get-the-data.html

B. Solution Overview

User #1: House-Owners

For the house-owners, the interface provided by Airbnb allows hosts to manage and prepare for current and future guests to their home, including feedback on how previous guests found their stay. However, this feedback given by Airbnb's own systems only rely on quantitative measures of the 1 to 5-star ratings left by reviewers, and do not account for the actual textual information that people take the effort to include in their reviews. In order to give hosts more detailed information about how their guests' opinion of their home, the dashboard will show **a clustering** model, where they can see the most talked about things for all LA listings, as well as for their own listings in order to easily see what topics are being frequently mentioned in their listing's reviews as well as across the rest of LA, to make changes in response to guests' feedback in order to improve the experience and potential business.

User #2: Potential Travellers

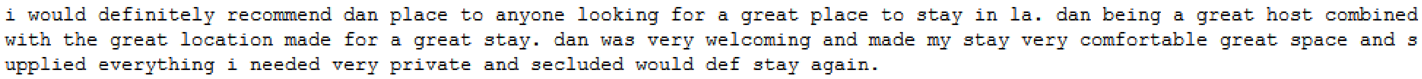
Currently, if a traveller wants to rent an Airbnb, they would spend a lot of time on reading through reviews as each listing tends to have more than 100 reviews per listing. This leads to inefficient usage of time, and in order to prevent this we have built a **review summarizer** and **sentiment analyser** to summarize key points being talked about a listing in all its review while the sentiment analyser facilitates comparison of listings and quick view of changes in sentiment towards a listing across time.

There are no customizable filters right now to filter based on personal preferences (for e.g.- Cleanliness, location, etc.) This prevents users from quickly finding what they want and leads them to pour through each review to know which listing clean or which listing has a good location. Thus, we have built a **chatbot** that lets travellers' key in their own personalized questions like - "Which are top 5 listings in BelAir, LA based-on cleanliness". This provides flexibility to travellers helping them make quick decisions.

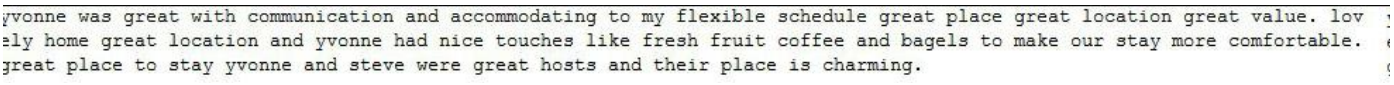
C. Solution Details

Feature #1: Review Summary of Individual Listings

1. **Purpose:** The most popular listings have hundreds of reviews written for them, with some even numbering over 500. A summary of all the English reviews will allow both house-owners and potential travellers to quickly gain an overview of what past guests at the house have to say about the listing.
2. **Methodology:**
   a. Group reviews according to their listing ID. The reviews are each treated as a document and the combination of all listings with their reviews is treated as one corpus.
   b. Filter out non-English reviews using *langdetect* library.
   c. Text Pre-processing consists of tokenization and stopword removal. Stemming was not done after comparing summary generation results with and without stemming. This is because in our generated summary, we needed words in the English Dictionary.
   d. Construct word frequencies from the remaining words in the corpus using Term Frequency (TF) within the corpus.
   e. Score individual sentences of the corpus based on their word frequencies. Thus, the sentences containing more popular words will receive a higher score as they are more likely to represent the summary of the entire corpus.
   f. Form a summary using the top *n* scored sentences using a priority queue. For the demo, we chose *n* to be 3 but the number can be tweaked accordingly to suit different cases.
3. **Results:** After testing the results of the langdetect library, an interesting observation was that the library had perfect precision in detecting English reviews should the review contain enough content (more than 10 words) as there were no False Positives found. False Negatives (English reviews labelled as other languages) were short, low quality reviews that were unlikely to contribute much to our TF weighting and therefore it was acceptable to remove these reviews in our filtering. After generating the summaries for every single Airbnb listing in our dataset, we observed that the majority of the summaries talked about the hospitality of the host, as seen in the sample summary output below. From this, we realise that most reviewers on LA Airbnb prefer a personal touch and to interact with their host as opposed to being left alone, with words such as “accommodating” and “comfortable”. Another observation would be that many reviews also talk about the location of the house, with a number of summaries also including phrases that mention about how “great”, “perfect” or “close” the location of the listing is.



```
i would definitely recommend dan place to anyone looking for a great place to stay in la. dan being a great host combined
with the great location made for a great stay. dan was very welcoming and made my stay very comfortable great space and s
upplied everything i needed very private and secluded would def stay again.
```



```
yvonne was great with communication and accommodating to my flexible schedule great place great location great value. lov
ely home great location and yvonne had nice touches like fresh fruit coffee and bagels to make our stay more comfortable.
great place to stay yvonne and steve were great hosts and their place is charming.
```

Feature #2: Sentiment Analysis of Individual Listings

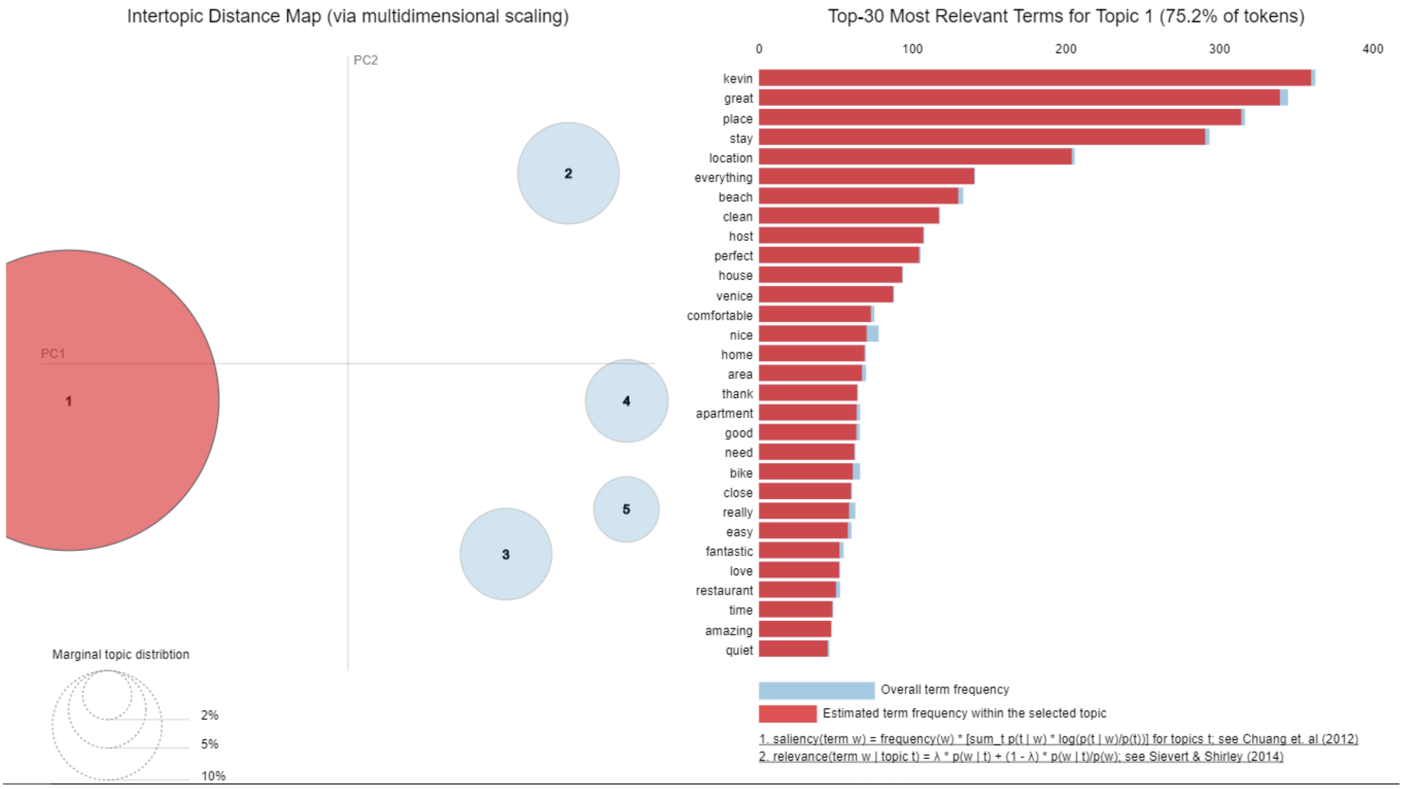
1. **Purpose** Following from Feature #1, the reviews of a listing are read to understand if they are positive or negative compared to other listings. Furthermore, when there are hundreds of reviews it gets time consuming if done manually like now. Thus, this feature allows users to see sentiment score of each review over time for a listing and facilitates comparison and faster decision making. In turn, this feature also allows a house owner to see whether his ratings are improving or worsening over time and might get him to make some changes in his listing or behaviour.
2. **Methodology:**
   a. Group reviews according to their listing ID. Each review is treated as an individual document.
   b. The reviews are cleaned to remove stop words and stemmed. Since we do not need the actual word, just the sentiment behind it, we can perform stemming.
   c. Creating a manually labelled training dataset: used nltk classifier to label 1000 reviews and then manually vetted through them to change the sentiment where we deemed necessary
   d. Training the classifier based on training dataset.
   e. Use the classifier to predict the sentiment for each review that was passed.
3. **Results:** From an initial screening of the review sentiments we found that majority of the reviews tend to be positive (93%). Diving further into it, there were three reasons for this:
   a. Only the top listings in LA were filtered initially to shrink the data which might have left popular listings and secondly
   b. The positive reviews of a listing tend to be towards host so majority of them had good comments about the host while negative ones criticized the facilities in the listing (For eg- Air conditioner, swimming pool, bed)
   c. Since the training dataset we have had an imbalanced number of reviews, where most were positive, we believe that the classifier has a higher chance of predicting a positive sentiment

Feature #3: Cluster Analysis / Topic Analysis

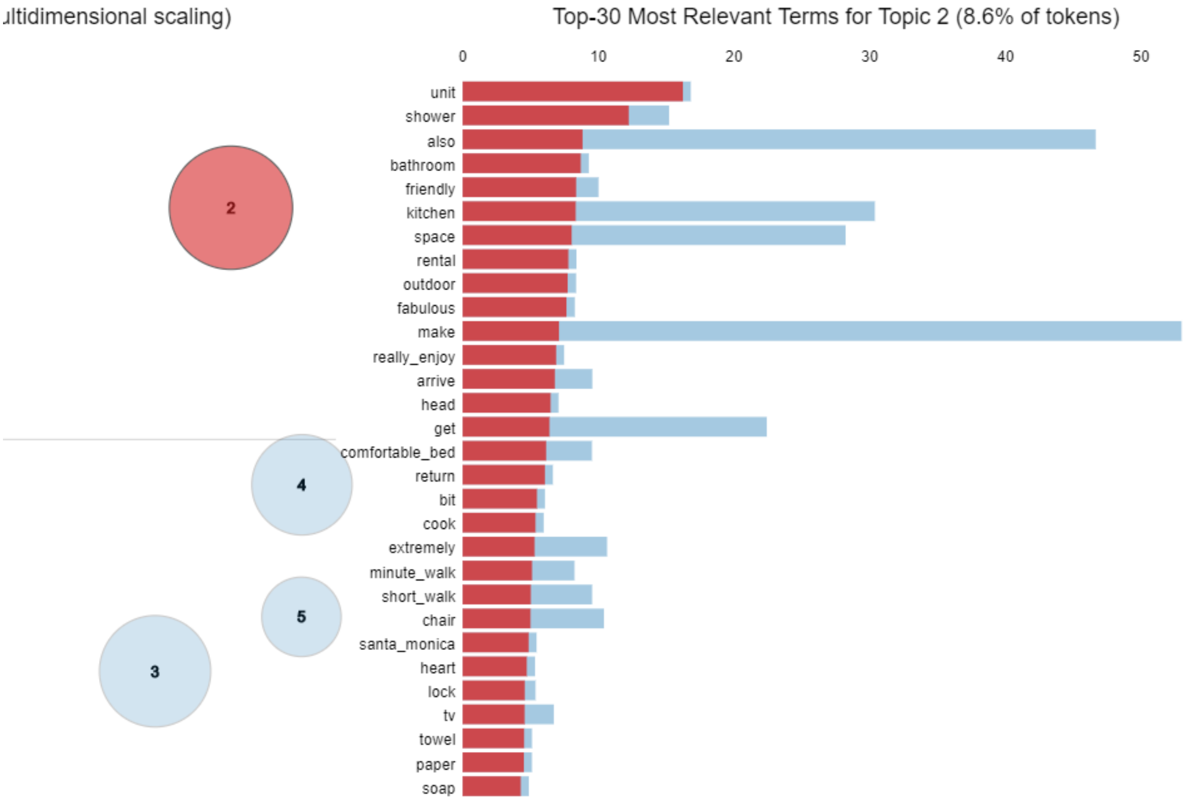
1. **Purpose:** Analyse what features travellers look for in a hotel which is indicated by what they comment about. This will help house-owners in understanding what travellers are looking for and thus, make better decisions.
2. **Methodology:** The idea is to cluster reviews based on certain traits i.e. area/location, food, etc. We do that using the following techniques:
   a. Accept the listing ID from the user
   b. Extract all reviews from this particular listing from the dataset
   c. Pre-process the data before clustering- includes removing stopwords, only keeping alphanumeric words, and lemmatizing the words. Words were lemmatized instead of stemmed as we wanted to retain the word form and meaning. Stemming the word might reduce the word to something meaningless i.e. stemming ‘hospitality’ would give us ‘hospit’ while lemmatising the same word returns ‘hospitality’, retaining its meaning in the word.
   d. Create a dictionary and corpus after identifying unigrams, bigrams and trigrams with Gensim’s phraser model.
   e. Conduct cluster analysis using LDA (and tune the parameter to 5 clusters. An increase in the number of clusters often see an overlap of topics.)
   f. The results were visualised on a pyLDAvis graph.

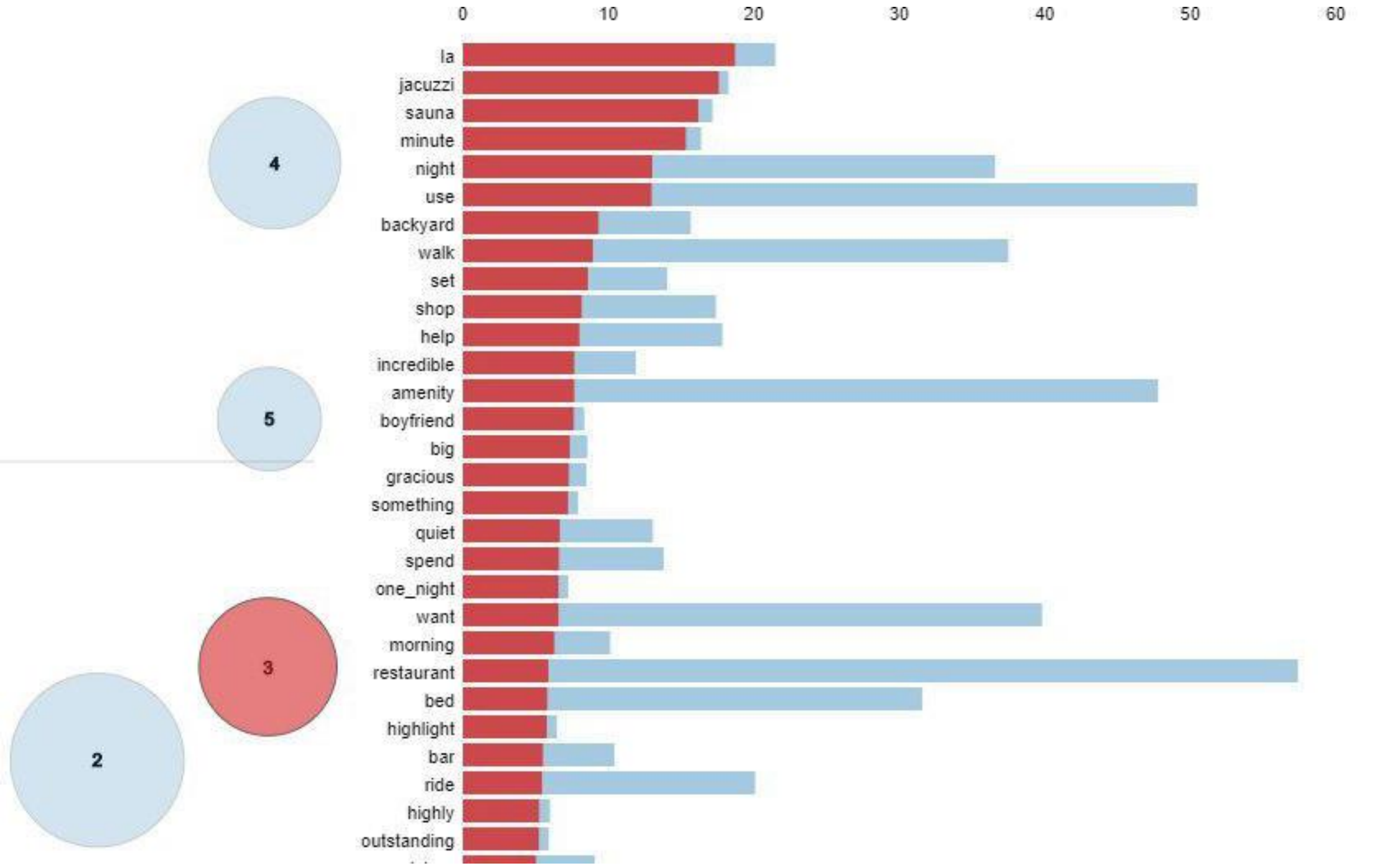
3. Results:

a. For example, for listing #893786, we can see that cluster 1 talks about the location of the house, cluster 2 talks about the house amenities; for listing 6438015, cluster 3 talks about amenities for that listing as well.



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)



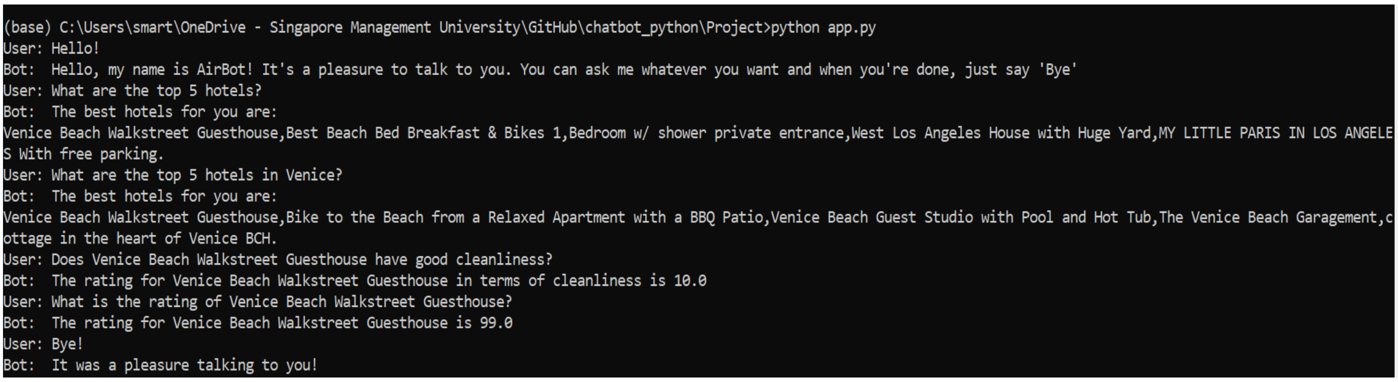


Feature #4: Chatbot

1. **Purpose:** The chatbot is catered towards making the lives of travellers easier. Presently, AirBNB does not provide an option for filters to search hotels based on features such as "cleanliness". We want to allow the users to type in their questions and get a customised response. For example, a question can be as detailed as "What are the top 5 hotels in LA below $100 based on cleanliness". This question would take the user a while to figure out if he were to use the AirBNB website directly
2. **Methodology:**
   a. Accept the query from the user
   b. Pre-process the query by removing all non-alphanumeric characters except '$' (as it will be used in detecting price), and create a list out of the lemmatized words
   c. Match intent- use regex to detect if user has entered a greeting, question, goodbye, or thanks.
      - If not question, return with a standard response
      - If question, detect question pattern using regex. Some question patterns we have are:
        1. *"what (is|are) the top (.*) hotel (.*)"*. Allows users to add filters and search for hotels
        2. *"what (is|are) the rating of (.*)",* Allows users to search for specific hotels either based on different features or overall

   d. Extract entities in the question
      - Extract prices/neighbourhood/numbers. Use Stanford NER parser to detect whether user has entered a price (MONEY), a specific neighbourhood (GPE), or the number of hotels he wants (CARDINAL)
      - Send to pre-defined functions. These functions will then query the results from the dataset and create a list of recommended hotels
   e. Generate Answer
      - Add the list of recommended hotels to a pre-defined response template so that the user is happier with the interaction
      - Respond- Send message back to user

3. **Results:** Unlike the other features, the results of the chatbot (like what questions do users ask mostly) will be generated once actual users start using them. However, following is a screenshot of what a typical user-chatbot interaction will look like.



```
(base) C:\Users\smart\OneDrive - Singapore Management University\GitHub\chatbot_python\Project>python app.py
User: Hello!
Bot:  Hello, my name is AirBot! It's a pleasure to talk to you. You can ask me whatever you want and when you're done, just say 'Bye'
User: What are the top 5 hotels?
Bot:  The best hotels for you are:
Venice Beach Walkstreet Guesthouse,Best Beach Bed Breakfast & Bikes 1,Bedroom w/ shower private entrance,West Los Angeles House with Huge Yard,MY LITTLE PARIS IN LOS ANGELE
S With free parking.
User: What are the top 5 hotels in Venice?
Bot:  The best hotels for you are:
Venice Beach Walkstreet Guesthouse,Bike to the Beach from a Relaxed Apartment with a BBQ Patio,Venice Beach Guest Studio with Pool and Hot Tub,The Venice Beach Garagement,c
ottage in the heart of Venice BCH.
User: Does Venice Beach Walkstreet Guesthouse have good cleanliness?
Bot:  The rating for Venice Beach Walkstreet Guesthouse in terms of cleanliness is 10.0
User: What is the rating of Venice Beach Walkstreet Guesthouse?
Bot:  The rating for Venice Beach Walkstreet Guesthouse is 99.0
User: Bye!
Bot:  It was a pleasure talking to you!
```

Feature #5: Dashboard

1. **Purpose:** A one-stop platform for travellers and Airbnb hosts. The dashboard allows users and hosts to view sentiments and customer feedback specific to a particular listing. For every listing, the dashboard displays
   a. **Word cloud-** containing the most frequent words used by travellers in their feedback. This information is useful to the hosts of the listings because it identifies the most talked-about aspect of the listing e.g. cleanliness, bedding. This allows hosts to identify potential areas of concern, therefore making it easier for the host to make specific improvements to their services.
   b. **Pie chart-** displays the overall rating of the listing: Positive, Negative or Neutral. For travellers, it informs them whether that particular listing is generally liked by previous guests. Naturally high Positive ratings would mean that the listing meets the general expectations of travellers.
   c. **Time-series line graph-** displays the variation in the overall ratings of a particular listing over a five-year span. From this diagram, users are informed of how consistent the accommodation is in meeting the expectations of travellers. Erratic line graphs would indicate fluctuating service standards, whereas more consistent-patterned lines indicate a certain degree of consistency. This is useful information for the hosts to upkeep their standards and not be complacent.
   d. **Intertopic distance map-** depicts the most talked-about aspects of the accommodation

2. Methodology

a. Word cloud

- Summary analysis (Feature #1) results for each listing compiled in CSV format. Each row contains a summarised review of the listing. There are multiple rows corresponding to one listing.
- All rows are combined and input into (https://www.jasondavies.com/wordcloud/#%2F%2Fwww.jasondavies.com%2Fwordtree%2Fcat-in-the-hat.txt)
- SVG object is obtained

b. Pie chart (Framework: D3.js)

- The team reused code from (http://bl.ocks.org/diethardsteiner/3287802)
- Input used was the CSV file obtained from sentiment analysis (Feature #2) containing the ratings (scale: 0.0 – 1.0) corresponding to each listing

c. Time-series line graph (Framework: D3.js)

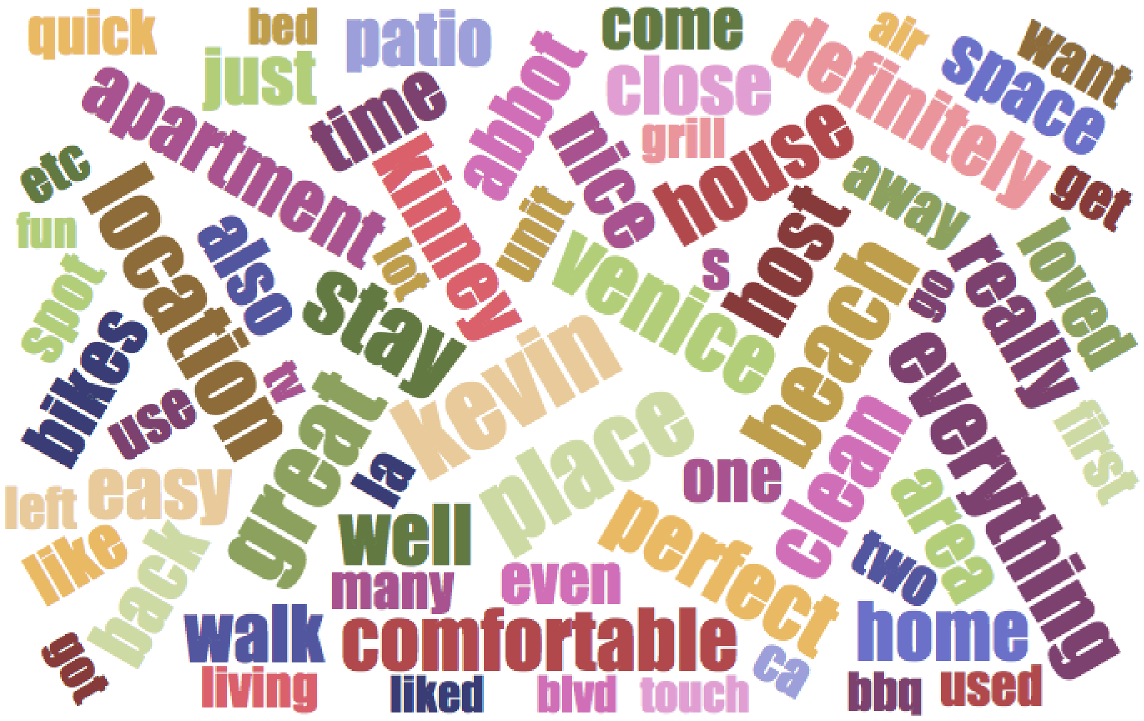
- The team reused code from (http://bl.ocks.org/d3netxer/10a28b7aee406f4e7fce)
- Input was the same file used for the Pie Chart

d. Inter-topic distance map

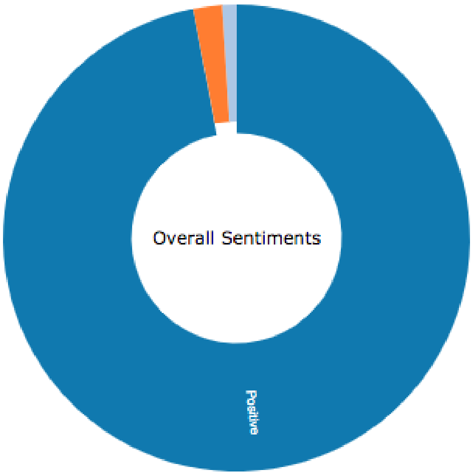
- The team generated the visualisation through the Jupyter Notebook using pyLDAvis
- The visualisation was exported as a HTML file into our webpage

3. Results

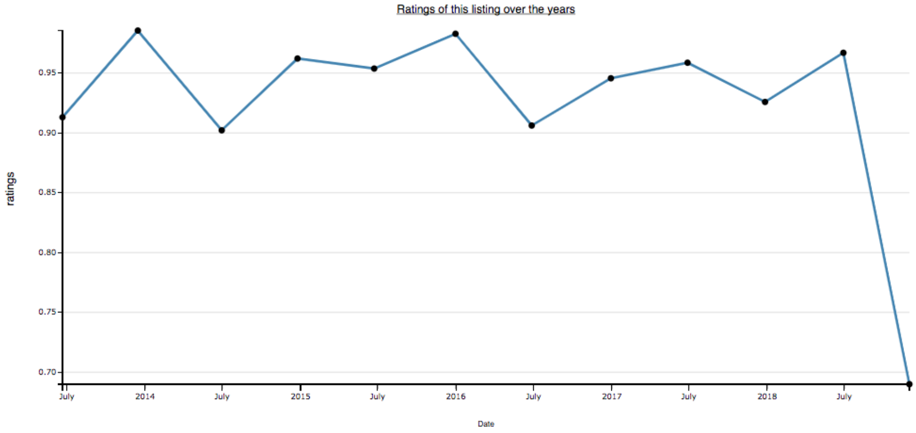
a. **Word cloud:** For this listing, the most common, frequently talked-about words are “location”, “great”, “close”, “beach”. This shows the areas of concern.



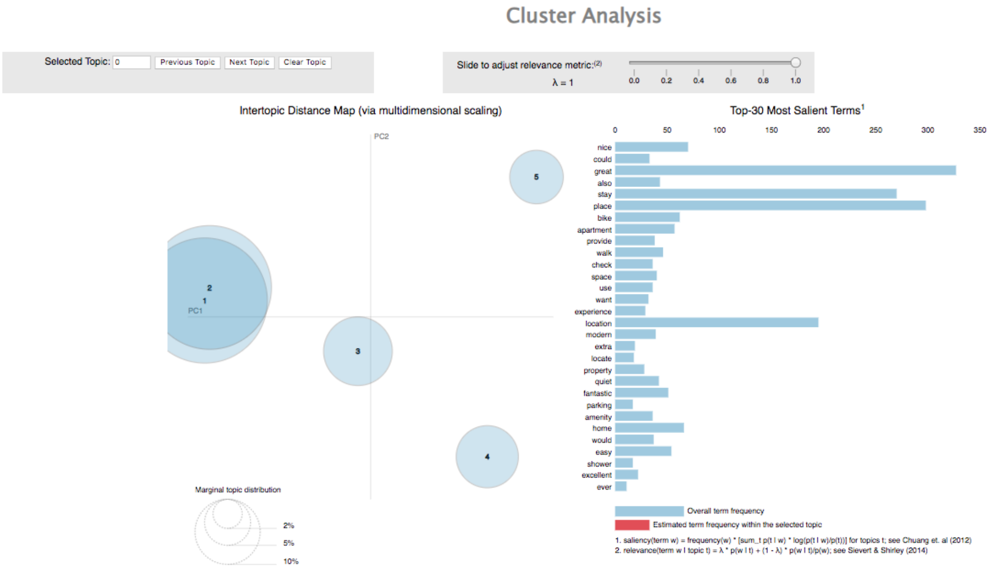
b. **Pie chart:** This indicates that the listing receives 93% positive ratings from all its guests, 4% negative ratings and 3% neutral ratings.



c. **Time-series line graph:** This indicates that the ratings for this listing fluctuates from below above 0.95 during 2014 to below 0.70 in 2018. The host can identify an erratic trend, which might be a potential cause for concern as the service might be inconsistent.



d. **Intertopic distance map:** From this diagram we identify the topics that are most talked-about.



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

D.Challenges and Mitigations

1. **Data Cleanup**

   The raw dataset obtained from Inside Airbnb was extremely large (over 1 million reviews) and contained non-English reviews as well as automated cancellation messages as well. There was a need to be able to filter out these non-helpful reviews and this was accomplished using Langdetect library to filter out non-English reviews and the manual filtering of reviews containing the string "This is an automated posting" in order to obtain a cleaned dataset that contains meaningful reviews.

2. **Review Summary**

   Initially, the review summaries were generated using the TF*IDF algorithm for term weighting. This resulted in poor results as the TF*IDF algorithm penalised terms that appeared commonly across multiple reviews as well as long reviews, which is the exact opposite of what we wanted to achieve. This issue was mitigated by switching to TF to weigh the terms over TF*IDF.

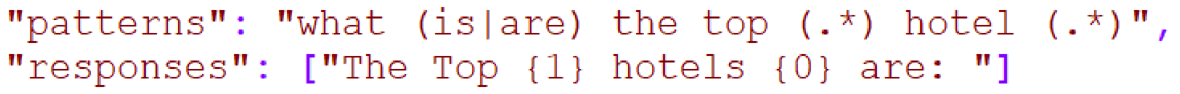
3. **Topic Analysis**

   Before identifying bigrams and trigrams, we realised many phrases such as 'Los Angeles' or 'Venice Beach' was split up into 'los', 'angeles', 'venice' and 'beach', thus giving us inaccurate results and the meaning of the phrase were lost. In order to fix this, we used Gensim's Phraser model to identify bigrams and trigrams by putting together words that are commonly used side by side in the data. Examples of the the result we got were 'would_highly_recommend', 'within_walking_distance' and 'mini_fridge'. These results were more helpful in understanding what were users talking about.

4. **Chatbot:**

   The users can ask questions in any format. Same question can also have multiple answers. Moreover, since the questions are very small and similar ("What are the top N hotels?", "What are the top N hotels below $100?", "What are the top N hotels in Long Beach?" etc), we could not use term frequency to detect question pattern.

   Solution: Therefore, we used regex to detect the patterns. We used regex objects like:



```
"patterns": "what (is|are) the top (.*) hotel (.*)",
"responses": ["The Top {1} hotels {0} are: "]
```

   Thus, for a question of the pattern "What are the top 5 hotels in Long Beach", we will respond with "The top 5 hotels in Long Beach are: " and fill in the blanks with the answer generated.

5. **Dashboard:**

   We initially intended to utilise the Plotly dashboard framework but due to time constraints, we were not able to implement in time and had to resort to drafting up a web page from scratch, using off-the-shelf visualisations as well as D3.js visualisations. This compromised on the data storage aspect, forcing some of our input data to be hard-coded.

E. Future Work and Conclusion

1. **Review Summary:** The TF method of weighing the sentences to select for the summary tends to result in repetition of words and points. While this is what we want for our analysis, the user may find it repetitive to read a summary that contains the same keywords. An area of improvement would be to use abstractive summarisation instead of the current extractive method in order to condense and paraphrase points made by reviewers.

2. **Review Sentiments:** Expand the model to fit other locations: currently the training dataset only consists of reviews from LA while if we move to other locations like London we will have to customize training data with listing reviews from London because people in different places tend to write about different things.

3. **Cluster Analysis:** The drawback of this model is that the clusters right now do not have labels assigned to them. We tried to create our own dictionary to train the model to assign labels to the clusters, but the results were very inaccurate. So, the final model classifies customer reviews into non-labelled clusters and then we can manually interpret what each cluster talks about. If we had more time and access to a training dataset for assigning cluster labels, it would help make our clusters more meaningful and faster to interpret.

4. **Chatbot:** We intend to make our chatbot smarter by training it such that it can detect question patterns even if they're not exactly the same as pre-defined. We would also like to add some tastefulness in the bot such that it can remember what has been asked before and detect negation. For example, if the user has not asked for any filters, then bot can prompt them for it and remember the list of filters they enter in the form of a conversation. Lastly, the bot cannot handle multiple questions together as of now, so we would like to train our model for that purpose as well. Lastly, we would also like to make it usable by the house-owner by adding some question patterns like "What do the users talk about regarding my listing?", "Do the users prefer cleanliness or location", etc. We believe this will help him get a better understanding of the users and thus, give them a better experience

5. **Dashboard:** Utilise a dedicated database system so as to automate the input of data into our diagrams.

F. Project Experiences and Reflections

Team Member	Contribution	Learning Point
Ahmad Saifullah Bin Mustaffa	Dashboard	*“This project has helped me to explore wider possibilities in constructing dashboards. The time constraint and unfamiliarity with the Python language hindered the use of Python-based dashboards, but this has taught me the importance of widening my skillsets.”*
Gan Min Xuan Darren	Summarizer	*“This project has allowed me to experience the entire process of conducting data analytics. From choosing an appropriate data source, data clean-up, pre-processing and actual analysis, it’s been a great learning experience that will help me for the future. Also, don’t test your code for data cleanup on a dataset of 1 million review”*
Lu Yun Jing Ada	Topic Analysis	*“This is my first analytics project and I learnt a lot through it from choosing the data to understanding it, deciding the scope of the analysis, cleaning up the data and finding various methods to analyse it. There is more than one solution to any problem and we choose the more optiomal solution through our own knowledge, research or even trial and error.”*
Muskaan Gupta	Topic Analysis	*“I never worked with visualisations and interactive dashboards on python before, so that was new and interesting.”*
Shubham Periwal	ChatBot	*“Regex can be very useful if one knows how to use it.”*
Vikram Sanghi	Sentiment Analysis	*“A successful data analyst is better at statistics than programmers and better at programming than statisticians”*

G. Appendix

References:

1) https://machinelearningmastery.com/multi-class-classification-tutorial-keras-deep-learning-library/
2) https://towardsdatascience.com/journey-to-the-center-of-multi-label-classification-384c40229bff
3) https://stackabuse.com/hierarchical-clustering-with-python-and-scikit-learn/
4) https://towardsdatascience.com/airbnb-rental-listings-dataset-mining-f972ed08ddec
5) https://stackabuse.com/text-summarization-with-nltk-in-python/