CUSTOMISED ITINERARY CREATION USING NEURAL NETWORK

Dr.(Mrs.) Nupur Giri , Pooja Bhatia, Ritika Mendjoge ,Siddham Sharma,Ajitesh Singh

HOD(Computer Science Department ,VESIT), Student(B.E, Computer Science Department ,VESIT), Student(B.E, Computer Science Department ,VESIT), Student(B.E, Computer Science Department ,VESIT), Student(B.E, Computer Science Department ,VESIT)

Computer Science,
VESIT, Mumbai, India

Abstract : A neural network based system is designed to create at-go itineraries based on interests of the traveller. Data is collected by scrapping various websites. After pre-processing, the data is labelled. A neural network is used to predict POIs other than those given by the user. The suggested POIs are knit in an itinerary using time dependent orientation approach. An Android Application is used to make the system deliverable.

I. INTRODUCTION

Planning travel itineraries have always been tedious and often time consuming. Lack of knowledge of the routes and place of interest result in excessive investment of time in research by the traveller. Since too many specifications are involved in the process which vary person to person, existing applications act as tools to rather simplify the process, however, they do not make the entire process of travel itinerary planning hassle free.

Within past few years, tourism all over the world has become more diversified and personalised. The millennial generation is responsible for this change in trend. With the advent of mobile technology and increase in the millennial population, demand for travel planning simplicity is evident. As of now, the existing applications provide information necessary to plan a vacation (such as restaurants, hotels, flights) or information pertaining to places to visit in the vicinity. However, none of them combine all the above features into a personalised travel plan. A system is proposed to make the process of itinerary planning automated by integrating AI (Artificial Intelligence) into a user friendly platform.

II. LITERATURE SURVEY

To determine the most suitable suggestions, most implementations depend on memory based methods [1] as well as collaborative filtering [2]. Nonetheless, the provided solutions individually fail to provide effective solutions for itinerary recommendation.

[3] is one of the first attempts in tourism which involves creating a mobile application to model the tour guide activity. The paper accepts the importance of context, but restricted to location awareness.

The guide system presented in [4] provides recommendations based on the user's profile. The drawback of the approach is the amount of information needed from the user in order to customize the trip.

Another idea already proposed revolves around selection of POIs and Mapreduce technique. [5] has proposed a system which creates customized travel itineraries automatically based on the user's preferences by combining the single day itineraries into multi day itineraries.

The user is required to provide a list of POIs that he intends to visit. All possible single day itineraries are generated and stored in a DFS system. These single day itineraries are constructed with respect to the POI. Itinerary indexing technique is used to store and map the one day itinerary using key value pair strategy. Due to the popularity of certain POIs, the index file will contain large number of single day itineraries containing those POIs. This might lead to memory overflow in the sorting process

Itineraries consisting POI will be split into n groups and each group can be sorted in memory. By scanning we can get a sorted list for all itineraries involving a POI.

The partial results of the possible itineraries are extremely large. Mapreduce is the solution to partition the partial results and generate parallel itineraries. Parallel computing provided by the Mapreduce reduces the runtime of the pre-processing and the Mapreduce algorithms can remove the duplicated itineraries in a simple way. After the construction of itineraries, m best itineraries are selected from the indexes according to the user request. POIs are selected by their weights. Then, in each iteration, a group is formed, which contains a subset of POIs that can be accessed within a day. There are maximally k groups generated. All groups are used as reference for searching the index.

New solutions are searched and are used to replace the greedy itineraries. A threshold for the maximal number of adjustments is set. In each iteration, independent sets for the existing itineraries are found. Benefit can be checked by performing replacements. If the benefit is larger than 0, the resulting itineraries are updated as the new ones. Hill climbing algorithm is used in order to avoid suboptimal solution.

III. METHODOLOGY

In the proposed methodology, the user is required to provide certain information which is essential to build an itinerary. Also, the generated itinerary can be dynamically altered as and when the user wants.

Creation of itinerary is mainly a two-step process.
a. Pre-processing
b. User side application

a. Pre-processing:
In pre-processing stage, information is scrapped from various blogs, since they provide intricate details of the user's experience. Scraping is done using selenium package in python. Then the relevant information was extracted from the scraped data using Natural language Processing and stored in MongoDB database. A hybrid recommendation system model was built using neural networks. The recommender gives the most likely places the user would like to visit. A metaheuristic time dependent orienting problem with time windows is used to generate the complete start to end route i.e. complete itinerary. Sub-steps of pre-processing are mentioned in fig. 3.1 and are described below.

i. Web Scraping:

The first step is getting paragraphs of data from different travel blogs. In scrapping we basically first try to get the external and internal links inside the blog that we are trying to scrape as shown in Fig. 3.2. The scraper keeps adding the links in a buffer as long as it finds new links. Simultaneously the scraper visits each links one by one and stores the data of each locations in a *csv* format.

ii. Information extraction using NLP:

Pattern matching is done based on predefined regular expression. The commonly used information extraction is named entity recognition (NER). NER classifies the text into predefined categories. Name of person, locations, monetary values, quantities, can be extracted by NER technique easily with good accuracy. The important information can be extracted by part of speech tagging. NLTK package in python is a very popular library used for NLP tasks. NLTK package in python is pre trained but can be trained again for a specific blog articles. Thus using NLTK package and text mining we get detailed information of POIs in structured format.

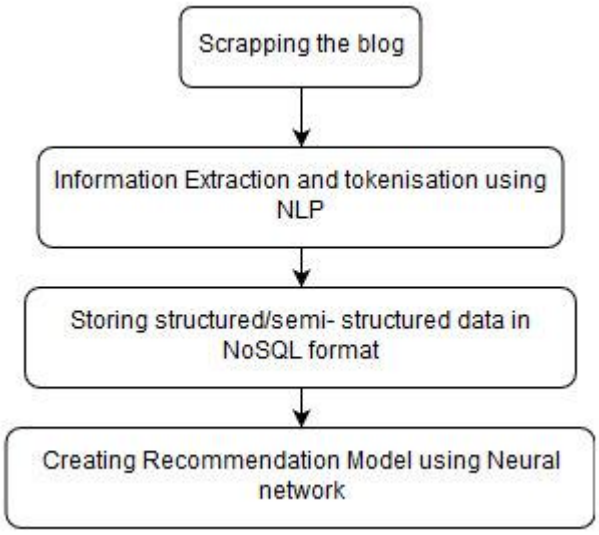

Fig. 3.1

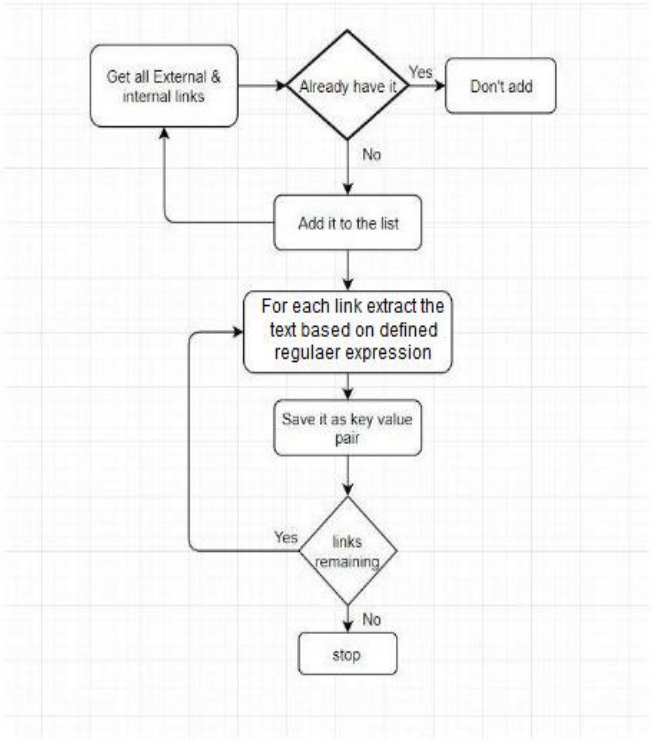

Fig 3.2

iii. Storing structured data in NoSQL format:

NoSQL i.e. “not only SQL” stores data in key-value pairs and provides storage and retrieval other than relational databases. Here we use nested key-value pairs to store entire information of a particular location. Fig 3.3 shows the complete overview of the database structure of one Location.

iv. Creating recommendation model using neural network:

A hybrid recommendation system is built which takes parameters such as place to visit, no of days, interest, type of traveller budget, no of people from the user and suggests the POIs that the itinerary should include. For training the neural network, back propagation algorithm is used. Training dataset used consisted of itineraries already followed by popular travellers scraped from the tourism websites. Implementation of the neural networks is done using keras in python (high level neural network API). All the categorical values are converted into numbers representing respective categories using LabelEncoder. Since neural network works only with binary values, categorical values are converted into binary values using One Hot Encoding. Relu activation function is used for all the layers except the output layer. For the output layer Softmax activation function is used. K-Fold cross validation was used to get high accuracy of the model. Fig 3.4 depicts a general representation of ANN (Artificial Neural Network).

```
{
  "Goa": {
    "Location_1": {
      "Things to do": "..",
      "Food": "..",
      "Hotels": "..",
      "Minimum time" : "..",
      "day/night": "..",
      "Type of place": "..",
      "Type of traveller": "..",
      "Rating": "..",
      "Cost": ".."
    },
    "Location_2":{
      ..
    },

  "budget_1": {
    "Amount": "..",
    "Location_involved": ["Location_1", "Location_2",".."],
  },
  "budget_2":{
    ..
  }

}
```

Fig. 3.3

Steps for K-Fold cross validation:

1 .The entire dataset is divided into k equal subsets, each subset is also called as fold. Let folds be f1, f2,

2. For i = 1 to k:

Fold f(i) is kept as the testing dataset and the rest k-1 subsets are used as training dataset. Train the model using cross validation k-1 subset of training data and check for accuracy

3. The final accuracy is the average of all the k cases of cross validation accuracies.

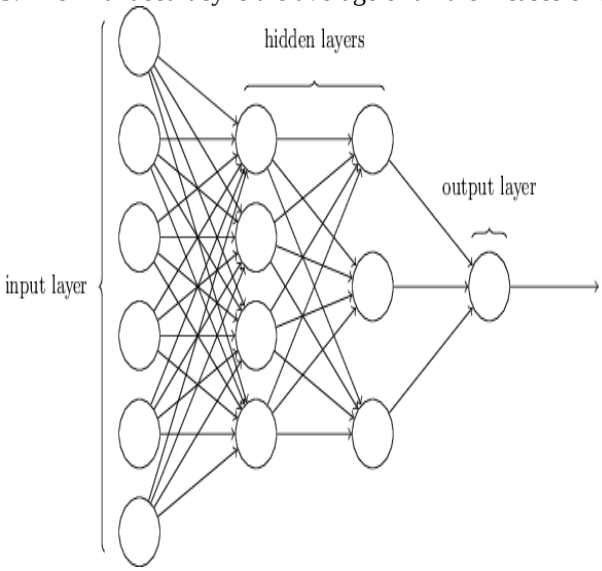

Fig 3.4

b. User side application

Flow of the application is

- User selects a location to visit
- Inputs like Place to visit, no of days, interest, type of traveller Budget, no of people is asked from the user.
- The user is even given an option to select the POIs he/she wants to include in the itinerary.
- This input data is pre-processed and fed into the neural network model.
- Neural network model gives the predicted POIs the user could visit.
- Priority is given to the POIs selected by the user and if time permits more POIs is selected from the recommender system.
- The final route is decided by the metaheuristic time dependent algorithm with time windows.
- After the itinerary is ready, the user has an option to dynamically change the itinerary at any time, even in between the trip.
- In the end user feedback is recorded and all the discrepancies are recorded to improve the recommender system model.

Creating the android application:

Once the data is structured it becomes easy to fetch the data quickly according to the user's input.
We integrate Google maps API, for live navigation and dynamically changing of itinerary. The programming languages used to create the application were java, objective c, XML and NodeJS for backend. We used volley from sending and receiving requests in JSON format from the android application.

Feedback system

A feedback system is used for improving the correctness of the itineraries generated. Users following the itineraries are constantly tracked to see the difference in the followed path and the itineraries. Even user feedback is asked after the completion of the trip. This data is recorded and used to train the neural network. Fig 3.5 shows the flow of the feedback system.

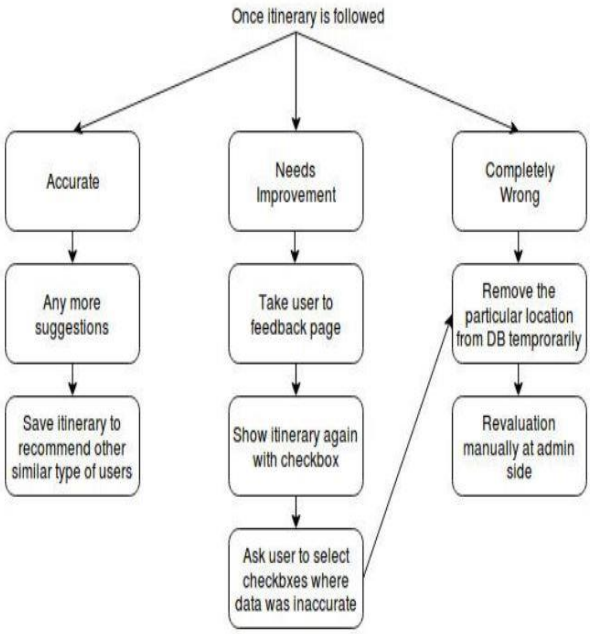

Fig 3.5

Time dependent orientation problem with time window:

Deciding the sequence of the POIs to be visited is also very important task to maximize the number of POIs the user visits. For this Time dependent orientation problem with time windows is used. Time dependent orientation problem with time windows creates a route for a single day where the start and the end POI is the hotel the user is residing in. Such single day itineraries are combined to give the complete start to end itinerary.

Google maps API:

The itinerary is displayed using Google maps API and the user is dynamically tracked if he follows the itinerary using the application. Google distance matrix API gives the distance between nodes and also the time it would take to reach from node-1 to node-2 including the traffic time. So the itineraries generated are pragmatic

IV. CONCLUSION

As compared to previous methods, the proposed method is pragmatic as well as scalable. This novel method adds a new perspective to consideration of various parameters to generate a self-learning system for personalised usages.

V. REFERENCES

[1] Greg Linden, Brent Smith, and Jeremy York, "Memory based recommender system", Amazon.com Recommendations Item-to-Item Collaborative Filtering, January 11, 2010.

[2] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl ," Item-Based Collaborative Filtering Recommendation Algorithms", GroupLens Research Group/Army HPC Research Center Department of Computer Science and Engineering University of Minnesota, Minneapolis, MN 55455

[3] G. Abowd, C. Atkeson, J. Hong, S. Long, R. Kooper, M. Pikerton,"Cyberguide: A mobile context-aware tour guide," Wireless Networks, vol. 3, pp. 421-433, 1997.

[4] K. Cheverst, N. Davies, and K. Mitchell, "The role of adaptive hypermedia in a context-aware tourist guide," Communications of the ACM: Special Issue on Adaptive Web-Based Systems and Adaptive Hypermedia, No. 45, 2002, pp. 47-51

[5] Automatic Itinerary Planning For Traveling Services
Gang Chen, Sai Wu, Jingbo Zhou, and Anthony K.H. Tung, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 26, NO. 3, MARCH 2014

[6] Renita Crystal Pereira, Vanitha T, "Web Scraping of Social Networks", International Journal of innovation Research in Computer and Communication Engg, Vol3,Spl issue 7,Oct 2015.

[7] Antonio Moreno, Teófilo Redondo, "Text Analytics: the convergence of Big Data and Artificial Intelligence", International Journal of Interactive Multimedia and Artificial Intelligence, Vol. 3, Nº6,2016.

[8] Derrick H. Nguyen and Bernard Widrow, "Neural Network For Self Learning Control System". August 6, 2002.

[9] Abbaspour, A. and Samadzadegan, F. Time-dependent personal tour planning and scheduling in metropolises In Expert Systems with Applications 38:12439-12452 2011 Donati, A. and Montemanni, R. and Casagrande, N. and Rizzoli, A. and Gambardella, L.

[10] W. Souffriau, P. Vansteenwegen, G.V. Berghe, and D.V. Oudheusden, "A Path Relinking Approach for the Team Orienteering Problem," Computers and Operations Research, vol. 37, pp. 1853-1859, 2010.