Tourist Attractions Recommendation and Information System

Indraayudh Talukdar
Antara Das
Pranshu Patel
Md. Taasir
Arun Sen
Amey Pawar

ABSTRACT

In this fast paced life of today, it is very difficult to find some place for an offbeat place for vacation. The reason being people do not have the necessary time to spare for this work. In this paper, we present an automated software addressing this need. We allow free text query as well as search by image. We maintain a number of indices to find the appropriate results from the free text query. Image feature extraction using deep learning is being done which will help in finding the architectural similarity of monuments or pilgrimages.

ACM Reference Format:
Indraayudh Talukdar, Antara Das, Pranshu Patel, Md. Taasir, Arun Sen, and Amey Pawar. 2018. Tourist Attractions Recommendation and Information System. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Searching for off-beat places for a vacation is time taking process and finding that time in this fast-paced life is a time taking process. In this paper, we come forth with a information retrieval system which allows a free text query. The query can allow the person to search for a particular category tourist spot like beach, forest, historical places and so on. Next we allow the tourist spots to be chosen according to states or cities as well and they can be paired with categories as well. If a person directly asks about a particular place, we will provide the near by places as well along with details of the queried place. It is very difficult to understand a user's exact need. Some might prefer to go to small peaceful virgin sea beach. Some might even want to take on historical places which have the same architectural features.

2 RELATED WORKS

In [1], a user's representation is made, and their neighbours are created based on their preferences and previously visited places. Cosine similarity is used for similarity calculation along with memory-based and model-based collaborative filtering.

In [2], user groups are created with KNN and based on the current situation, user group preference, and conveyance facilitating its reach in the present scenario, the recommendation is provided. Like our work they have also tried to give brief descriptions and images of the destinations, though such resources are gathered from external websites from the web, filtered and then ranked before presenting to the user. So these information system is dependent on the freely accessible tourist destination information and images available over the web, whereas we are trying to curate a decent dataset to provide the users with such essentials.

In [3], the user's travel and personal information is gathered, and a deep neural network is used to provide recommendations. Next, IoT devices are used to obtain real-time information, which modifies the recommendation.

In [4], recommendations for a user will be dependent upon his profile and past travel experiences. The places are recommended using an ensemble recommender based on collaborative filtering, content-based filtering and demographic filtering. Also, a dataset catering to the attributes and ratings of the places and user details is also created.

In [5], the user's travel photos or any other travel based or activity based photos are taken, which are considered as photo query, and then it is given to a CNN based pretrained image classifier for low resource devices called as EfficientNet-Lite to extract features from it and associate one of the 40 label tags to it. Then recommendation for tourist attractions is made by measuring cosine similarity between the vector formed from the user input image using probabilities and the feature vectors of tourist destination images of Indonesia.

3 METHODOLOGY

3.1 Dataset

Our dataset was prepared using Apify, a platform that enables developers to automate and scale web scraping, data extraction, and web automation tasks. Apify provides a range of tools for creating, executing, and monitoring scrapers and other automation workflows.

With Apify, we were able to set up web scraping jobs that ran on a regular schedule, extracting data from websites and APIs and storing it in a structured format such as JSON, CSV, or Excel. Specifically, we used the Google Maps Scraper within Apify to quickly extract data from hundreds of Google Maps locations, including reviews, images, opening hours, location, popular times, and more.

Our dataset includes features such as the name of the place, subtitle of the place, image URL of the place, latitude, longitude,

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

category of place, description of place, address of place, city of place, state of place, neighborhood, review tags associated with the place, and review score associated with the place.

We divided the tourist places into categories, including pilgrimage, beach/sea, hill station/mountain, wildlife/forest, adventure, historical place, museums, trekking, desert, and smart city.

We scraped tourist places across India, including 28 states and 8 union territories. Our dataset includes approximately 3,200 entries in total, providing a comprehensive overview of tourist places to visit in India.

3.2 Inverted Indices

In inverted indices part, we create the indices city-index, state-index, category-index, cross-index and near-by-place-index. City-index contains the row indices corresponding to a particular city. This is analogous for state-index and category-index. The near-by-place index will provide the near-by tourist spot of a particular tourist spot asked by the user in query. Here near-by is meant by tourist spots within 200km. The distance is measured using the latitude and longitude of the places present in our dataset. The cross-index will give us provision for coupling the categories and state or city. This is means, it will help in finding the tourist places of a particular category in a particular city or state.

3.3 Query Processing

For query processing, we first keep a check if the user is asking about a particular tourist spot. If its true, then details of that place and its near-by places will be shown. Now, if the person asks for a particular category of tourist spot like beach, pilgrimage or historical place and other categories like this. Next, if we have a city or a state, we show all the tourist spots in there. Now, if we have category and state, we make use of the cross-index. First segregation of the categories and places(state or city) is done in two separate lists and then cross-product of both the lists is done. The bigrams obtained in the cross-product is used to find all the tourist of particular category and a particular place. If we have more than one place or category, a union is obtained.

3.4 Image Feature Extractor

For the image feature extraction part here we present a baseline deep learning model with two convolutional neural net (CNN) layers. The first layer have 32 feature maps with (3,3) kernel. Next we have maxpooling layer followed by another CNN layer with 64 feature maps and (3,3) kernel. Then we have another maxpooling layer. After this, we flatten everything and feed the data to fully connected layer of 256 neurons. Then finally we have the output layer of 11 neurons. The philosophy behind this image feature extraction corresponds to architecture similarity of buildings or tree structure similarity in forests. This will help the user to find the historical places and pilgrimages with same architectures simply by uploading an image. The training set will be the images of places and ground truth output labels would be the categories of a place.

3.5 User Interface

We have used flask for integration because we are making a ML model and so integration of UI with the model becomes easy with FLask. We have created a main.py file where all the routing process In main.py file we have rendered a home page where the user will land first after entering our website. The user will be asked to enter a free text query in the search bar. After entering the query, the user will enter a results page where the all the names of places and images of places will be displayed to user.

In the main.py file we have used the libraries urllib.request, imghdr and base64. In the main.py file we have defined a routing to results page where first the entered query will be retrieved through request.form['query'] method. After that the query will be passed to the model which will give a dataframe as a output.

From the dataframe all the resultant urls will be stored in an array. After that all the urls stored in URLs array will be read one by one with the help of urllib.request and then the image url will be changed to base64 encoding with the help of base64 library and then it will be appended to another array. This array will be passed to the results.html page along with the dataframe for displaying the results. In the results.html page we have used html and css for designing our page.

4 RESULTS

The results will be mentioned in two parts. One the part which will be the training accuracy for the DL model and the other is the UI. In the accuracy for the DL model, we have traning accuracy of 0.4463.

Next we have the user interface presented here.

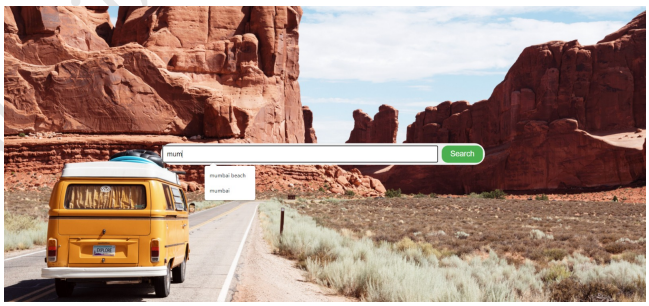

Figure 1: user Interface

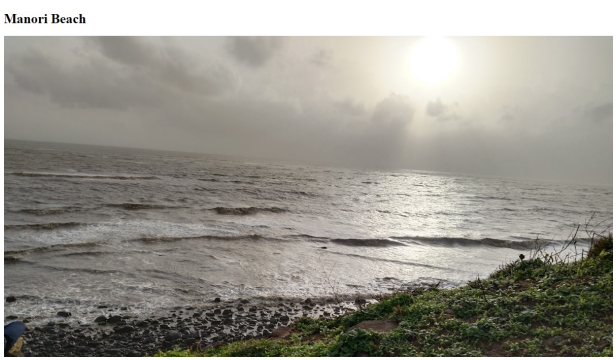

Figure 2: Output after a query

5 FUTURE WORKS

For future works we will include certain features about each place in our data to aid the free text query information retrieval. Next we would improve our DL model and extract the image features. The image features will be stored against each image. When the user will upload image, we will extract its features and measure similarity with our stored images. Then places will be ranked based on similarity.

6 REFERENCES

[1]Cepeda-Pacheco, J.C. and Domingo, M.C. 2022. Deep learning and Internet of Things for tourist attraction recommendations in smart cities. Neural Computing and Applications. 34, 10 (May 2022), 7691–7709. DOI:https://doi.org/10.1007/s00521-021-06872-0.

[2]Fudholi, D.H. et al. 2021. Deep Learning-based Mobile Tourism Recommender System. Scientific Journal of Informatics. 8, 1 (May 2021), 111–118. DOI:https://doi.org/10.15294/sji.v8i1.29262.

[3]Jia, Z. et al. 2015. User-Based Collaborative Filtering for Tourist Attraction Recommendations. 2015 IEEE International Conference on Computational Intelligence Communication Technology (Feb. 2015), 22–25.

[4]Kbaier, M.E.B.H. et al. 2017. A Personalized Hybrid Tourism Recommender System. 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA) (Oct. 2017), 244–250.

[5]Smirnov, A. et al. 2013. Recommendation system for tourist attraction information service. 14th Conference of Open Innovation Association FRUCT (Nov. 2013), 148–155.