Tourist Attractions Recommendation and Information System

Antara Das Amey Pawar Arun Sen
Indraayudh Talukdar Md. Taasir Pranshu Patel
Indraprastha Institute of Information Technology, Delhi

ABSTRACT

The project is a web application designed to provide users with a comprehensive view of different tourist places and their associated details according to their preferences. India is a huge and diverse country with enormous tourism potential. In this project, a multimodal search engine is designed where a user can choose to either give a text query or a photo query to see a collection of relevant destinations. To promote tourism in India and facilitate better travel planning, there is a need to publish a non-profit website that helps citizens and international travellers with information about a plethora of Indian destinations and also recommends places based on their interests.

ACM Reference Format:
Antara Das Amey Pawar Arun Sen, Indraayudh Talukdar Md. Taasir Pranshu Patel, and Indraprastha Institute of Information Technology, Delhi. 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, 4 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 an information retrieval system that provides users to search for an Indian destination using either a free text query or by uploading an image as a photo query. The text query can allow the person to search for a place based on place title or using a city name with or without the category of the spot, like a 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 nearby places as well, along with details of the queried place. A spell correction feature is added in our text search query part to get near to the user's want. Also, we use the descriptions and user reviews to form a feature set for a place which can be used to search for a place or places of particular characteristics. In the absence of particular matching of a place with our data, the texts are converted to an n-gram query, and we check for states, cities and categories of places that might meet the user's demands. It is very difficult to understand a user's exact needs. Some might prefer to go to a small peaceful virgin sea beach or a historical place which has a kind of architectural features. To get a better understanding of user query, an option to upload image is also provided. The query image would give aid in finding places with similar geographical features if it is any natural place or architectural features if it is any man-made structure. The main motive is to understand the user's needs and providing information according to it.

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

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.

If we consider the real-time applications or websites in the tourism industry, the existing websites fall short of providing personalized travel recommendations and general information services. They handle only a limited variety of queries and fail to show appropriate results when there is no city or state name in the text query.

Also, none of these existing works provides the flexibility to give both text query and photo query. We have tried to address many of these problems to produce a novel demanding solution.

3 METHODOLOGY

3.1 Dataset

Due to the lack of a comprehensive public dataset that includes information and pictures of numerous tourist destinations, we have built a new dataset for this project.

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, 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, near-by-place-index and 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. The place-index contains the row indices for each place titles.

3.3 Query Processing

For query processing, we first keep a check if the user is asking about a particular tourist spot. If the query exactly or partially matches with any of the place name in the dataset, with a similarity score more than 0.5, then details of those places will be returned promptly. Now, if the person asks for a particular category of tourist spot like a 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 N-grams 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. To filter out the output content the descriptive words in the text query are utilized to rank the results based on their relevance with the input query and popularity score and top 100 results are returned as final output. In fig.1 the pipeline of the text query processing is elaborated.

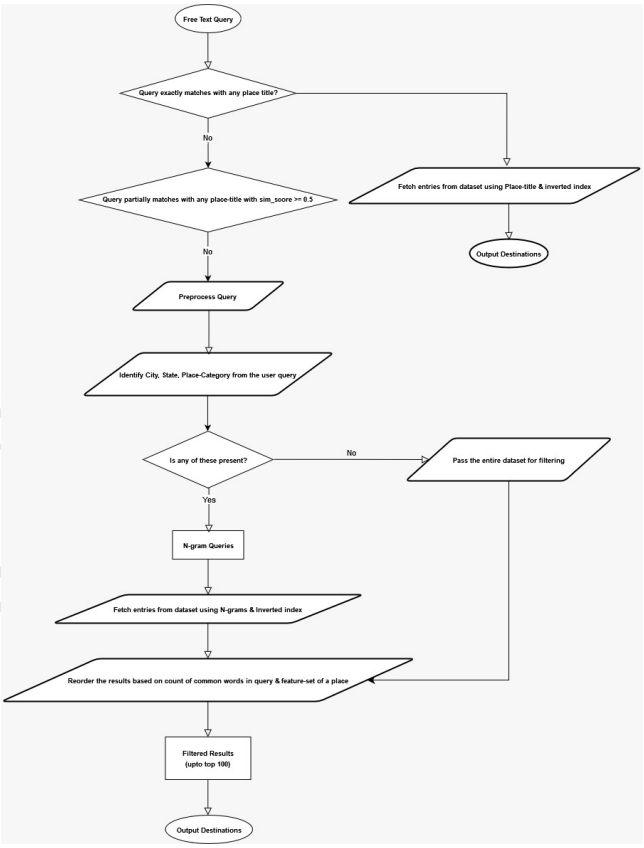

Figure 1: Text query processing pipeline

3.4 Image Feature Extractor

For the image feature extraction part, we first tried out training a CNN model with images of places and the categories of the images as ground truth output labels. After training the model with 45k images, the features extracted at the last layer for each image are captured and stored for future reference. At the runtime, when a user uploads a new image, its features are predicted using the model and then comparing it with the existing feature set, a maximum of 50 ranked images are returned as output.

Due to lack of accurate label annotation, this model did not work satisfactorily. Also, there is a need to multi-label annotation as some places may fall under multiple labels simultaneously, like Hill Station, Forest, Tourist Attraction etc. To improve the photo

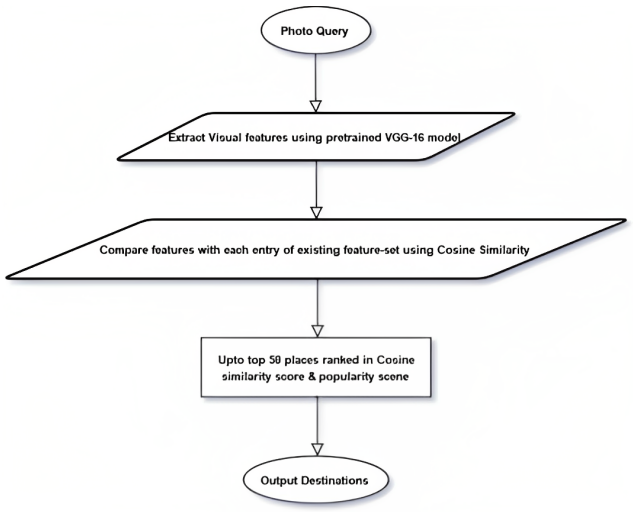

Figure 2: Image Feature Extraction Pipeline

matching efficacy next, we have used a pre-trained CNN model called VGG-16 to extract the features of the train and test images. In fig. 2 the pipeline of the text query processing is elaborated.

The philosophy behind this image feature extraction corresponds to capturing the visual or architectonic similarity of places. This will help the users to find places they are looking for simply by uploading an image.

3.5 User Interface

The Interface is built using React for the front-end and Flask for the back-end.

The search-by-text-query option allows users to enter keywords related to the place they are interested in, and the application returns a list of relevant images along with their descriptions and location. Clicking on any of the images from the list displays their detailed description, review tags, review rating, and a list of nearby places.

The search-by-image option is particularly useful when users do not have a specific name or keyword to search for or the user prefers to find a place visually similar to the query photo. By uploading an image, the application uses its image recognition technology to provide similar images to the inputted image, helping users to find related places.

Overall, the user interface is highly user-friendly and intuitive, providing users with easy access to the information they need.

4 RESULTS

Next we have the user interface presented here.

5 FUTURE WORKS

The future scope of this work expands in the direction of building a NER model to automatically identify city, and state from text query and few other techniques to optimize the text query processing time. For the image processing model we can train a new CNN-based model or fine-tune a pre-trained model on our image collections for better visual feature extraction. For this, it is required to more accurately annotate the category labels of the pictures. For result evaluation, a user-based survey can be done to measure the degree of relevance of the results and the ranking efficacy in terms of NDCG.

Figure 3: home page

Figure 4: Home page with query

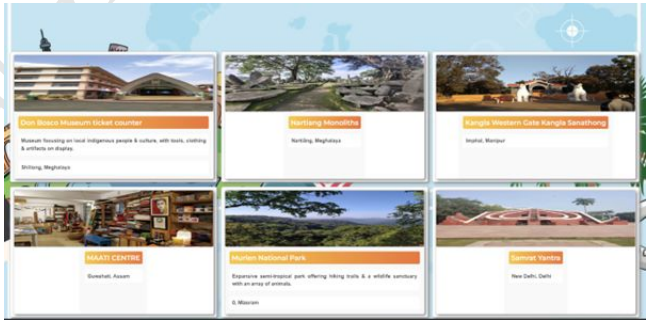

Figure 5: Results for the input text query

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.

Figure 6: Near by places

Figure 7: query image

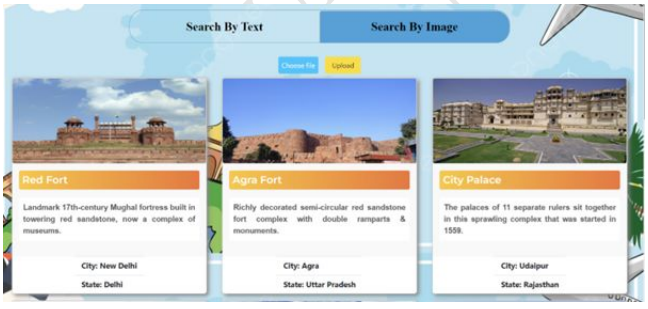

Figure 8: query image output

[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.

[6] https://apify.com/apify/google-maps-scraper

[7] https://apify.com