Hotel Like Home

Report

Contents

Introduction

In today's world, social media plays a key role in influencing the people's actions and thereby transforming business. The key objective of this project is to process the unstructured hotel reviews written online and condense the overall insights into a readable format for both customers who are looking for a perfect hotel and the hotel management who wants to improve the customer experience in their hotel. Through the usage of Natural Language Pre-Processing, Text mining and sentiment analysis techniques, we thrive to provide a better platform for reviewing the reviews for both stake holders.

Overview

Customer reviews content on the social media are important source to consumers and the business unit. Study shows that 94% of people say that a negative online review has convinced them to avoid a business [1]. Research shows that 91% of people regularly or occasionally read online reviews, and 84% trust online reviews as much as a personal recommendation and 68% form an opinion after reading between one and six online reviews. But the important note here is, any business will have at least one negative review, how do we ensure that the consumers do not choose a bad buy over good buy based on these biased reviews. In addition to this perspective, these online reviews are used by the business management to provide an efficient way to understand the customer experience and their areas of improvement.

There is a prevalent adoption of social media in the hotel industry. As the hotels operate in a competitive and dynamic environment, it is important for the management to stay updated on the customer experience to achieve success in their business. (Berezina et al., 2016). Ye et al. (2011) indicate that a large percentage of customers rely on the online user-generated reviews to make online purchase decisions for hotels, higher than any other product category [2]. So the hotel managers should act on their areas of improvement and respond to the customer reviews to avoid shooing away potential consumers. On the same note, the consumers have to ensure that they are not carried away by few good reviews written online, which they might regret later. In our project, we have focused on developing an interactive application which not only condenses the hundreds of reviews, but also allows comparison of a selected hotel's performance across other hotels in the market. In the rest of our report, we will explain the text mining process employed and interpret the results of our analysis.

Data Source

We obtained a list of ~33,000 reviews for 1000 hotels across The United States from Datafiniti[3]. The dataset includes the following details:

Metadata:

Field Name	Description
Hotel Location	Latitude and Longitude information
Hotel Name	Name of the hotel
Address	Address of the hotel
Review	Review given by the customer
Rating	Rating given by the customer

Methodology & Results

The methodology used in our project can be summarized as the following five aspects.

1. Corpus creation
2. Data pre-processing
3. Topic modelling
4. Sentiment analysis
5. Output analysis

Corpus creation

The raw data we get from Datafiniti's business DB is in a CSV format, we read CSV data into python file to conduct subsequent analysis. Then, we sort the imported data frame by HOTEL NAME, corpus creation logic has been done on sentence level where each sentence is itself a list of words. Then, run a FOR loop on all rows of the dataset to create one txt file for each row with the data from the REVIEW column. All the text files will be generated where ipynb file is placed, we create a folder named Hotel Corpus and stored all text file in that.

The FOR loop is as below:

```
for i in range(0,len(hotel_sents_df)-1):
    if  hold_hotel != hotel_sents_df['name'][i]:
        j=1
        hold_hotel = hotel_sents_df['name'][i]
    f = open(str(hotel_sents_df['name'][i])+" - "+str(j)+'.txt', 'w+')
    f.write(str(hotel_sents_df['review_sent'][i]))
    f.close()
    #print(i)
    i+=1
    j+=1
```

Pre-processing

The preparation is an iterative process with data review steps after each iteration to validate the necessity to tweak the stop words removal step with adding/removing domain specific stop words to the default stop word library for the English language. The initial steps of pre-processing text data are as follows:

- Tokenization

When we perform corpus creation, tokenization is done while loading the files, that is, the original text is split into individual words and stored as a list of words in Python.

- Case conversion to lower

We want to change everything to lowercase, in English, sentences start with capitalized words. However, for many text analysis tasks, we should not differentiate between a capitalized word and its original form, but it is easier to use lowercase in later analysis.

- Non-alphabetic characters removal

We got a challenge in our dataset, that is special characters in hotel names, which may cause folder creation problem. Thus, we remove non-alphabetic characters, it will give us a further clean text.

- Stop-words removal

NLTK also has a built-in stop word list for English that can come in handy when we need to remove stop words from a text collection. Check words in document, if they are in stop list then remove them.

- Non-English words removal

Another challenge in our dataset is we have non-English reviews, which means some reviews are written in Spanish, Chinese and so on. This will affect our analysis results, so we should remove these non-English words.

- Stemming

The last step in pre-process is stemming. NLTK also has a built-in Porter stemmer we can use. The Porter stemmer is used to obtain only the unique stems of words.

```
stop_list = nltk.corpus.stopwords.words('english')
new_stop_words = ['hotel','room','negative','good','great','love','recommend','grove']
for i in range(0,len(new_stop_words)):
    stop_list.append(new_stop_words[i])
stemmer = nltk.stem.porter.PorterStemmer()
d = enchant.Dict("en_US")
```

```
fids = hotel_corpus.fileids()
docs1 = []
for fid in fids:
    doc_raw = hotel_corpus.raw(fid)
    doc = nltk.word_tokenize(doc_raw)
    docs1.append(doc)
docs2 = [[w.lower() for w in doc] for doc in docs1]
docs3 = [[w for w in doc if re.search('^[a-z]+$', w)] for doc in docs2]
docs4 = [[w for w in doc if w not in stop_list] for doc in docs3]
docs5 = [[w for w in doc if d.check(w)] for doc in docs4]
hotel_docs = [[stemmer.stem(w) for w in doc] for doc in docs5]
```

Topic Modelling

In general scenario, reviews or any other text data is unstructured and doesn't specifically talk about just one element/topic. Specially, when it comes to reviews people tend to write reviews considering different elements of an item/place that they are reviewing.

Our dataset contains reviews given by people for 1000 hotels across the Unites States. When we read through sample reviews, each review was speaking about various elements/topics of a hotel. For example, consider the below review for the Hotel Russo Palace, Mableton taken from our dataset –

"If you're looking for a hotel where the action is, go elsewhere. The hotel is on the Lido, which is about a 10-15 minute waterbus ride away from Venice. However, if you're looking for a place on the ***Lido di Venice****, this is* ***a well-priced option****. The* ***staff*** *was great with handling/recommending dinner reservations, even during an insane Carnevale weekend. The doors are a little thin and since the building is old, the* ***soundproofing needs improvement****. The quarters are cramped, but the room comes with* ***breakfast****. If we wanted to stay on the Lido again, this would be a good option, but next time I think we'll opt for a place closer to the action. Recommended."*

If we notice clearly, this reviewer talks about different things about the hotel as highlighted.

1. Location – He speaks about the Hotel location and suggests about the nearby places.
2. Value for Money – This is well priced option if a person is looking for a hotel on the Lido di Venice.
3. Staff – According to the reviewer, the staff was good.
4. Room Amenities – Building is too old and needs soundproofing, so the reviewer rates negative for the room amenities.

If we study the overall review, the sentiment is positive but in detail if we break down by various aspects spoken in the review like above, we see that the sentiment for the Room Amenities is negative. In regard to the customers preference, each customer's preference is different. Therefore, based on the review, this hotel is not suitable for a person who is going on a leisure trip and would prefer nice room amenities over other aspects like location, staff, etc. Hence, it is important to analyze sentiment of various aspects/topics that are spoken in a review.

STEP 1

First step is to understand the most talked about topics in all the reviews, therefore for this purpose we used Topic Modeling.

Topic Modeling is a process to automatically identify topics present in a text object and to derive hidden patterns exhibited by a text corpus. Topic Modeling is different from rule-based text mining approaches that use regular expressions or dictionary-based keyword searching techniques. It is an unsupervised approach used for finding and observing the bunch of words (called "topics") in large clusters of texts. Topics are formed by finding co-occurrence of words.

Latent Dirichlet Allocation for Topic Modelling

LDA assumes documents are produced from a mixture of topics. Those topics then generate words based on their probability distribution. Given a dataset of documents, LDA backtracks and tries to figure out what topics would create those documents in the first place.

We used gensim module for running the LDA model. The inputs given to LDA model are Corpus, Dictionary & Number of Topics.

STEP 2

As LDA is unsupervised model, we tried running the model multiple times with different no. of topics to get meaningful topics. Then we finalized the model with 7 topics which gave us meaningful topics to interpret. We also manually labelled all the above topics based on the words distribution for each topic.

The output of LDA model shows top 10 words for 7 topics –

1. Location

```
(0, '0.028*"locat" + 0.015*"park" + 0.014*"conveni" + 0.012*"right" +
0.012*"check" + 0.011*"fine" + 0.010*"except" + 0.009*"view" +
0.009*"easi" + 0.009*"awesom"')
```

2. Staff

```
(1, '0.061*"stay" + 0.057*"staff" + 0.044*"friendli" + 0.033*"help" +
0.027*"would" + 0.019*"definit" + 0.018*"wonder" + 0.016*"place" +
0.016*"enjoy" + 0.016*"time"')
```

3. Value for Money

```
(2, '0.011*"best" + 0.010*"stay" + 0.010*"breakfast" + 0.010*"night" +
0.008*"one" + 0.008*"valu" + 0.008*"highli" + 0.007*"complaint" +
0.007*"happi" + 0.007*"money"')
```

4. Food

```
(3, '0.024*"servic" + 0.023*"love" + 0.021*"everyth" + 0.019*"restaur"
+ 0.015*"excel" + 0.015*"locat" + 0.014*"close" + 0.014*"amaz" +
0.013*"food" + 0.012*"decor"')
```

5. Nature of Stay

```
(4, '0.010*"work" + 0.009*"like" + 0.008*"smell" + 0.008*"need" +
0.008*"date" + 0.008*"could" + 0.007*"door" + 0.007*"ask" +
0.007*"smoke" + 0.007*"get"')
```

6. Room amenities

```
(5, '0.020*"overal" + 0.012*"experi" + 0.012*"hot" + 0.011*"king" +
0.011*"water" + 0.011*"shower" + 0.011*"thank" + 0.010*"bed" +
0.009*"big" + 0.009*"reason"')
```

7. Cleanliness/Comfort

```
(6, '0.073*"clean" + 0.061*"comfort" + 0.037*"nice" + 0.037*"bed" +
0.029*"room" + 0.023*"quiet" + 0.015*"price" + 0.014*"well" +
0.014*"pillow" + 0.013*"spaciou"')
```

Step 3 – Saved the finalized LDA model and used these topics generated for further analysis.

Step 4 – Now that we know all reviews which are spoken about, we will use these topics generated by LDA model to understand what each document (which is at a sentence level of each review) speaks about, therefore we generated document topic distribution matrix for each document.

Output – Data frame that shows the topic score for each document.

File_ID	Hotel Name	Staff	Location	Cleanliness/Comfor	Nature of stay	Value for money	Restaurant/Food	Room amenities
1785 Inn - 1.txt	1785 Inn	0.040306099	0.040215433	0.040135887	0.040375393	0.040178154	0.04149738	0.757291675
1785 Inn - 10.txt	1785 Inn	0.052775029	0.683612823	0.05270727	0.052717295	0.052718077	0.05274165	0.052727818
1785 Inn - 11.txt	1785 Inn	0.641890883	0.05962044	0.059762776	0.059667319	0.059727415	0.059663277	0.059667923
1785 Inn - 12.txt	1785 Inn	0.044738278	0.167447358	0.608945429	0.044858672	0.044692982	0.044658892	0.044658348
1785 Inn - 13.txt	1785 Inn	0.045416646	0.045398436	0.045457054	0.045698829	0.045395479	0.727192938	0.045440648
1785 Inn - 14.txt	1785 Inn	0.041729018	0.041644964	0.041678093	0.749858737	0.041702144	0.041739766	0.041647308
1785 Inn - 15.txt	1785 Inn	0.230904534	0.039533246	0.03955166	0.433856159	0.039548997	0.176911309	0.039694071
1785 Inn - 16.txt	1785 Inn	0.036974639	0.435835242	0.036973003	0.037047222	0.126054764	0.290102452	0.037012644
1785 Inn - 17.txt	1785 Inn	0.049551133	0.048267625	0.048062906	0.356580764	0.048149027	0.196429223	0.252959341
1785 Inn - 18.txt	1785 Inn	0.04468102	0.225814879	0.378509969	0.044968527	0.044681016	0.044727955	0.216616616
1785 Inn - 19.txt	1785 Inn	0.10347686	0.042460781	0.47232765	0.042699303	0.042600121	0.253971219	0.042464074
1785 Inn - 2.txt	1785 Inn	0.042106271	0.042140611	0.042147677	0.513902783	0.042101294	0.042489156	0.275112212
1785 Inn - 20.txt	1785 Inn	0.351397693	0.041725233	0.041507412	0.216787919	0.265070319	0.041848004	0.041663405
1785 Inn - 21.txt	1785 Inn	0.048784416	0.048799619	0.048781171	0.514117837	0.048782371	0.04880337	0.2419312

Step 5 – As we have already split our reviews into sentence level, we are only interested in Top most prevalent topic for each document. So, we ranked the topic distributions for each document and picked top topic with highest topic score for each document.

Output - Data frame that shows most prevalent topic for each document.

File_ID	Hotel Name	Top1 Topic
1785 Inn - 1.txt	1785 Inn	Room amenities
1785 Inn - 10.txt	1785 Inn	Location
1785 Inn - 11.txt	1785 Inn	Staff
1785 Inn - 12.txt	1785 Inn	Cleanliness/Comfort
1785 Inn - 13.txt	1785 Inn	Restaurant/Food
1785 Inn - 14.txt	1785 Inn	Nature of stay
1785 Inn - 15.txt	1785 Inn	Nature of stay
1785 Inn - 16.txt	1785 Inn	Location
1785 Inn - 17.txt	1785 Inn	Nature of stay
1785 Inn - 18.txt	1785 Inn	Cleanliness/Comfort
1785 Inn - 19.txt	1785 Inn	Cleanliness/Comfort
1785 Inn - 2.txt	1785 Inn	Nature of stay
1785 Inn - 20.txt	1785 Inn	Staff
1785 Inn - 21.txt	1785 Inn	Nature of stay

Sentiment Analysis

Sentiment Analysis is the next important approach in text mining, here we performed the sentiment analysis on each sentence of the review. Let's have a look at an example review.

In this review, it's natural to summarize that there is a positive sentiment. But there is a negative point which needs to be addressed by the hotel management if multiple customers are facing the issue. Thus, Sentence level sentiment analysis enables us to identify the sentiment of each sentence for which we have already identified the topics in the prior section. We used two different approach for classifying the sentiment of the reviews which are explained as below:

a) Lexicon Approach

The lexicon-based approach involves calculating orientation for a document from the semantic orientation of words or phrases in the document (Turney 2002). Sentiment lexicons or a list labelled words by their sentiment polarity are used to match with the words in the documents to calculate their overall polarity. We have used a python package Sentlex that uses WordNet's sentiment lexicon SentiWordNet and a sentiment classifier that considers the negation of sentiment words that can change the polarity of a document. It takes the pre-processed document as input and produces a positive and negative score based on the words in the document compared with the Lexicon words.

```
In [6]: classifier.classify_document(input_text, tagged=False, L=SWN, a=True, v=True, n=False, r=False, negation=False
Out[6]: (1.1118254346272922, 0.16043343653250774)
```

The detailed results of the classifier are as below, identifying negation words, positive and negative indicator words to calculate the score. The scores are calculated for all documents which are sentences in a review of the hotels and the documents are labelled as overall positive/negative on the topic which has the highest document topic distribution score on that document. This classification is then evaluated on a ground truth manually labelled document set on a confusion matrix.

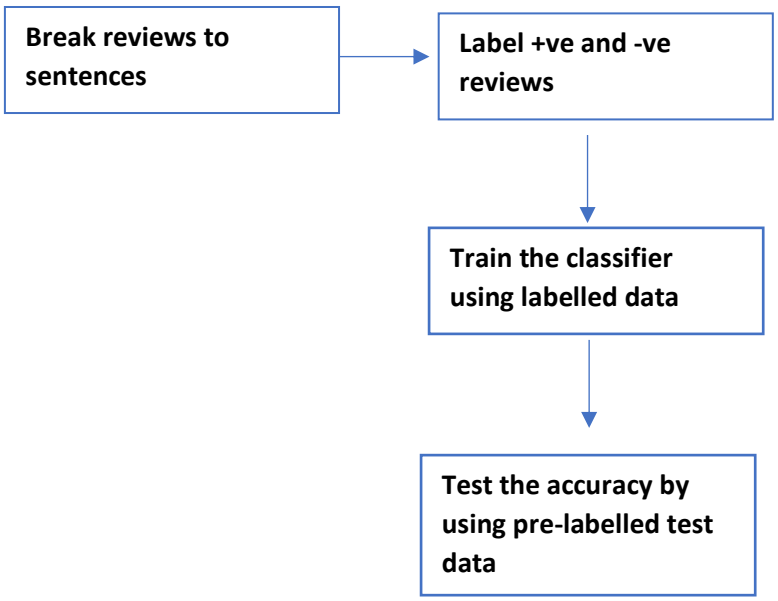
```
In [7]: classifier.resultdata

Out[7]:
('annotated_doc': 'we/PRP had/VBD##NEGAT:NONEG##POS:0.00986842105263##NEG:0.026315789473
a/DT great/JJ##NEGAT:NONEG##POS:0.3125##NEG:0.0 time/NN at/IN the/DT tirreno/NN hotel/NN ,/, very/RB friendly/RB and/CC helpful/JJ##NEGAT:NONEG##POS:0.25##NEG:0.0 ,/,
nothing/NN was/VBD##NEGAT:NONEG##POS:0.0288461538462##NEG:0.0 ever/RB too/RB much/RB trouble./NNP the/DT rooms/NNS were/VBD##NEGAT:NONEG##POS:0.0288461538462##NEG:0.0
in/IN excellent/NN condition/NN ,/, very/RB clean/JJ##NEGAT:NONEG##POS:0.286764705882##NEG:0.0441176470588 and/CC comfortable/JJ##NEGAT:NONEG##POS:0.195##NEG:0.09 ./.',
 'doc': 'we had a great time at the tirreno hotel, very friendly and helpful, nothing was ever too much trouble. the rooms were in excellent condition, very clean and
 comfortable.',
 'found_list': ['had/VBD',
  'great/JJ',
  'helpful/JJ',
  'was/VBD',
  'were/VBD',
  'clean/JJ',
  'comfortable/JJ'],
 'resultneg': 0.16043343653250774,
 'resultpos': 1.1118254346272922,
 'tokens_found': 7,
 'tokens_negated': 0,
 'unscored_list': []}
```

b) Classifier Approach

Naïve Bayes' classifier is a probabilistic classifier based on the Bayes' theorem, considering Naïve (Strong) independence assumption. An advantage of Naïve Bayes' is that it only requires a small amount of training data to estimate the parameters necessary for classification. Abstractly, Naïve Bayes' is a conditional probability model. Despite its simplicity and strong assumptions, the naïve Bayes' classifier has been proven to work satisfactorily in many domains. Bayesian classification provides practical learning algorithms and prior knowledge and observed data can be combined. In Naïve Bayes' technique, the basic idea to find the probabilities of categories given a text document by using the joint probabilities of words and categories. It assumes of word independence. The starting point is the Bayes' theorem for conditional probability, stating that, for a given data point x and class C[4]:

$$P(C / x) = P(x/C)/P(x) \quad (1)$$



As stated in the above flow chart, we trained the NLTK Naïve Bayes classifier using labelled data of 650 review sentences and tested the accuracy using the pre-labelled test data of 250 review sentences. The prediction accuracy of Naïve Bayes Classifier is 71%

c) Model comparison using evaluation metrics

Contingency table and Evaluation metrics for Lexicon classifier:

		Predicted label	
		Positive	**Negative**
Ground Truth Label	**Positive**	153	16
	Negative	28	48

Evaluation Measure	Value	Formula
Accuracy	82.04%	(TP+TN)/(P+N)
Error rate	17.96%	(FP+FN)/(P+N)
Precision	84.53%	TP/(TP+FP)
Sensitivity (TP rate)	90.53%	TP/P
Specificity (TN rate)	63.16%	TN/N

Contingency table and Evaluation metrics for Naïve Bayes classifier:

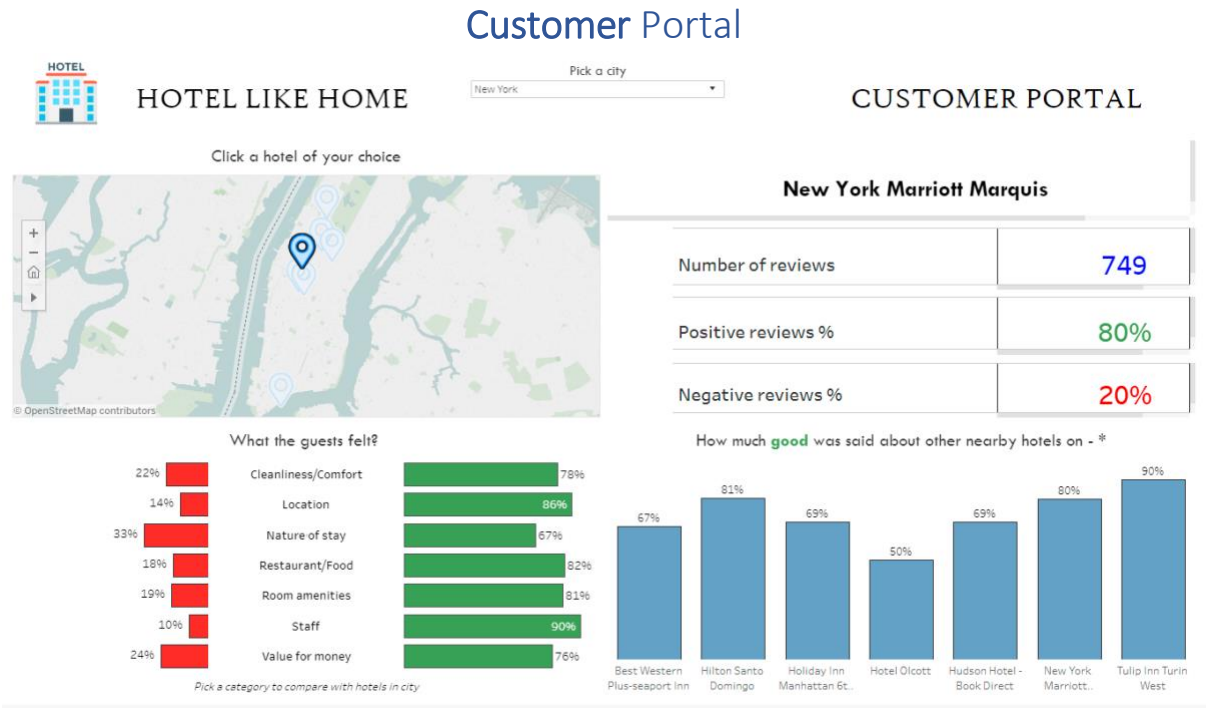
		Predicted label	
		Positive	**Negative**
Ground Truth Label	**Positive**	128	28
	Negative	43	46

Evaluation Measure	Value	Formula
Accuracy	71.06%	(TP+TN)/(P+N)
Error rate	28.94%	(FP+FN)/(P+N)
Precision	74.85%	TP/(TP+FP)
Sensitivity (TP rate)	82.05%	TP/P
Specificity (TN rate)	51.68%	TN/N

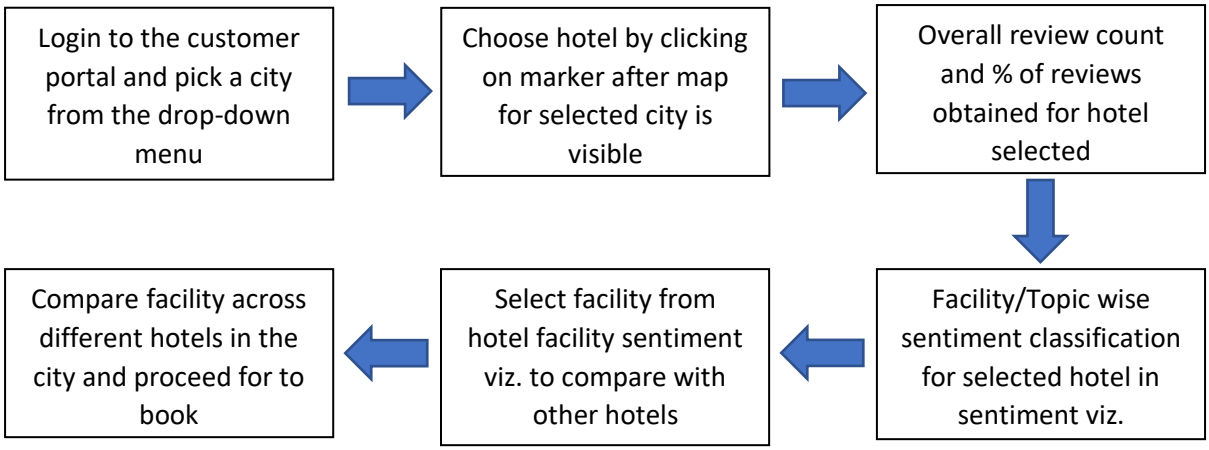
Because of higher accuracy and precision, we have chosen Lexicon classifier. We have used the results from Lexicon classifier for our further analysis.

Hotel Like Home Application

Hotel like Home application is a comprehensive one-stop solution for our stake-holders through two portals, i.e. Customer portal & Management portal. The outputs from the text analytics framework are visualized in these portals for ease of hotel booking for the customer and hotel performance comparison for the management side. The application is built on tableau.



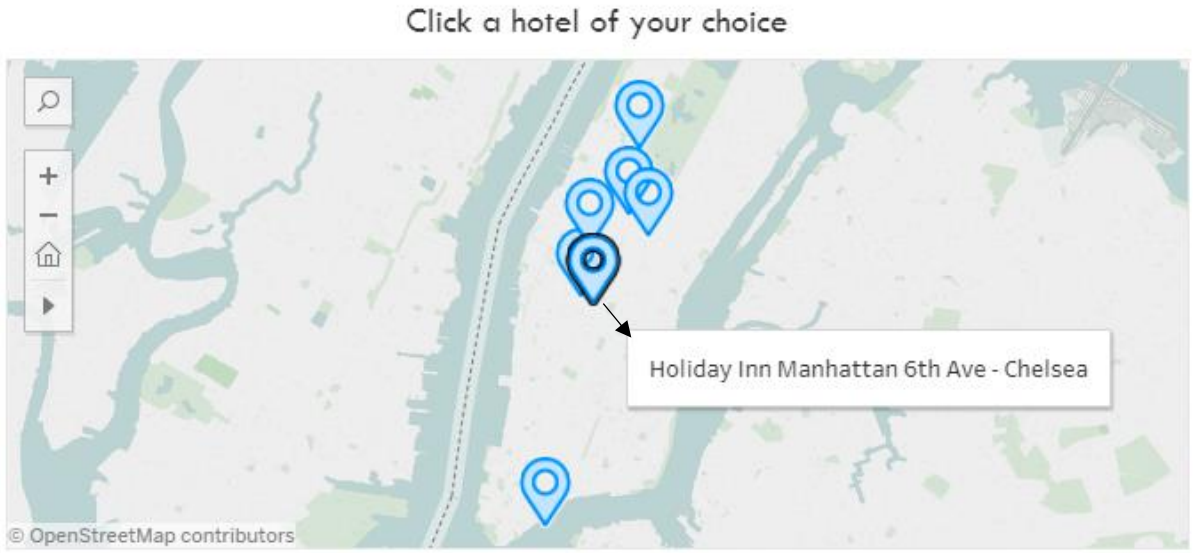
Instruction flow for Customer Portal



The customer portal has 4 visualizations built in:

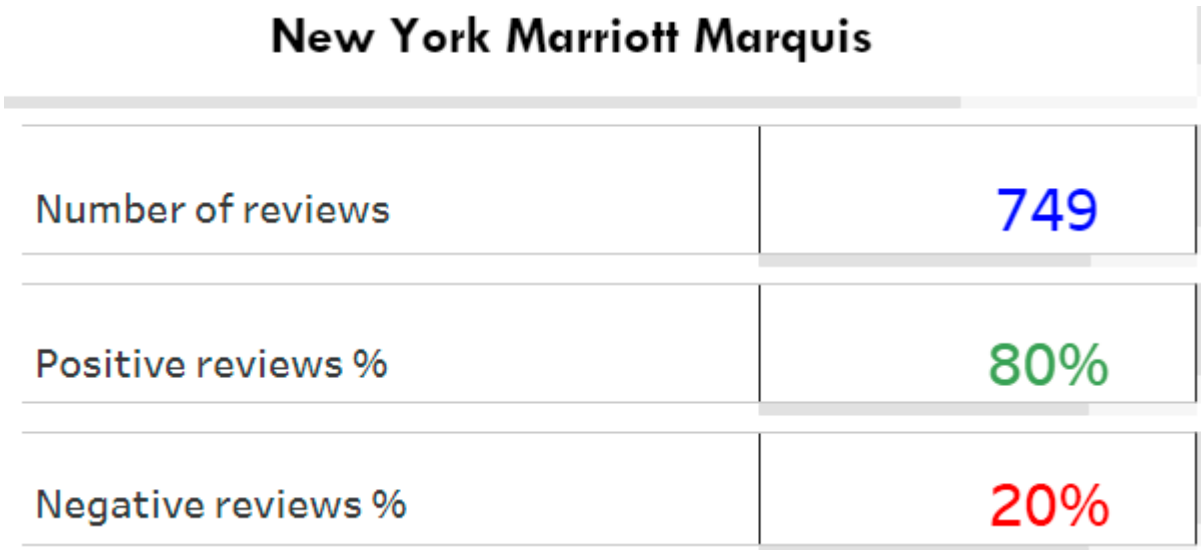
1. Map location hotel selector
2. Overview of hotel reviews
3. Hotel Facility sentiment polarities
4. Facility comparison across city

Map Location Hotel Selector



After choosing a city to book a hotel, the Hotel selector map gets displayed. Guests can have a look about hotels by clicking on the location markers. Hovering over the markers displays the name of the hotel.

Overview of hotel reviews

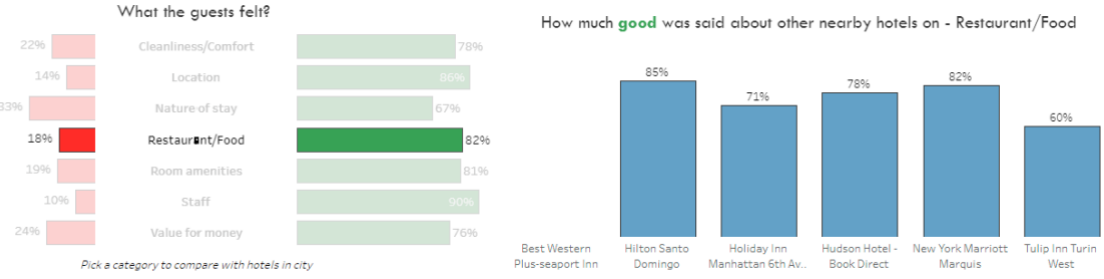


When a hotel is selected on the map, the total number of reviews and the percentage of positive and negative reviews can be seen in this segment of the portal. This gives a overall picture of the hotel but not about the individual facilities which is explored in the next segment of the portal.

Hotel facility sentiment polarities

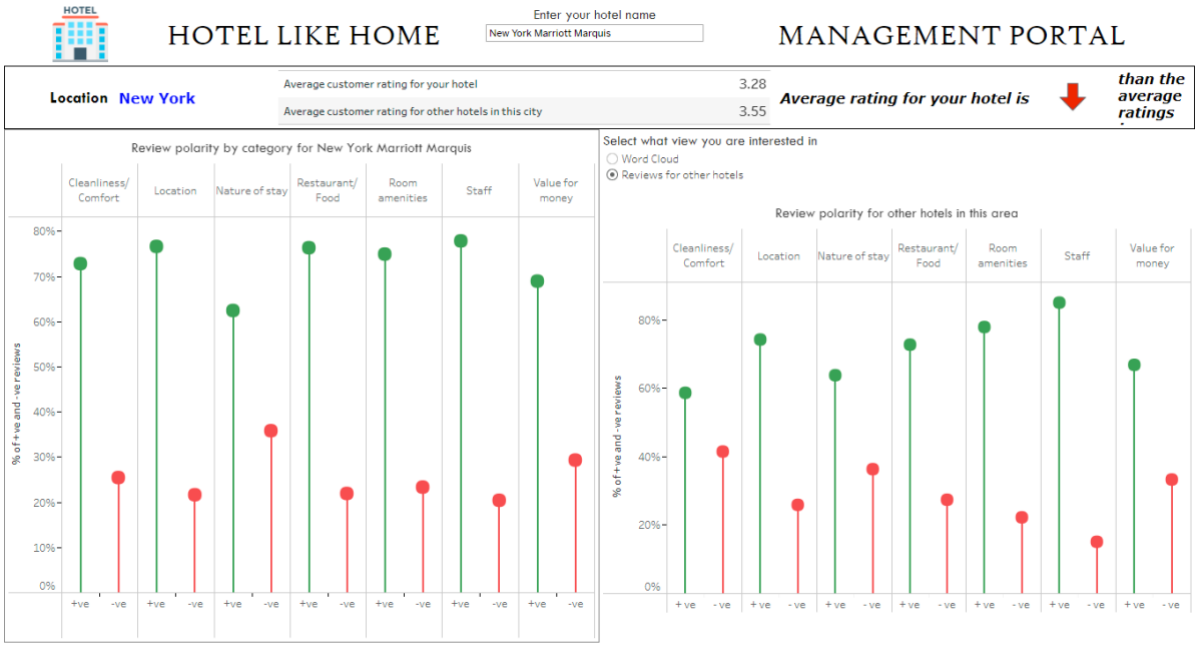
A customer wants to know how the prior guests felt about the facilities of the hotel which he/she might want to book. So, we have used LDA to perform topic modelling on all the reviews to find out the most prevalent topics in the reviews. After a topic is assigned to a review, we perform sentiment analysis under each of the topics to get the sentiment polarity percentage for each topic in a hotel. This polarity percentage is visualized in the above diverging bar chart. A guest can look into this viz and decide to book the hotel if the facility he/she is looking for is good in the hotel. The guest might also want to compare how the hotel fares against its competitors in the city for a facility which is of interest.

Facility comparison across city

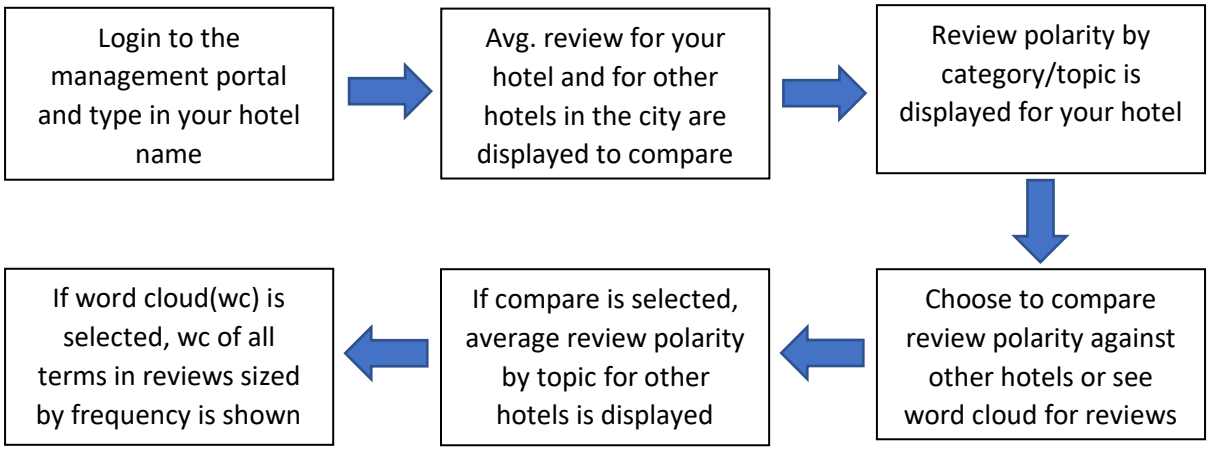


Let's say the customer wants to know what the guests felt on Restaurant/Food in other nearby hotels in the city. It is seen that 82% of people have spoken good about food for the hotel selected → New York Marriott Marquis and from the comparison chart, 86% of people have felt the food is good in Hilton Santo Domingo. The bars represent only the positive polarities of a selected topic. So, if a guest's preference is Food, Hilton Santo Domingo can be booked rather than Marriott Marquis

Management Portal



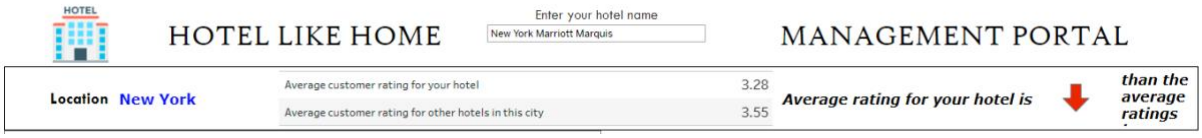
Instruction flow for Management Portal



The management portal has 4 visualizations built in:

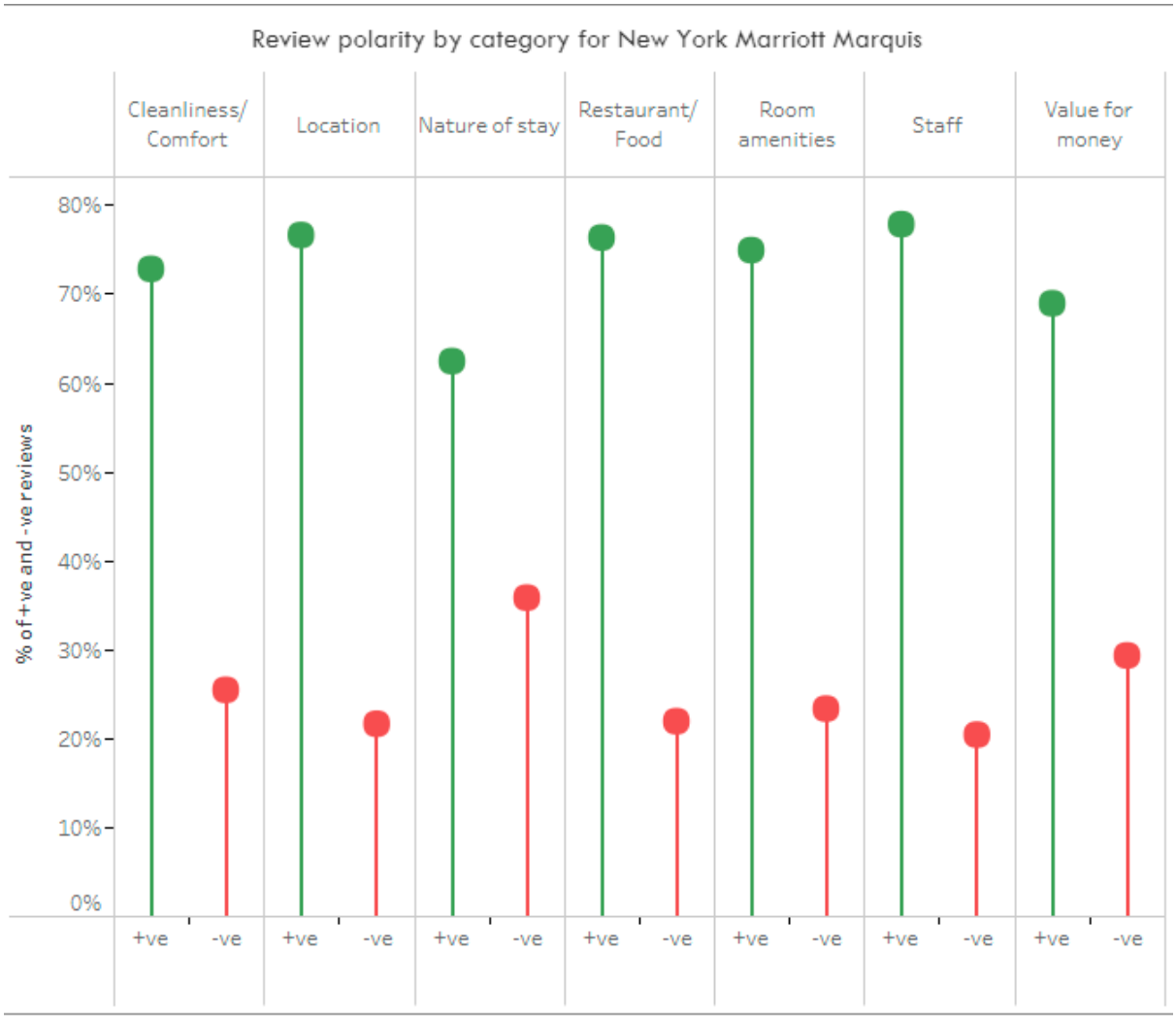
1. Average review comparison pane
2. Review polarity by facility for your hotel
3. Review polarity comparison for facilities against other hotels in the city
4. Word Cloud for reviews for your hotel

Average review comparison pane



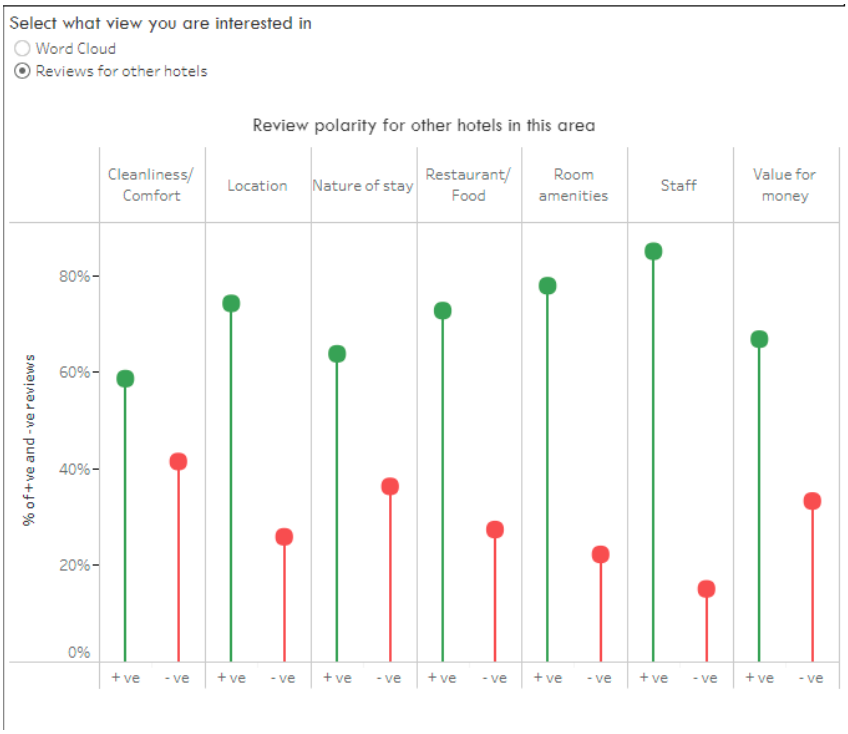
When your hotel name is entered, average customer rating for your hotel and average customer rating for other hotels in the city is displayed. So, a comparison can be made to see how your hotel is performing against the other hotels in the city. Now, an overall idea has been obtained but if the management wants to see which facilities are doing good and which needs to be improved, a study across facilities is needed which takes us to the next segment of the portal.

Review polarity by facility for your hotel



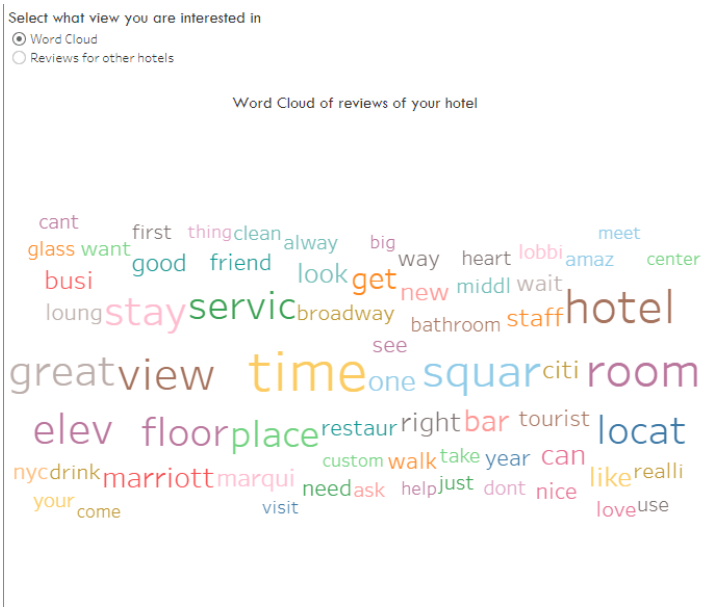
This segment of the portal displays the review polarity by category for your hotel. A lollipop dodged bar chart for each topic is used to represent the % sentiment. For example, from the above viz., it is seen that nature of stay had a higher percentage of negative reviews than the other facilities. So, this facility can be focused for improvement.

Review polarity comparison for facilities against other hotels in the city



After selecting polarity comparison, we can view the average review polarity percentage across topics for other hotels in the city. Here, it is seen that in New York, all the hotels are doing very well in Staff so if your hotel is getting a lot of negative reviews about the staff, more emphasis could be placed on this topic to ensure your hotel doesn't lose the competitive edge.

Word Cloud for reviews for your hotel



When word cloud view is selected, a word cloud of all the terms in the reviews is displayed. The words are sized by how frequently they appear in the reviews. Higher the frequency, bigger the

words. For ex, great,view,locat appear big which means that they have been spoken more about in the reviews. This word cloud gives an overall picture to the management about what is being spoken in the reviews about their hotel.

Discussion and Gap analysis

The review data was a mix of English, German and French language reviews for the hotels. In the pre-processing step, we removed all the non-English words from the documents before creating the corpus for further analysis. This was a considerable data loss when the topic modelling and sentiment analysis were performed.

> Wir buchten das Hotel fr 4 Nchte in der Woche vor Ostern. Der Aufenthalt war durchweg angenehm. Das Frhstcksbuffet war reichhaltig und frisch (allerdings rate ich von dem Rhrei ab). Wenn etwas aufgebraucht war, wurde es umgehend aufgefllt. Vom Frhstcksraum aus konnte man den Blick ber die Lagune genieen. Obwohl unser Zimmer im Erdgeschoss zur Nebenstrae hin lag, war es dort sehr ruhig. Die Einrichtung ist geschmackvoll und zweckmig. An Sauberkeit mangelte es nicht. Obwohl wir nicht den Anspruch stellten, wurden die Handtcher tglich gewechselt. Das einzige Manko war die Temperatur des Duschwassers. Es wechselte stndig von ziemlich kalt zu sehr hei. Erwhnenswert ist auch das Waschbecken. Der Wasserhahn ist ziemlich klein und so ber dem hervorstehenden berlauf angebracht, dass es schon schwierig ist, die Hnde darunter zu waschen.

In the initial project methodology design, we had grouped the review data from all the users at each hotel level and created the corpus for analysis for reducing computing power. After performing the initial pre-processing steps, the topic modelling step using LDA produced hazy and uninterpretable topics as the document topic distribution didn't have clear separation. As LDA relies on the co-occurrence of words to put together the topics grouping the reviews caused the words related to all topics the users have discussed to co-occur and the accuracy was very poor. So, we modified the corpus creation step design to create the corpus documents by breaking down the reviews to a more nuclear format of sentences and re-performed the topic modelling and saw immense improvement in the results. As the co-occurrence was corrected, the document topic distribution provided 7 clearly distinguishable and domain relevant topics which we've interpreted and done sentiment analysis on.

The sentiment analysis was performed using a simple Lexicon based sentiment classifier which doesn't account for sarcasm and strength of sentiment words. So, the analysis has been scoped only on polarity level and not on strength level which led to some reviews like below to not be accurately classified.

> My Experiences:
> 1) Laughter is the best therapy: The staff would gladly laugh at all your complaints, like they're not supposed to cater to anything.
> 2) Always look forward to tomorrow: "Tomorrow", that's their answer to almost all your grievances.
> 3) Learn to adjust: Even if there is no latch on your bathroom, with 3 room-mates
> 4) The worst is yet to come...: The very first day housekeeping was done (in over a week), all the water from washing utensils choked the drain, and our room became a sweet smelling pool :)
> P.S. Just moved out. Came for 2 months, moved out in 10 days. Save yourself the trouble.

Future work and conclusion

In conclusion, the results of the methodology have produced highly accurate results in line with the business problem in hand. There is scope for future extension of our work from the analysis process and dashboard design for stake-holders' perspectives. The non-English reviews in the data can be accounted for separately instead of being removed in the pre-processing step to perform the LDA topic modelling and sentiment analysis with an appropriate language sentiment lexicon. A seeded

topic modelling approach can be applied in place of a fully unsupervised LDA with topics as per stakeholder needs to identify any missed-out topics of interest. The sentiment analysis algorithm can be improved on by adding on the word level lexicon-based classifier to one that can handle sentiment strength, by word strength, capitalization and emojis and to also understand context and sentiment. The dashboard currently holds a word cloud for the hotel management to view, which is in a hotel level, this can be enhanced to a sentiment level word cloud for a clear view of what is liked, what needs improvement in the hotel. The dashboard can be enhanced to a recommendation engine, where a user keys in a location and the app can display hotels based on entered or machine learned user preferences.

References

1. *Google Is Dominating the Review Market (2018). Retrieved 02 August 2018*, from https://www.reviewtrackers.com/online-reviews-survey/
2. Tian, X., He, W., Tao, R., & Akula, V. (n.d.). Mining Online Hotel Reviews: A Case Study from Hotels in China.
3. *Hotel Reviews (2017). Retrieved 20 July 2018*, from https://data.world/datafiniti/hotel-reviews
4. Dey, L., Chakraborty, S., Biswas, A., Bose, B., & Tiwari, S. (2016). Sentiment Analysis of Review Datasets Using Naïve Bayes' and K-NN Classifier. *International Journal of Information Engineering and Electronic Business, 8*(4), 54-62. doi:10.5815/ijieeb.2016.04.07
5. Shivam, B. (2016, August 24). Beginners Guide to Topic Modelling in Python. Retrieved from https://www.analyticsvidhya.com/blog/2016/08/beginners-guide-to-topic-modeling-in-python/
6. Topic modelling with LDA. https://medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05
7. Sentiwordnet sentiment lexicon. http://sentiwordnet.isti.cnr.it/
8. Sentlex - Python library for performing lexicon-based sentiment analysis. https://github.com/bohana/sentlex/blob/master/README.md