Text-Based Hotel Review Retrieval System

Lavanya Adhikari
lavanya21043@iiitd.ac.in
MT21043

Arnab Chatterjee
arnab21015@iiitd.ac.in
MT21015

Yash Goyal
yash21106@iiitd.ac.in
MT21106

Aurko Mitra
aurko21166@iiitd.ac.in
MT21166

Shubham Sharma
shubham20324@iiitd.ac.in
MT20324

ABSTRACT

There is a cornucopia of hotel reviews at one's disposal these days. Almost 80% of travellers say that they use sites like TripAdvisor before making a decision about the hotels they decide to spend their time in. Our project centres around giving a text-based search interface to the user, which they can use for the retrieval of the hotels pertaining to their needs in a ranked manner. This ranking will be done by first obtaining the average aspect ratings while parsing the dataset itself, and then performing a range search on the index to identify all reviews for that hotel that have a rating greater than the rating in query for each aspect identified. We implemented various models while implementing this where Support Vector Machin(SVM) gave us the best outcome with 93.7% accuracy. For ranking, we also take into account the quantity of exact phrase matches. The aspect fields were then sorted in descending order to achieve a good ranking. The interface of our project was made using primarily HTML, CSS, and Javascript. The Python Flask web framework was used to connect the front-end and back-end.

KEYWORDS

Information Retrieval, Text-based retrieval, Aspect Segmentation, Aspect Based Sentiment Analysis, Ranking

1 PROBLEM STATEMENT

Our project concentrates around offering the user with a text-based search interface via which they may obtain hotels that satisfy their requirements in ranked order. To accomplish this, we implemented aspect segmentation, and then utilised the cosine similarity metric to determine the aspect of each sentence, and then assigned the aspect with which it had the greatest resemblance. The sentiment score for each statement was then determined. Several classification models were trained utilising sentences as independent features and the aspect as the target variable. Accuracy is the evaluation statistic used to judge the performance of our model. Finally, the hotels retrieved using our ranking and retrieval module are shown as an output on our front-end depending on their total rating.

2 INTRODUCTION

The importance of the hotel reviews by the consumers can be recognised with this study conducted recently on more than 23,000 TripAdvisor users which showed that over 81% of users tend to read the reviews before making a decision. Over 86% stated that seeing the reviews on such sites made them feel more confident in their booking decisions, as well as over 52% stated that they would never book a hotel with no reviews. So when faced with such statistics, we can realise how important user reviews are for consumers.

Typical reviews, like the one displayed in Figure 1, talk about multiple aspects of the hotel, for example, cost, room condition, and administration, yet the reviewer just gives a general rating of the hotel; without an explicit and clearly defined rating on every aspect, a user would not be able to easily know the reviewer's opinion on each aspect. Going past the general rating to comprehend a reviewer's viewpoints on different components is significant on the grounds that various reviewers might give similar hotels similar by and large evaluations for very various reasons. For instance, one commentator might have partaken in the setting while another may have enjoyed the room. To help customers recognize the two, grasping a commentator's appraisal of every one of the essential categories is imperative. Regardless of whether we acquire the

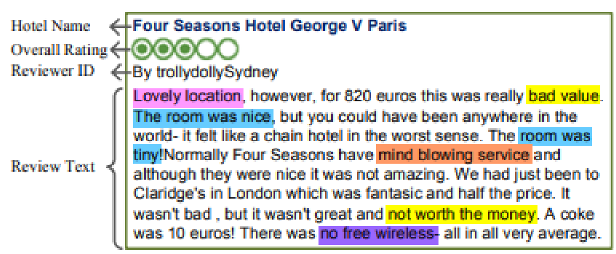

Figure 1: An example of a normal review

rating that every reviewer doles out to every aspect, it will be lacking in light of the fact that it might mean different things to different reviewers. A reviewer's definition of "cheap" will also differ based on where they are staying and how well they are aware of the prevailing rates in that area. Intuitively, when a reviewer is more concerned with the location, they are more willing to endure a greater cost. To comprehend such nuanced distinctions, it is vital to explain the relative importance weight that a reviewer assigns to each aspect when providing the overall rating.

We wanted to provide a more systematic and functional way in which users can retrieve the reviews of the hotels which are suited to their own niche needs and demands. Our project centres around giving a text-based search interface to the user, which they can use for the retrieval of the hotels pertaining to their needs in a ranked manner. Right now we do not have well-established systems which allow the user to search hotels based on certain particular aspects, or rate the hotels based on those particular aspects only. This system will allow the user to select a hotel that ranks highest on the aspect that matters the most to them.

We used the HotelRec [1], a massive hotel recommendation dataset based on TripAdvisor, with 50 million reviews, which contained the URL of the user's profile and hotel, the date, the overall rating, the title of the review, the written text, and any subratings that were provided. We executed aspect segmentation on this dataset after conducting basic preprocessing, which divides the sentences in a review into subsets according to each aspect. The cosine similarity metric was then used to assess the aspect of each sentence, and the aspect with which it bore the most resemblance was allocated. After that, the sentiment score for each statement was calculated. Finally, using phrases as independent features and the aspect as the target variable, multiple classification models were trained. The Accuracy and Confusion Matrix is the evaluation statistic used to assess our model's performance. Finally, the hotels retrieved using our ranking and retrieval module are shown as an output on our front-end depending on their total rating.

3 LITERATURE SURVEY

In Tran et al. [7] we could see how the authors had applied various machine techniques, both supervised and unsupervised on a set of fixed dataset in order to identify the various aspects mentioned by the users in their reviews and then they measured customer's opinions towards those aspects using a deep learning model. But this study was conducted more toward the benefit of the hotel owners and the size of the dataset was limited. Additionally, the tracking sentiment for a topic over time does not take into consideration the seasonal effects especially in the hospitality industry.

The Duan et al. [2] did something similar but they expanded their dataset, by taking reviews from various sites, over various seasonal periods. They used the sentiment analysis technique to decompose user reviews into five pre-mediated dimensions and based on Those dimensions tried to find their effect in shaping users' overall evaluation and content generating behavior. But again the pre-concieved dimensions and again limited dataset, along with the side for which the study in done for does not help the users.

In this recent paper. Zahid [9] sentiment analysis is performed on the basis of user reviews using three different classifiers. The classifiers used in this paper are "Naive Bayes","Random Forest" and "Support Vector Machine", and performance metrics like precision,recall,accuracy and F1-score are used.It was concluded that multinomial naive bayes with appropriate parameter settings, performs the best. Again with this paper we can see that even recently most of the crucial work done in this field is done on the sentiment analysis of the reviews, rather that what we are proposing.

Coming to the aspect nature of our proposed system, in Gojali and Khodra [3] consists of aspect and sentiment extraction, and determination of the sentiment's orientation. In this paper, a system is proposed to extract the aspect sentiment pair and compute the rating for each grouped aspect. Approach starts with selecting the subjective sentences in the reviews. Then, it extracts aspects and opinions from the sentences, and determines the orientation of the sentiment. Aspects are clustered together by using WordNet with the prior knowledge of the aspect categories. Finally, it measures the rating for each aspect category, and evaluated using criterias such as precision, recall, and F1-Measure.

Summarization is a general-purpose strategy for dealing with information overload. Sentiment informed summaries are strongly favoured than non-sentiment baselines, implying the utility of modelling sentiment and aspects when summarising opinions. However, Existing works such as Hu and Liu [4] on aspect-based summarization aimed solely at gathering all reviews and representing main opinions on various elements of a specific topic. While aggregated opinions can provide a basic picture of a topic, the details in each review are lost; also, the distinctions between reviews/reviewers are not taken into account, therefore the aggregated sentiment is based on reviewers with diverse tastes.

In Titov and McDonald [6] suggested extracting characteristics while also predicting related ratings: they utilise topics to characterise aspects and integrate a regression model fed by ground-truth ratings. They have, however, made the assumption that the aspect evaluations are explicitly included in the training data. In contrast, we presume that the aspect evaluations are latent, which is a more generic and realistic circumstance.

Most notably, none of the preceding work takes into account the reviewer's attention on certain aspects, i.e. aspect weight. Our research seeks to deduce aspect evaluations and weights at the level of individual reviews. Till we reached the work by Wang et al. [8] which implements Aspect segmentation which can explore review text data with companion overall ratings to simultaneously discover: 1) latent topical aspects, 2) latent ratings on each identified aspect, and 3) latent weights placed on different aspects by a reviewer.

We are taking this work further by implementing Aspect Based Sentiment Analysis [5], and then implementing various models for classification.

4 DATASET

We used the HotelRec [1], a massive hotel recommendation dataset based on TripAdvisor, with 50 million reviews. We collected the following information for each review: the URL of the user's profile and hotel, the date, the overall rating, the title of the review, the written text, and any sub-ratings that were provided, as you can see in fig 2.

{"**hotel_url**":"Hotel_Review-g194775-d1121769-Reviews-Hotel_Baltic-Giulianova_Province_of_Teramo_Abruzzo.html", "**author**": "ladispoli", "**date**": "2010-02-01T00:00:00", "**rating**": 4.0, "**title**": "Great customer service", "**text**": "Great customer service and good restaurant service is what made this experience so wonderful for my family. Giulianuova is a pretty simple , not so \"wow\" town, with not very clean beaches as I was expecting when booking my reservation. What saved my vacation was staying at Baltic. Baltic is a very simple but functional hotel, but what makes it so special are the people that work there. It seems that as a part of their job training they had take some kind of class on humanity and spirituality, because the way they treat every single person with respect and smile is just amazing. Good job to the manager Massimo (who knows how to hire great people)!", "**property_dict**": ***{"sleep quality": 4.0, "value": 4.0, "rooms": 3.0, "service": 5.0, "cleanliness": 3.0, "location": 3.0}***}

Figure 2: A review in our dataset

All of the reviews in the dataset were assigned subratings ranging from 1 to 5 on various factors. For our project, we needed a dataset with reviews subratings that were consistently designated for the same criteria. As a result, we extracted around 15 million reviews labelled with subratings for the same six factors. We ran our models on 500k reviews extracted from this new dataset.

5 ASSUMPTIONS

While the most basic configuration of our problem would include identifying potentially undiscovered features as well as latent ratings/weights on various aspects, in this project, we assume that we are provided a few keywords describing each component. This assumption is reasonable because it is possible to explicitly declare the primary elements of any given entity type in this manner; also, such a setup allows the user discretion over which aspects are investigated.

6 METHODOLOGY

6.1 *Data Preprocessing*

We did these basic preprocessing steps on our dataset after extracting the reviews with subratings for the same 6 parameters.

- Converting words into lowercase
- Removing punctuations
- Removing stopwords
- Stemming each word using Port Stemmer

Aspects	Seed Words
Value	value, price
Rooms	room, space
Location	location, locate
Cleanliness	clean, dirty
Service	service, manager
Sleep Quality	comfortable, quite

Table 1: Initial Aspect Seed Words

We must ensure that the same aspects are used in our prediction because we only have ground-truth aspect ratings on the pre-defined 6 aspects. As a result, for each pre-defined aspect, we manually select a few seed words and feed them into the aspect segmentation method. The initial aspect terms utilised are shown in Table 1.

6.2 *Aspect Segmentation*

The goal of aspect segmentation is to separate the sentences in a review into subgroups that correspond to each aspect. Because we believe that only a few keywords are provided to explain each element, we created a boot-strapping technique to locate more appropriate phrases for each characteristic.We chose the selection threshold as 5. We basically divided up all of the reviews into sentences and stored them in a list, then matched the aspect keywords in each sentence of the list and counted the matching hits for each aspect. We label the sentence with an aspect, and if there is a tie, we label the sentence with numerous aspects. Then, for each word, we compute the Chi-square (χ^2) measure, which is the difference between the observed and predicted frequencies of the outcomes of a set of events or variables.

The words under each aspect are then ranked according to their χ^2 value, and the top p words for each aspect are added to the corresponding aspect keyword list. If the aspect keyword list is intact or the iteration step limit is exceeded, we output the annotated sentences with aspect assignments. Otherwise, we go back and start matching the aspect keywords in each sentence of the list from the beginning.

- Converting words into lowercase
- Removing punctuations
- Removing stopwords
- Stemming each word using Port Stemmer

Aspects	New Seed Words
Value	value, price, resort, fishing, beach,...
Rooms	room, space, hotel, stay, clean,...
Location	location, locate, cabin, museum, haveli,...
Cleanliness	clean, dirty, terrace, fresh, sanitary,...
Service	service, manager, paris, butterfly, basket,...

Table 2: After Aspect Segmentation Seed Words

After aspect segmentation, we would get k partitions of each review d, and represent them as a k × n feature matrix W_d, where W_{dij} is the frequency of word w_j in the text assigned to aspect A_i of d normalized by the total counts of words in the text of that aspect.

We deleted phrases that were not connected with any aspect after aspect segmentation. If we mandate that all evaluations include all six aspect descriptions, there will only be 780 reviews left covering 184 hotels. To prevent sparsity and missing aspect details in the review, we concatenated all the reviews commenting on the same hotel into a single "review" (dubbed "h-review") and averaged the overall/aspect ratings over them as the ground truth ratings.

6.3 *Aspect Based Sentiment Analysis*

Following that, we separated each review into sentences and constructed a Data Frame with each phrase as a row, along with other information such as review id, hotel name, hotel location, and related review rating. Then we discovered the cosine similarity metric for determining the aspect of each phrase by comparing similarity with the corpus of each aspect (obtained after completing aspect segmentation) and then assigning the aspect with the highest similarity. Using the python library, TextBlob we found out the sentiment score for each sentence.

We then created a dataframe with ReviewId, sentence, aspect, and sentiment score as columns. Finally, the classification model was trained using sentences as independent features and the aspect as the target variable. Accuracy and Confusion Matrix is the evaluation statistic we utilised to judge our model's performance. We will next proceed by selecting the best model based on accuracy.

6.4 *Retrieval and Ranking*

Our retrieval and ranking module works as follows: first, the user text query is taken and broken down into several components. Each of these component is assigned attributes with the assistance of the model that has already been trained on the dataframe. The dataframe was generated from the hotel dataset using aspect segmentation, where ẋŕepresented sentences from the reviews and ẏŕepresented aspects.

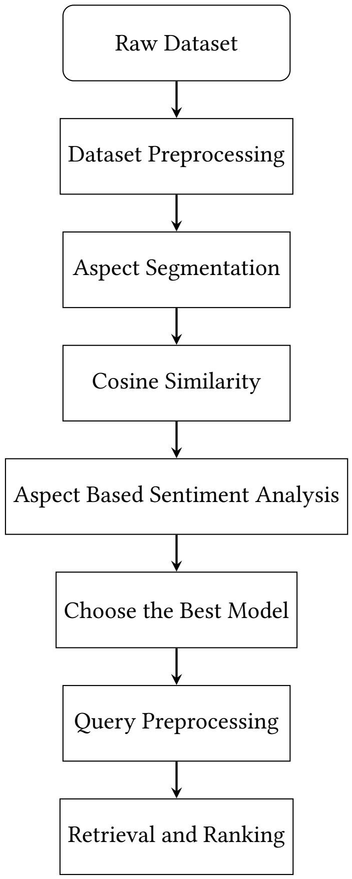

Figure 3: Flow Diagram

Following the assignment of an aspect to each component of the user query, the next step is to assign a sentiment score to each of these components. We now have several aspects with appropriate weightages for the entire user query. If an aspect appears more than once in a query, we will take the average of its scores and apply it to that aspect for that query.

Then we created another dataframe with records for only one hotel in each row. We then amalgamated all of the reviews for that hotel to create a single composite review for that hotel. The objective for doing so is to have an average rating for all aspects for a certain hotel in order to give the most accurate assessment of that hotel based on the reviews for that specific aspect.

That dataframe now contains five columns: name, location, merged review, dictionary containing all aspects and their respective weightage for that combined review, and total rating (weightage average of all aspects). Then it chooses all of the hotels with all of their aspect values greater than the values provided by the query, i.e. choose a hotel and any of its aspect values must be greater than the value of that same aspect in the query. Finally, the module ranks the retrieved hotels according to their overall rating, and this is our final list.

7 MODELS APPLIED

We used various machine learning models such as:Naive Bayes, Decision Tree, Logistic Regression, KNN, Random Forest and Support Vector Machine(SVM) on our preprocessed data.

The main issue in classifying texts is that they are a jumble of characters and words. We require a numerical representation of those words in order to input them into our K-NN algorithm, which computes distances and makes predictions. As a result, we used a numerical representation, which is a bag of words with tf-idf.

The overhead of computing distances for each data point whenever we wish to predict is extremely expensive. As a result, K-NN is ineffective for real-time prediction. When the data collection is tiny, K-NN should be preferred. You must also scale all of the features to normalised measure since we do not want the units of one feature to have a substantial influence over the units of another. In real-world data, Naive Bayes assumes conditional independence and attempts to estimate the optimal solution. As a result, Naive Bayes is a rather quick classifier. Even when we applied it on a smaller dataset, it did not provide us with good accuracy in our dataset. Because decision trees are greedy, grown deep, and unpruned, they have lower accuracy. As a result, each tree has a large variance (and thus a strong tendency to overfit) but a smaller bias. They are vulnerable to little changes in the data: a minor adjustment can result in a significantly different tree. As a result, throughout our baseline model implementations, we did not achieve satisfactory results with the decision tree. As a result, they were insufficient as a baseline model.

SVM Classifiers are supervised learning models with associated learning algorithms that evaluate data and recognise patterns used for classification. In a high or infinite-dimensional space, a support vector machine produces a hyper-plane or set of hyper-planes. The hyperplane with the greatest distance to the nearest training-data point of any class achieves a decent separation. In general, the wider the margin, the lower the classifier's generalisation error. Hence, our SVM gave us our highest accuracy at 88.7%.

8 FRONT-END UI

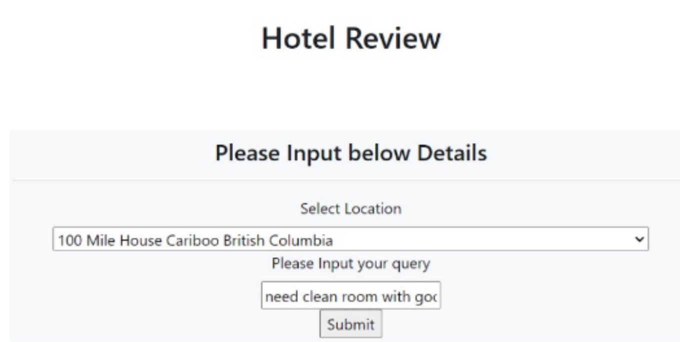

Figure 4: The front-end

Our project's front-end was built mostly with HTML, CSS, and Javascript. The Python Flask web framework was used to connect the front-end and back-end. The pickled model and data were loaded for speedy query processing as well as retrieval of hotels matching the queries. The dashboard may be accessed through *localhost:5000*, which corresponds to the form's GET request method. When a GET request is received, the user is prompted to provide their query and the location of their choosing before submitting.

The submit method is equivalent to the POST method, in which the user's query is retrieved from the form. After that, a function call is made to process the query and return the list of hotels with their respective calculated ratings. The results are then filtered based on the location chosen and ranked in descending order of derived ratings. Finally, the data is restructured and delivered to the front-end so that it may be displayed to the user.

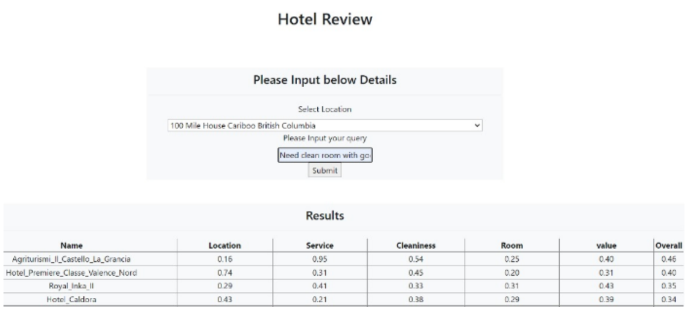

Figure 5: The results of the user query displayed

9 EVALUATION METRICS

The below table represents the results when these models were run on 500,000 reviews of our processed dataset.

Models	Accur.	Precision	Recall	F1
Naive Bayes	0.625	0.810	0.434	0.477
Decision Tree	0.655	0.643	0.582	0.607
Logistic Regression	0.701	0.865	0.577	0.655
KNN	0.552	0.604	0.407	0.444
Random Forest	0.578	0.870	0.368	0.4700
SVM	0.887	0.896	0.777	0.7977

Table 3: Models Applied

We are using Accuracy as our evaluation metric, and after applying various models on 500k reviews, we got maximum accuracy with Support Vector Machine at 88.7%.

10 CONTRIBUTIONS FROM MEMBERS

Below are the contributions by each team member:

Arnab Chatterjee: *Aspect Segmentation, Aspect Based Sentiment Analysis, Ranking and Retrieval*

Aurko Mitra: *Data Pre-processing, Models Evaluation, Implemented Classification models*

Lavanya Adhikari: *Data Pre-processing, Literature Survey, Implemented Classification models, Documentation of the Entire Project*

Shubham Sharma: *UI front end, Flask App, Connecting back end and front end, Deployment using Heroku*

Yash Goyal: *Aspect Segmentation, Cosine Similarity, Aspect Based Sentiment Analysis, Implemented Classification models, Ranking and Retrieval*

11 CONCLUSION

As a result, we were able to successfully construct the Text-Based Hotel Review Retrieval System. It is superior to other existing systems because we used aspect segmentation to assign aspects to each query statement entered by the user, and then we performed the final retrieval and ranking of hotels based on that.

REFERENCES

[1] Large-Scale (50M) Hotel Reviews Dataset. 2020. Dataset. http://lia.epfl.ch/Datasets/Full_HotelRec.zip.

[2] Wenjing Duan, Qing Cao, Yang Yu, and Stuart Levy. 2013. Mining online user-generated content: Using sentiment analysis technique to study Hotel Service Quality. *2013 46th Hawaii International Conference on System Sciences* (2013). https://doi.org/10.1109/hicss.2013.400

[3] Susanti Gojali and Masayu Leylia Khodra. 2016. Aspect based sentiment analysis for review rating prediction. *2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA)* (2016). https://doi.org/10.1109/icaicta.2016.7803110

[4] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '04* (2004). https://doi.org/10.1145/1014052.1014073

[5] Ioannis john Pavlopoulos. 2004. ASPECT BASED SENTIMENT ANALYSIS.

[6] Ivan Titov and Ryan McDonald. 2008. A Joint Model of Text and Aspect Ratings for Sentiment Summarization. In *Proceedings of ACL-08: HLT*. Association for Computational Linguistics, Columbus, Ohio, 308–316. https://aclanthology.org/P08-1036

[7] Thang Tran, Hung Ba, and Van-Nam Huynh. 2019. Measuring hotel review sentiment: An aspect-based sentiment analysis approach. *Lecture Notes in Computer Science* (2019), 393–405. https://doi.org/10.1007/978-3-030-14815-7_33

[8] Hongning Wang, Yue Lu, and Chengxiang Zhai. 2010. Latent aspect rating analysis on Review Text Data. *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '10* (2010). https://doi.org/10.1145/1835804.1835903

[9] Saman Zahid. 2020. Sentiment Analysis of Hotel Reviews - Performance Evaluation of Machine Learning Algorithms. (Mar 2020). https://doi.org/10.13140/RG.2.2.21026.96965