sustainability

MDPI

Article

A Context-Aware Location Recommendation System for Tourists Using Hierarchical LSTM Model

Wafa Shafqat and Yung-Cheol Byun *

Department of Computer Engineering, Jeju National University, Jeju 63243, Korea; wafashafqat92@gmail.com
* Correspondence: ycb@jejunu.ac.kr

Received: 12 April 2020; Accepted: 13 May 2020; Published: 18 May 2020

Abstract: The significance of contextual data has been recognized by analysts and specialists in numerous disciplines such as customization, data recovery, ubiquitous and versatile processing, information mining, and management. While a generous research has just been performed in the zone of recommender frameworks, by far most of the existing approaches center on prescribing the most relevant items to customers. It usually neglects extra-contextual information, for example time, area, climate or the popularity of different locations. Therefore, we proposed a deep long-short term memory (LSTM) based context-enriched hierarchical model. This proposed model had two levels of hierarchy and each level comprised of a deep LSTM network. In each level, the task of the LSTM was different. At the first level, LSTM learned from user travel history and predicted the next location probabilities. A contextual learning unit was active between these two levels. This unit extracted maximum possible contexts related to a location, the user and its environment such as weather, climate and risks. This unit also estimated other effective parameters such as the popularity of a location. To avoid feature congestion, XGBoost was used to rank feature importance. The features with no importance were discarded. At the second level, another LSTM framework was used to learn these contextual features embedded with location probabilities and resulted into top ranked places. The performance of the proposed approach was elevated with an accuracy of 97.2%, followed by gated recurrent unit (GRU) (96.4%) and then Bidirectional LSTM (94.2%). We also performed experiments to find the optimal size of travel history for effective recommendations.

Keywords: context aware; recommendation system; LSTM; feature importance; XGBoost; tourism

1. Introduction

The escalating variety of available data, products, applications and services brings colossal choices for the consumers. Over the past decade, Internet applications and online services are booming and have bombarded the users with enormous quantities of information. Therefore, from the pool of choices, the selection of a desirable item sometimes becomes a hideous task. Recommendation systems (RS) have emerged to address these challenges. These systems are altering our thinking process, aiding us to come closer to precisely what we need. In spite of the fact that varied abilities and solid advantages of such systems are already leveraged by various industries, there is still an unexploited potential of recommendation systems in every domain primarily medicine, tourism and online shopping. In this manner, such applications keep on increasing an expanding significance from a monetary point of view.

The travel industry is an area with a gigantic potential for recommendation frameworks to assist travelers with reducing the multifaceted nature of arranging and choosing. Arranging a trip includes looking for a lot of factors that are interconnected (e.g., transportation, dwelling, attractions) with constrained accessibility and where relevant angles may have a significant effect (e.g., spatiotemporal setting). To advance and adjust to the requests of real society, numerous ideal models were evolved to bring much more ideal models and hybrid approaches. All the proposed approaches have their own pros and cons. The tourist recommendation systems generally take user travel history and preferences to predict the next location to be visited. For any recommendation system, it is very important to focus on history data and other relevant contextual features to produce effective results [1]. Contextual knowledge in the recommendation systems develops the user's satisfaction. However, the identification of the most relevant contextual features is equally challenging as flooding the recommendation system with all possible contextual features may lead to a depleted performance and dimensionality issues. Therefore, in this paper we proposed a hierarchal deep learning-based approach that considers both the trajectory data and key contextual features.

Our contextual investigation targeted the travel industry in Jeju Island, South Korea. Over the previous decade, the island has transformed into a core of Korea's travel industry with numerous lavish inns and resorts, on account of an expanding number of guests from abroad and local areas. Both foreign and domestic tourists are very crucial as a primary source of income for the Jeju tourism industry. Apart from foreign tourists, domestic tourists have also increased in recent years. The graphs below in Figures 1 and 2 present the total number of visitors per month to the top attractions in Jeju from 2018 to 2019.

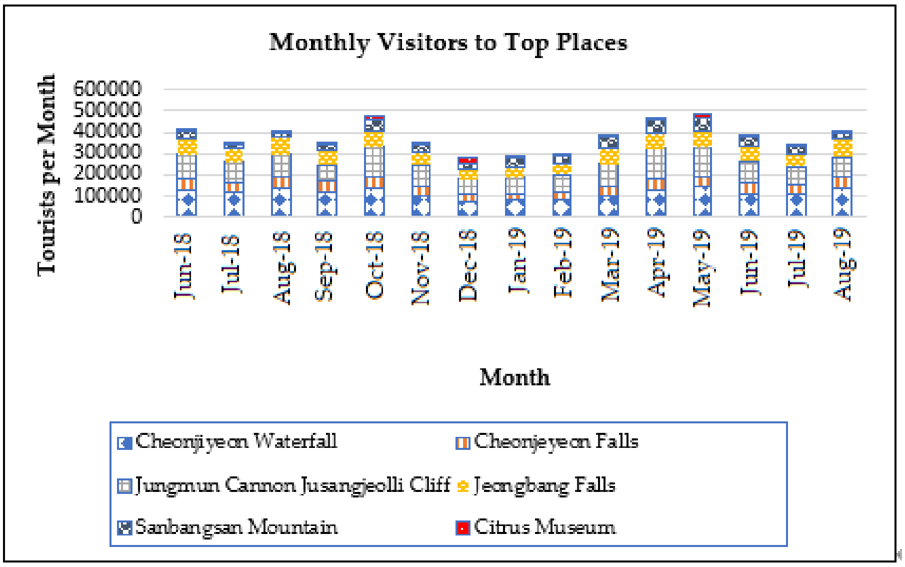


Figure 1. Statistics of tourists (2018–2019) on a monthly basis in popular places in Jeju [2].

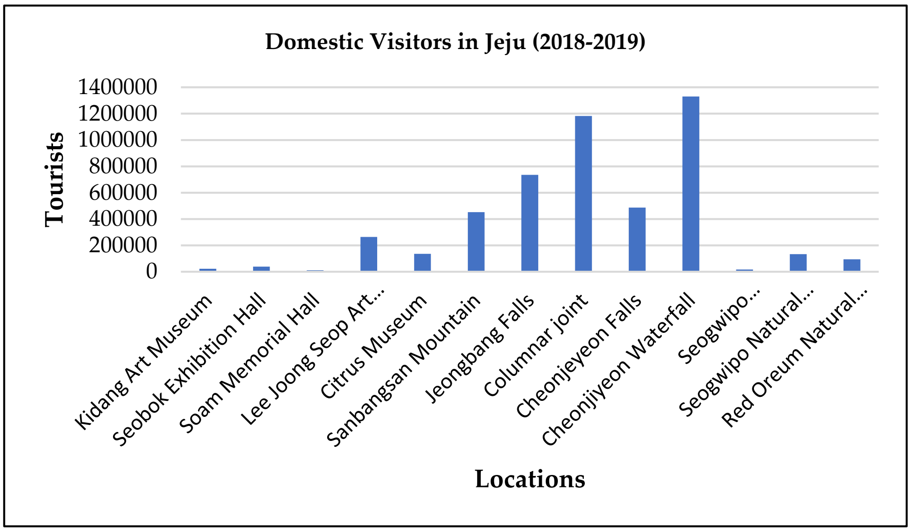


Figure 2. Statistics of domestic tourists (2018–2019) on a monthly basis in popular places in Jeju [2].

In 2019, there was an increase of 2.5% to 12.41 million domestic tourists in Jeju [3]. The figure above is a bar graph representing the number of domestic tourists to the top places in the year 2019. However, the recent coronavirus pandemic has also affected the tourism industry as it threatens a travel ban for many foreigners. This travel ban has resulted in a slight increase in domestic voyagers that brings a gleam of hope [4]. This global pandemic is now an important contextual factor as it has impacted the decisions of multiple stakeholders such as travelers, tourism management and the transportation industry. One such example is of AirBusan, a well known Korean domestic airline, which has declared a temporary increase in flights due to a recent increase in domestic travel requests [5]. Therefore, for any recommendation system it is important to take care of both the traveler's needs and the changing environmental context.

In this paper, we proposed a deep learning-based hierarchal model that captures temporal dependencies in the data by using preprocessed trajectory sequences. These sequences are converted to location embedding that is used as an input to a double layered long-short term memory (LSTM) model. This layer legitimately seizes the mobility patterns from user trajectories. Therefore, based on each user's behavior and decisions, prediction probabilities for the next location are produced.

The second layer of deep learning (also known as the context-enriched layer) is used to handle significant contextual data. These contextual data are of multiple types and comprise of different features. We aimed to explore the significance of different contextual features for a recommendation system and find the precise contexts that are actually responsible for accurate recommendations. Unlike the traditional systems that overload recommendation applications with all kinds of data, we aimed to find the best suitable and most relevant contexts to avoid the dimensionality issues. Our proposed system considers five primary categories of contextual features, namely location popularity, time, atmosphere, environment and distance. Location, weather and time are commonly used contexts in tourist recommendation systems. However various aspects have been generally overlooked such as distance, popularity and infectious risks due to atmosphere. These factors are rarely considered as contexts specifically in an implicit way as we proposed here. All the contexts have their own weightage according to the situation and the user's preferences. The weightages of these contextual features are prone to evolve or change over time. Therefore, we estimated feature importance to figure out what features have no impact on the recommendations. Those features were discarded later to avoid the overburdening of the data on the system. Finally, an objective function was formulated to maximize the weights of the desirable features and minimize the weights of the undesirable factors. This will result in a ranked list of places that users can choose to visit next.

In the rest of this paper, we first perform the literature review related to this work. The next section explains the proposed methodology in detail. Subsequently, the experimental details and results are presented. The paper is concluded with some discussions and concluding remarks.

2. Related Works

In this section, we present the literature review of the related works. We divided the related works into three sub-sections. In Section 2.1, we present the related works on recommendation systems; Section 2.2 presents the research related to deep learning in recommendation systems and Section 2.3 briefly presents work done on context utilization in recommendation systems.

2.1. Recommendation Systems

There are basically three types of recommendation systems [6,7]. These types are named as the collaborative filtering based recommendation systems, content-based recommendation systems [8–10] and the hybrid systems of both types.

The travel industry wave is indeed emerging and developing at an accelerated rate. Henceforth, the significance of point of interest (POI) recommendation systems has generated a considerable amount of attention. In [11], a traditional collaborative filtering based approach was being used to build a tourist spot recommendation system. Similarly, in [12], a case-based recommendation system was proposed for tourists. User preferences were calculated by using Bayesian networks in order to enhance the prediction performance for the next location to be visited by the users [13]. A multi-standard collaborative filtering based recommendation system was proposed in [14] for tourist spots. In this study, the authors used a Gaussian mixture model to enhance the accuracy of the results. A location-based recommendation framework was introduced in [15]. Similarly, in [16], a context-aware recommendation system for tourists was introduced. A hierarchal sampling-based hybrid recommendation system was proposed in [17].

In [18], images of the tourist location were used to overpower the challenges faced during the recommendation process such as that of data sparseness. Here, the authors first catered the cross-modal semantic dependencies in various image features by a Bayesian personalized ranking algorithm. A hierarchical sampling model was used to acquire the preferences of the users. In the end, the hybrid results were generated based on the previous steps. This proposed approach proved to be effective for group recommendations.

2.2. Deep Learning-Based Recommendation Systems

The accomplishments of deep neural network (DNN) models are extremely impressive in different fields such as discourse acknowledgment, PC vision and normal language preparing (for example [19–21]). Therefore, there are various methodologies proposed that exploit DNN models for recommendation systems [22–28]. In [22,24], the authors proposed multi-layer perceptron (MLP) models to catch the intricate structure of the communications between a user and an item. Such MLP-based models display a certain flexibility in their capacity to catch the client's unpredictable structure, utilizing a DNN-engineering and a non-direct capacity. Therefore, in [26–28], recurrent neural networks (RNNs) were exploited to display the consecutive request of user feedbacks.

Because of the intricate and overpowering parameters of DNN models, such DNN-based collaborative filtering (CF) approaches are inclined to overfitting. A few experimental investigations [23,29,30] have shown that the utilization of summed up refining strategies, for example dropout and regularization, similarly to pooling methods, can lighten the overfitting issues intrinsic to DNN-based models. Be that as it may, while the past endeavors referenced previously primarily center around how to abuse DNN models to upgrade the nature of proposals, not many endeavors have concentrated on the best way to stretch out such DNN models to address specific difficulties in suggestion frameworks. The customary RNN engineering is currently stretched to fuse the relevant data for context-aware recommendation systems.

2.3. Context-Based Recommendation Systems

The most commonly and widely used techniques for recommendation systems primarily include collaborative filtering (CF) procedures, matrix factorization (MF) [29] and Bayesian personalized ranking (BPR) [30]. These approaches usually make assumptions that the users who visit the same location have the same preferences and behaviors. Subsequently, they are most likely to visit the same place. The previous studies have shown that contextual information also plays a vital role in effective recommendation systems such as location, time and weather. [31–36].

In [37], a context-aware system for personalized citations recommendation was proposed. The system was based on a deep LSTM model and its results showed an improvement over previously proposed context-aware citation recommendation system. Trajectory prediction is also a very popular problem. In [38], a context-aware LSTM-based model was proposed to predict the movement of a human in crowded places e.g., a shopping mall or a museum. The experiments were performed on a publicly available pedestrian dataset and the results showed a better performance over state-of-the-art prediction models. In human activity recognition, context-aware models are proving to be an effective model. In [39], a 3D action recognition model was proposed that was built on a context-aware LSTM-based system. This model, with the help of contextual data, actively focused on the informative links in the sequence of actions.

3. Data

In this section, we present the distinctive specificities of the data utilized for the experiments. It is very important and a preliminary task to have a reasonable amount of data that can reflect and help discover useful patterns. Therefore, we spent a lot of time on data collection and preparation. As we were aiming at tourism in Jeju Island for the experiments, our data-crawling process was adapted accordingly. We mainly composed various travel blogs data, reviews against different locations and their corresponding star ratings, location coordinates, weather of a particular location, and user-generated data such as likes and shares. The description of each data source is given below.

3.1. Travel History Data

We had users' travel history data which included the previously travelled locations. These data were then processed to discover unique locations and then all the data were encoded into unique location IDs.

3.2. User-Generated Data

We collected online user-generated data in the form of reviews, ratings, the number of times a user searched for a location, the number of people that added a particular location to their wish lists, the number of times users have liked a place, etc. The features were mostly collected from TripAdvisor, VisitJeju and Google Recommendations.

3.3. Location Data

Besides user-generated data, we also observed each location's weather, humidity, distance, atmosphere risks, etc. The weather data were collected through OpenWeatherAPI [2].

4. Proposed Context-Aware Deep Learning-Based Approach for Tourist Spots Recommendations

This section is divided into subsections. Firstly, a brief overview of the system is presented. Then, a conceptual view of the system is explained followed by a complete architectural diagram and explanation.

The following subsections present the breakdown view of the overall architecture and each subsection broadly explains the working of a specific module.

4.1. Conceptual View

The proposed framework is a recommendation framework that joins different natural factors influencing the travel industry to evolve through the progression of procedures and give increasingly reasonable recommendations. The proposed framework comprised of information assortment, pre-handling, assessment learning and testing, objective functions and recommendations. Finally, the system generated the top n recommendations to the tourist after assessing all the relevant factors.

As shown in Figure 3, the conceptual framework of the proposed system enables tourists to search a query of their interest. There were two primary databases that separately handled tourist data and location data, respectively.

The data were prepared and used in the learning phase that used different machine-learning models. Firstly, the visiting probabilities for the next location were generated based on the user's previous travel patterns. Then, different contextual details were combined with the probabilities.

Both the recommendation model and context learning models are based on long-short term memory (LSTM) units. Finally, the system learns and recommends the top n places to visit based on the highest scores.

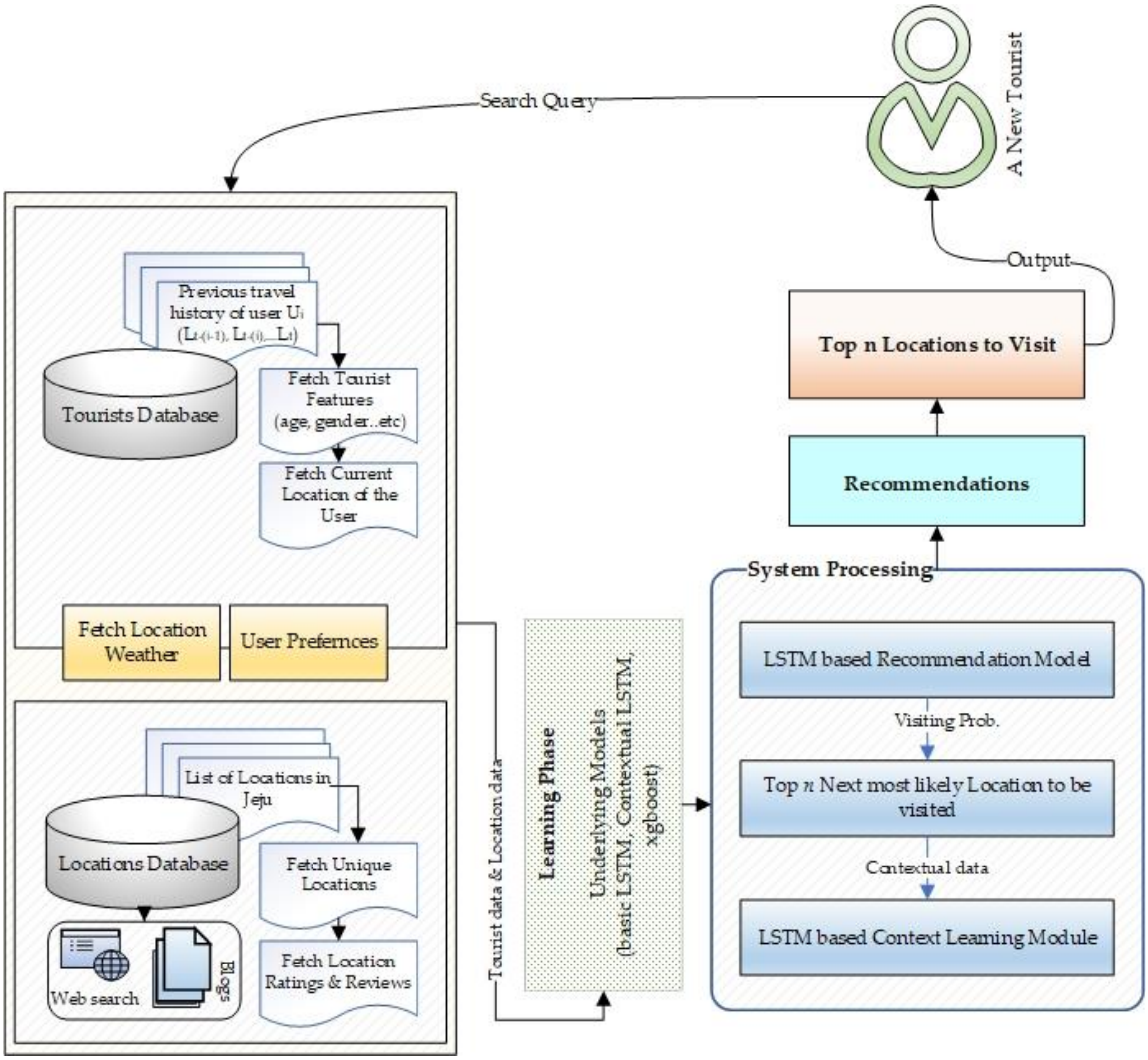

Search Query
A New Tourist
Output
Top n Locations to Visit
Recommendations
System Processing
LSTM based Recommendation Model
Visiting Prob.
Top n Next most likely Location to be visited
Contextual data
LSTM based Context Learning Module
Learning Phase
Underlying Models (basic LSTM, Contextual LSTM, xgboost)
Tourist data & Location data
Previous travel history of user Ui (Lt-(i-1), Lt-(i),...Lt)
Tourists Database
Fetch Tourist Features (age, gender..etc)
Fetch Current Location of the User
Fetch Location Weather
User Prefernces
List of Locations in Jeju
Locations Database
Fetch Unique Locations
Fetch Location Ratings & Reviews
Web search
Blogs

Figure 3. Conceptual view of the proposed model. (LSTM: long-short term memory).

4.2. System Architecture

A detailed view of the whole procedure is shown in Figure 4. The data were collected from multiple sources. First of all, a list of visitor areas in Jeju was organized. All these locations were searched on different tourist websites to collect relevant data. The most primary sources were TripAdvisor, Google and VisitJeju. We collected ratings, reviews, the total number of social network shares, wish list size and the number of times each location had being searched. For the traveler's data, apart from his preferences, we collected his previous travel history and current location. The weather data including temperature, wind and other weather conditions were also fetched for each location. As the data were being collected from multiple sources and their structures were very different, there were therefore multiple preprocessing units implemented to handle all the types of data. The preprocessing unit I was to process the review text data and the preprocessing unit II took all the travelers' data, primarily their travel history, and processed them. It was responsible for cleaning the data, fetching unique locations and assigning unique codes to each location so that they could be easily processed in the later steps. The data normalization unit brought all the data into the same range which allowed for a better and easier understanding of the data.

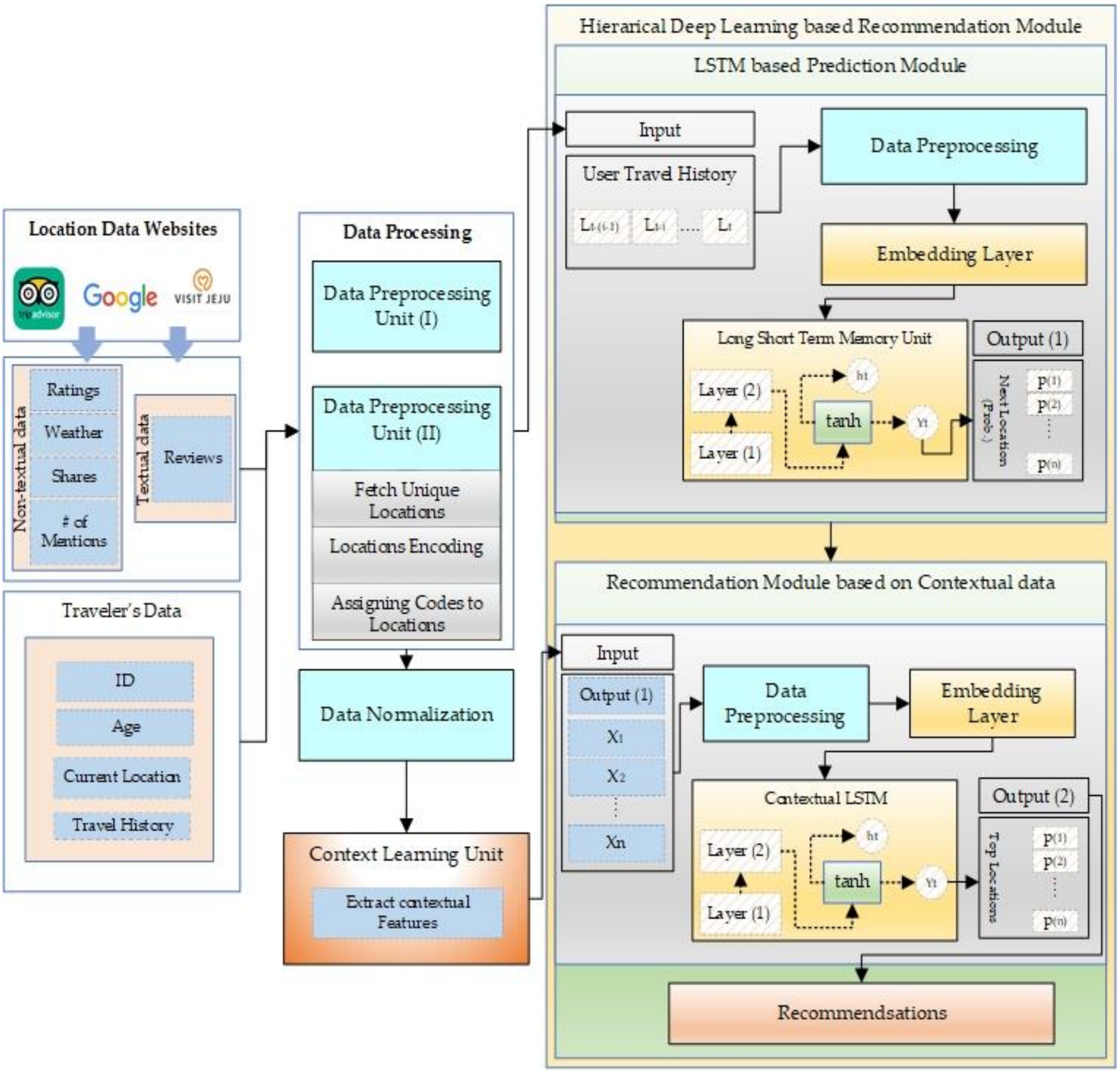


Figure 4. Proposed system architecture.

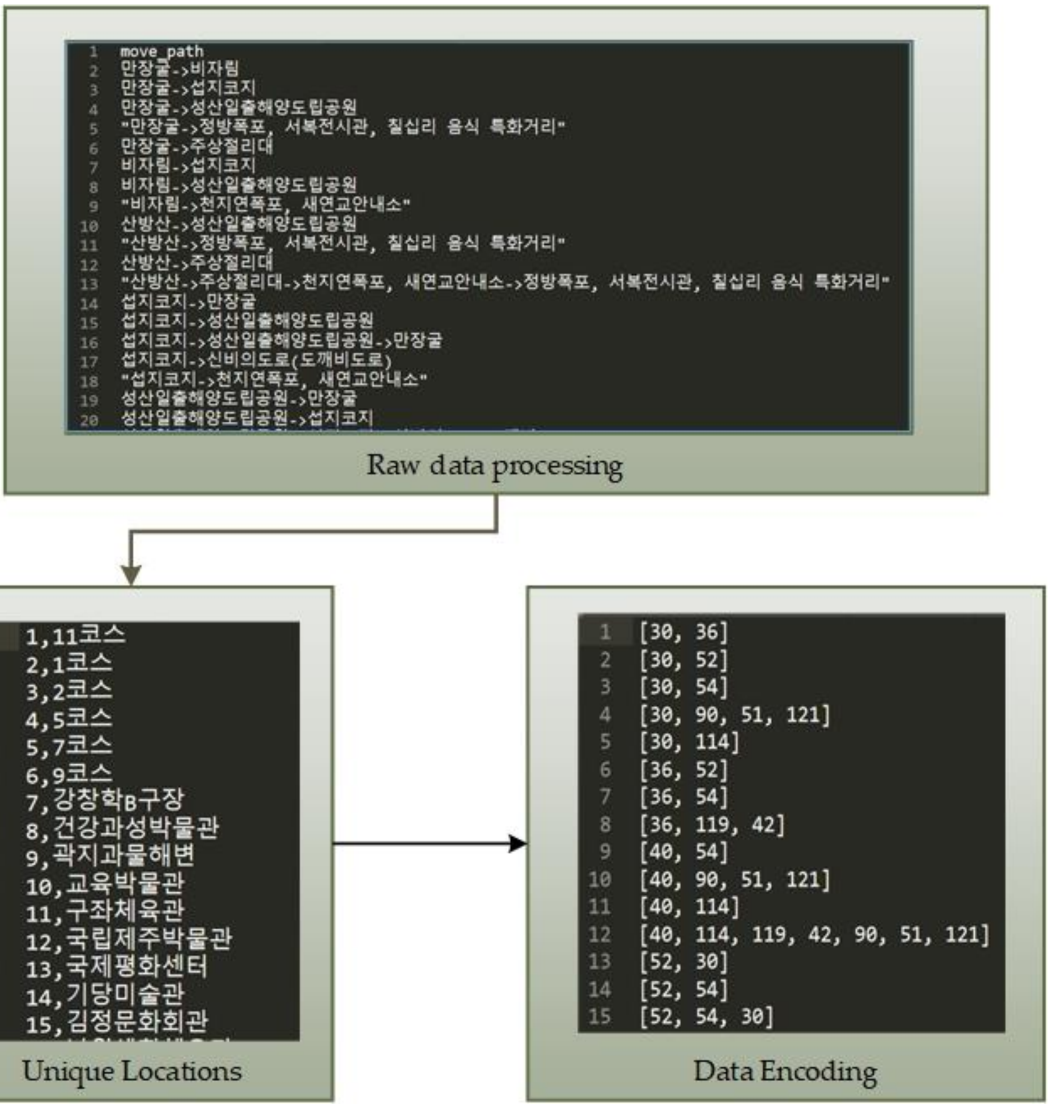
The proposed hierarchical deep learning-based recommendation module comprised of mainly three important layers. The layer I is an LSTM-based prediction module. Here, the traveler's previous visit records were used as input for an LSTM. The LSTM then predicted the visiting probabilities of next location for a traveler. The output of layer I became a part of the input data for the next layer in the framework. The layer II was also an LSTM-based contextual learning module. The input for this module came from the context-learning unit. The contextual data were combined with the output of layer I, i.e., the probabilities data. These combined data were used to feed the LSTM layers as input. The output of this module generated the top n locations to be visited.

4.3. System Modules

This section sheds light on each module of the proposed model. Each module was separately elaborated in more detail including its design, inputs and outputs.

4.3.1. Generating User Trajectories

User trajectories are the traces of a user's previous travels or locations visited so far. The first step of the path from the original mobility traced to the location prediction was characterized by trajectory discretization, a preprocessing phase transforming the raw traces into the input for the neural network model. Figure 5 shows the raw data for the user's movement traces.



```
move_path
만장굴->비자림
만장굴->섭지코지
만장굴->성산일출해양도립공원
"만장굴->정방폭포, 서복전시관, 칠십리 음식 특화거리"
만장굴->주상절리대
비자림->섭지코지
비자림->성산일출해양도립공원
"비자림->천지연폭포, 새연교안내소"
산방산->성산일출해양도립공원
"산방산->정방폭포, 서복전시관, 칠십리 음식 특화거리"
산방산->주상절리대
"산방산->주상절리대->천지연폭포, 새연교안내소->정방폭포, 서복전시관, 칠십리 음식 특화거리"
섭지코지->만장굴
섭지코지->성산일출해양도립공원
섭지코지->성산일출해양도립공원->만장굴
섭지코지->신비의도로(도깨비도로)
"섭지코지->천지연폭포, 새연교안내소"
성산일출해양도립공원->만장굴
성산일출해양도립공원->섭지코지
```

```
1,11코스
2,1코스
3,2코스
4,5코스
5,7코스
6,9코스
7,강창학B구장
8,건강과성박물관
9,곽지과물해변
10,교육박물관
11,구좌체육관
12,국립제주박물관
13,국제평화센터
14,기당미술관
15,김정문화회관
```

```
[30, 36]
[30, 52]
[30, 54]
[30, 90, 51, 121]
[30, 114]
[36, 52]
[36, 54]
[36, 119, 42]
[40, 54]
[40, 90, 51, 121]
[40, 114]
[40, 114, 119, 42, 90, 51, 121]
[52, 30]
[52, 54]
[52, 54, 30]
```

Figure 5. Location data encoding.

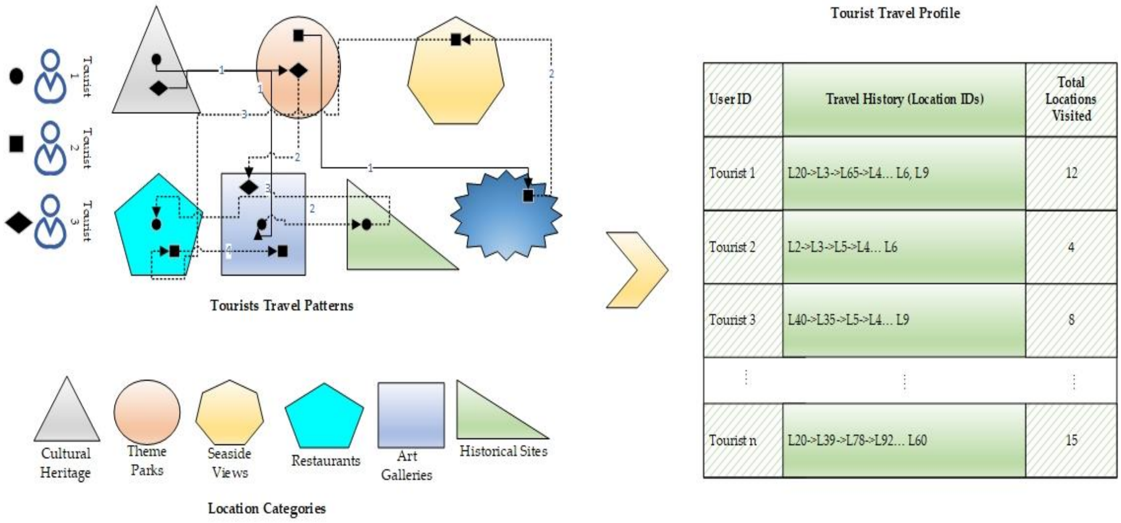
Each row represents a traveler's order for visiting different places. The original data were in Korean and there were different anomalies in the data which were removed during the preprocessing step. Firstly, we converted the data into English, and checked for each location whether it actually existed or not. For the easy understanding of the data, we encoded all of the data and assigned unique identifiers to each location.

After preprocessing, the next step was to generate user trajectories. Based on the unique locations identified in the data, we categorized the data into seven different location categories namely cultural heritage, theme parks, art galleries, restaurants, seaside views, historical sites and others. The others category included locations such as gyms, walking tracks or roads and residential areas. Each category included multiple locations as shown in Table 1.

Table 1. Location categories and popular places.

Location Categories	Popular Places	Number of Locations
Cultural Heritage/Museums	Teddy Bear Museum	10
Theme Parks	Lets Run Park	20
Seaside Views/Beaches	Hamdeok Beach	30
Art Galleries	Chocolate Museum	40
Restaurants	Baghdad, Kimbab Chunkuk	30
Historical Sites	History Museums	10
Others	Shopping Areas, Walking Tracks	7

Without the generality loss, a trajectory of length K was expressed by Ti = <l_1, l_2 . . . .l_K>, and all the locations at time ti were represented as li = ($LATi$,$LONi$,t_i), where the location li was enriched with a pair of latitude and longitude values, denoted as ($LATi$,$LONi$) and a time stamp t_i. Therefore, the preprocessed trajectory resulted in a visiting sequence of location IDs referred to as the "travel history" (L_1, L_2, . . . , L_N). Figure 6 illustrates how the trajectories of different tourists across different location categories are generated based on their travel traces. This way a tourist profile is generated by including user ID, travel history and the total number of locations visited so far.



Tourist Travel Profile

User ID	Travel History (Location IDs)	Total Locations Visited
Tourist 1	L20->L3->L65->L4... L6, L9	12
Tourist 2	L2->L3->L5->L4... L6	4
Tourist 3	L40->L35->L5->L4... L9	8
⋮	⋮	⋮
Tourist n	L20->L39->L78->L92... L60	15

Figure 6. User trajectory generation.

4.3.2. Next Location Prediction

The preprocessed trajectories of the different tourists were used as an input to the first layer of our proposed hierarchical deep learning model. This first layer of the hierarchical model was based on an LSTM network that took travel history data in the form of sequences of unique location IDs. The minimum length of the trajectory sequence was three and the maximum length was 10. The sequences with less than three location IDs were dropped.

The structure of this LSTM-based model consists of four primary layers: an input layer with location IDs as sequences of travel traces, an embedding layer on top of the input layer that generates embedding sequences for the location IDs, a deep learning layer that has a block of two LSTM layers with six LSTM cells in each layer, a fully connected layer and an output layer.

For all the users, the trajectory sequences were converted to the same or a fixed length sequence. In our case, the maximum length of input sequence considered was six. Therefore, all the input sequences had a fixed length of six. After this, an embedding vector was generated for each location ID in the sequence. The user trajectories were encoded into the sequences of the embedding vectors. These embeddings were later fed to LSTM layers as input.

The LSTM cell then took the sequence of six embedding vectors and predicted the next location to be visited. The output of the LSTM block became the input for the fully connected layer. The final layer was the output layer that predicted the next location's visiting probability. These probabilities were produced for each location in the data set. As shown in Figure 7. The location IDs with maximum probabilities had higher chances to be visited next by the traveler.

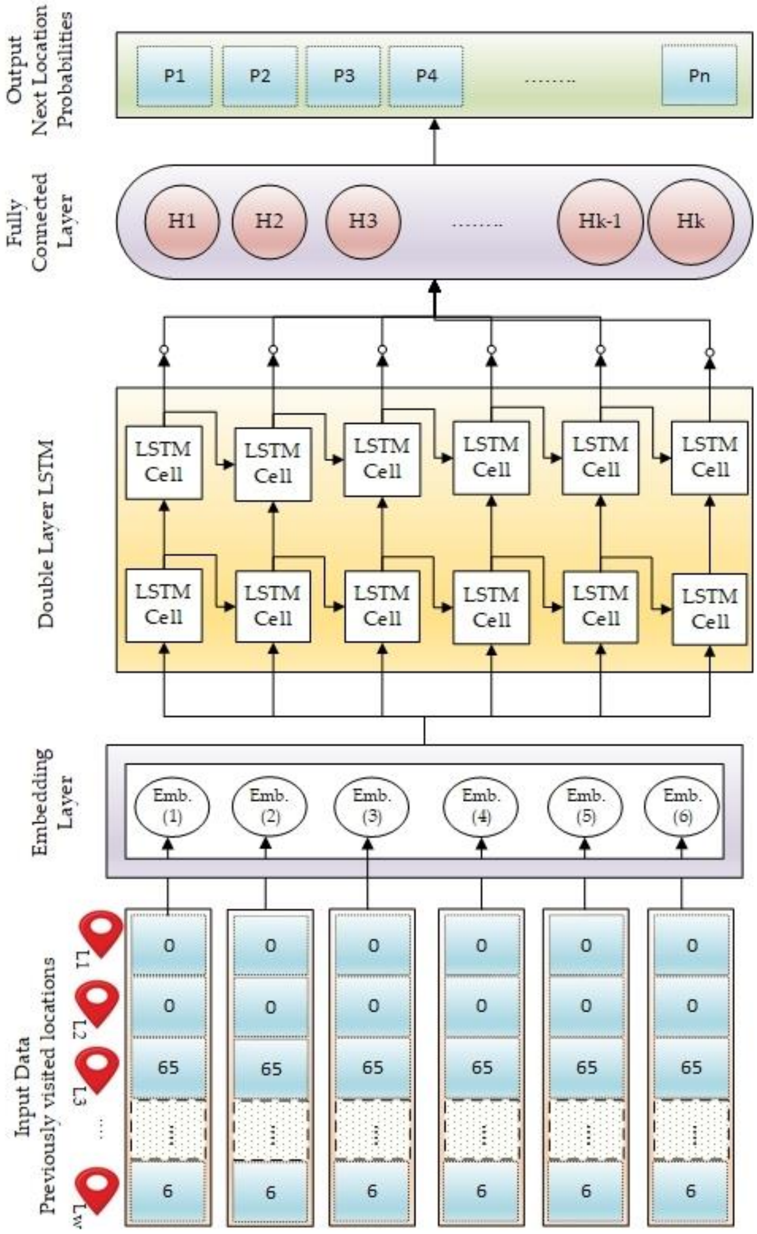


Figure 7. LSTM architecture for the next location prediction probabilities.

4.3.3. Contextual Learning Unit

This unit computed and extracted all the relevant contexts that could impact a tourist's decision to visit a location. These contexts were very important to generate effective recommendations. We crawled different features from multiple sources and we referred to them as "contextual features". All the possible contexts were categorized into five different classes. These categories reflected the popularity of a location based on its number of likes, shares via social networking sites (SNS), sentiment score etc., the environmental factors, the distance-based features and the time-relevant dependencies as shown below in Table 2.

Table 2. Categories of contextual features.

Contexts	Context IDs	Context Description	Range
Popularity	Location Rating	Avg. rating of location from TripAdvisor or Google Museum	[0–5]
	Sentiment Score	Score representing the overall emotional state of travelers.	[−100–+100]
	Number of Likes	Total number of likes a location has	[0–N]
	SNSshares	Number of times a location has being shared	[0–N]
	Wish list	Number of people interested in visiting	[0–N]
	Searched	Number of times a place is looked up	[0–N]
Environment	Weather	Weather conditions	[1–7]
	Temperature	Air temperature of the day	[−n, +n]
	Humidity	Air humidity	[0–N]
Distance	Distance	Distance from the current location to target place	[0–N]
	Nearby Places	Number of tourist spots nearby a specific spot	[0–N]
	Nearby Restaurants	Number of restaurants near a specific spot	[0–N]
Time	Time of the Day	Morning, afternoon, evening	[1–3]
	Weekend	Weekday or not	[0–N]
	Weekday	Weekend or not	[0–N]
Atmosphere risks	Health Risks	Air quality-based risks, pandemics	[1–]
	Environment Risks	Sudden incidents	[0–N]

(SNSshares: shares on social networking sites).

4.3.4. Contextual Features Categories:

All the contextual features are explained below:

Popularity and its Score Estimation

This category of contextual features reflects the importance and popularity of a location. There are multiple baseline parameters to access the popularity of a location such as location ratings, sentiment scores, number of likes, total social network shares, number of people interested in visiting this location and the number of times a place has being searched. Location ratings are a quick assessment parameter to find the quality of a place. It ranges between 0 and 5, where 0 means poor quality and 5 means best quality.

Sentiment score is another parameter that is calculated based on user reviews. These reviews are crawled from multiple tourist sites and an average score is calculated. This score quantifies the traveler's feeling or tone. It is a vital approach for gauging the general attitude towards a tourist spot. We calculated the sentiment score by using a tone analyzer. The range of this feature is between −100 and +100, where −100 reflects extremely negative sentiments and +100 reflects extremely positive sentiments. Other features include the total number of likes, the total number of shares on different social media, the total number of times the location has been searched and how many people have added it to their wish lists.

Popularity index is a very important contextual feature. It is a complex feature as it is derived based from multiple features relevant to a location's popularity. As shown in Figure 8, a candidate location refers to the location for which the popularity index is to be calculated. The nodes directly or indirectly connected to the candidate locations are referred to as "entities". The relations between the entities and the candidate locations are referred to as "paths".

To estimate the popularity of a candidate location, we calculated the individual scores or values of each path. This was achieved by first discovering the highest path score referred to as "S_{max}". S_{max} can be defined as the maximum score of each entity in a path of the location candidate. To calculate the path score, we first took the ratio of the number of entities in a path to $S_{max.}$

Therefore:

$$Score_{path\_i} = \frac{\text{Total score of all entities in path i}}{S_{\max}} \tag{1}$$

The total popularity score can be calculated as $P_{Score_{total}}$ by taking summation of all the path scores i.e., $Score_{path\_i}$. (i = 1−> 6):

$$P_{Score_{total}} = Score_{path\_1} + Score_{path\_2} + Score_{path\_3} + \cdots + Score_{path\_6} \tag{2}$$

$$P_{Score_{total}} = \sum_{i=1}^{6} Score_{path\_i} \tag{3}$$

At the end we calculated the log score of a candidate location as follows:

$$L_{score} = \log_{10} P_{Score_{total}} \tag{4}$$

A higher value of log score reflects the higher popularity of a location and a lower score depicts that the candidate location is not very popular.

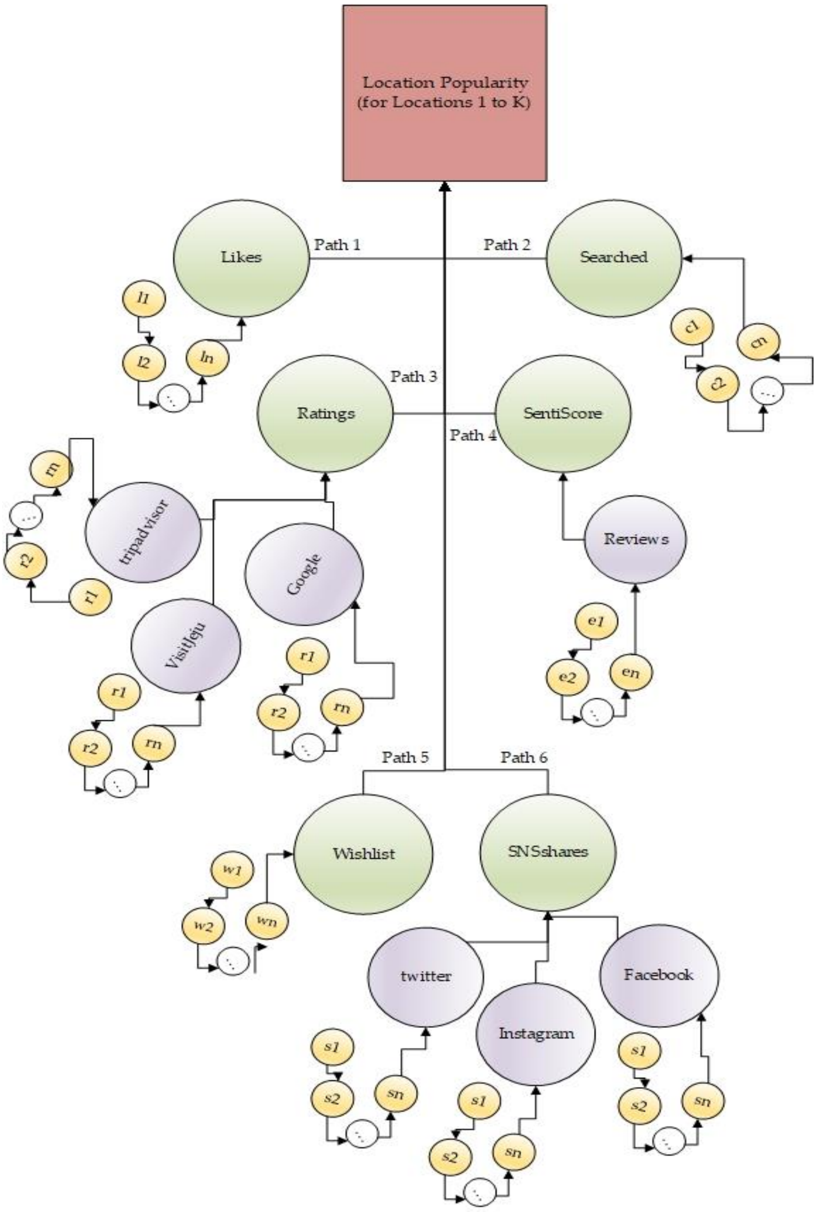


Figure 8. Popularity assessment tree for a specific location. (SNSshares: shares on social networking sites).

Environment

This category of contextual features reflects the importance of external or environmental factors that can impact the recommendations. There are multiple parameters in this category such as weather, temperature and humidity. The weather can be snowy, cloudy, rainy, sunny, etc., as shown in Table 3. The temperature is measured in the degree Celsius scale.

Table 3. Categories of weather.

Weather	Value
Sunny	1
Cloudy	2
Rainy	3
Snow	4
Clear	5
Drizzle	6
Thunderstorm	7

Distance

This category of features comprises the total distance of a traveler from his current location to the target spot, the nearby places as well as restaurants within 5 km. The geographical distance between the current location and the target location is computed by using the Haversine distance function (dist()) as shown in the equation below:

$$\Delta d_t = dist\Big(lat_{k,}\ lon_k,\ lat_l,\ lon_l\Big) \tag{5}$$

Time

This category is very important for a traveler and recommendation systems, as a user's choices and decisions are majorly affected by the time of the day and the day of the week. For example, in the case of workdays, people mostly prefer to visit nearby places, but on weekends they might have plans to visit faraway places as well. This category comprises of three features namely the time of the day, weekend or weekday as shown below in Table 4.

Table 4. Categories of time.

Time of the Day	Value
Morning	1
Afternoon	2
Evening	3

Atmosphere Risks

This class of features is very important especially on days when there are multiple health concerns. These features estimate the risk value and help travelers to take precautions accordingly as shown below in Table 5.

All these contextual features were calculated separately and then combined together. As all these features have different value ranges we therefore normalized the values in one range.

Table 5. Categories of atmosphere risks.

Atmosphere Risks	Descriptions	Value
Health Risks	Dust	1
	Ash	2
	Smoke	3
	Haze	4
	Pandemics	0
Environment Risks	Tornado	0
	Squall	1

4.4. Spatial Join

We collected various features and stored them in different tables. All features have different value ranges and formats. For the training and learning process, we performed a spatial join to combine the data tables into one single data table. It allowed for the easy understanding and handling of the data. Spatial join is a method of joining multiple tables into a single table, and it proceeded based on the following information:

(1) Location data: this data table contains location ID, location name and location coordinates (latitude, longitude);
(2) Next visiting probabilities: this data table comprises of unique location IDs, location name and the visiting probability at time t;
(3) Rating data: this data table comprises of unique location IDs, location name and user ratings;
(4) Observation data: this data table comprises of location coordinates, temperature, humidity, wind pressure, atmosphere risks (e.g., air quality);
(5) Other data: this data table contains location IDs, sentiment scores, shares on social networking sites (SNSshares), number of searches, wish lists, etc.

These five data tables were combined and were later used to perform feature selection.

4.5. Feature Selection

Feature selection methods are responsible for filtering the redundant features and selecting the most desirable and suitable set of features. The application of the feature selection technique on the dataset can diminish the impact of noise and also minimize the computational cost in the modeling phase. There have been many studies performed that show that the classification performance can be increased by applying feature selection [13,16].

4.6. Embedding Layer

To alleviate the challenges of computational complexity, the curse of dimensionality and sparseness in trajectory data, we introduced an embedding layer. This embedding layer does the one-hot encoding and converts the raw or traditional representations into low dimensional vectors. Similarly, the trajectory sequences are converted to sequences of low dimensional dense embedding. These embedding are then fed to the LSTM block.

4.7. Recommendation Process

Tourist spot recommendation systems aim to realistically place a list of fascinating sites for the customers which can be categorized as "must visit places" based on the chronological feedback (e.g., previous visited locations). Such systems have been progressively implemented by locality-based social networks such as Foursquare and Yelp to improve their effectiveness for customers. Lately, several RNN designs have been suggested to integrate contextual information related to user data and history, such as time of the day and location, to successfully seize the tourists' vigorous preferences.

However, these architectures assume that different types of contexts have an identical impact on the user preferences, which may not hold in practice. For example, an ordinary context—such as the time of the day—reflects the user's current contextual preferences, whereas a transition context—such as a time interval from their last visited venue—indicates a transition effect from past behavior to future behavior. To address these challenges, we proposed a hierarchical LSTM-based contextual information-enriched architecture that intelligently selects the relevant features such as distance, time and weather, which can improve the recommendations by using feature importance measures.

5. Implementation and Testing Environment

In this section, we discuss our implementation environment in detail. This section includes the experimental setup details, the explanation of the data collection process and data description, and the model structure.

5.1. Experimental Setup

The experimental setup is summarized in Table 6. The core system components included a long-term support (LTS) version of Ubuntu 18.04.1 as an operating system, with 32 Gb memory, and Nvidia GeForce 1080 as the graphics processing unit (GPU). The implementation was done in Python language along with some application program interfaces (APIs) such as Tensorflow and OpenWeatherMap.

Table 6. System's components and specification.

System's Components	Specifications
Operating System	Ubuntu 18.04.1 LTS
Memory	32 Gb
Language	Python
GPU	Nvidia GeForce 1080
Language Version	3.6.1
APIs	TensorFlow 1.13, OpenWeatherMap

LTS: long-term support; API: application program interfaces; GPU: graphics processing unit.

5.2. Experimental Data

The details for the experimental data are presented in Table 7. For all the experiments, we divided the data into 70% training data and 30% testing data. The total number of locations was 147, with a total of around 36k records.

Table 7. Characteristics of the data used for the experiments.

Experimental Data	Details
No. of locations	147
Reviews	176,834
records	~36 K
Sources	GoogleTripAdvisor, Jeju Tourism Organization OpenWeatherAPI, VisitJeju

For experimental purposes we used three different kinds of settings. The data records were built accordingly. We experimented on different length trajectory data. For this purpose, we generated three types of records, each containing a different number of trajectory length. The maximum size of the trajectory was considered to be 10 and the minimum size varied between three, five and eight locations. Following Table 8, the data setting is described in detail.

Table 8. Data characteristics based on the trajectory size.

Experimental Data	Trajectory Length (Min to Max)		
	3 to 10	**5 to 10**	**8 to 10**
Total rows	36,547	36,547	36,547
Discarded rows (len < Min)	13,440	21,596	29,321
Useful rows	23,107	14,951	7226
Padded rows	20,094	11,938	4213
Expansion rows (>Max)	3013	3013	3013
After expansion rows	14,680	14,680	14,680
Total desirable records after processing	34,774	26,618	18,893

We performed experiments based on the proposed LSTM model, gated recurrent unit (GRU) and bidirectional LSTM models. The parameters are shown in the table above.

6. Results

This section presents the results and analysis for the tourist spot predictions and recommendations based on contextual and sequential features. In Section 6.1, we presented the testing errors. In Section 6.2, we reported the accuracy of the results of proposed model in comparisons with other models.

6.1. Testing and Validation Loss

We used different learning rates to analyze the testing error of our proposed system. We used binary cross entropy as a measure of testing loss. The learning rates varied between 0.1 and 0.001, represented as LR_0.1, LR_0.01, and LR_0.001.

The graph below in Figure 9, shows the training and validation loss comparison of the predicted results for both the proposed LSTM and GRU. The length of the location sequence was considered between a minimum of three locations and a maximum of 10 locations. The number of epochs varied between 0 and 200. The test was performed on a total of 34,774 records. The results show that the performance of the proposed LSTM model was better compared to the GRU as the loss value for both the training and validation sets was less.

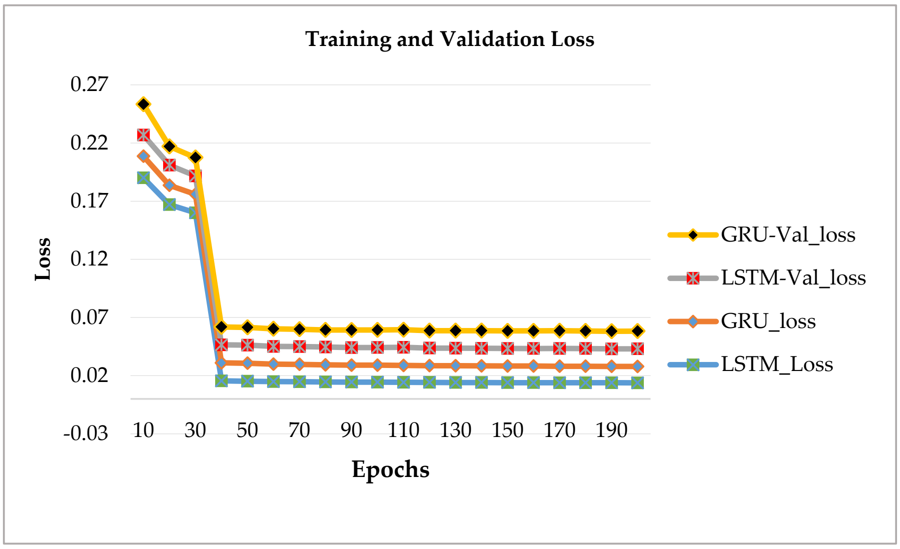


Figure 9. Epochs vs. training and validation loss (for a trajectory length from 3 to 10) in the different models. (GRU: gated recurrent unit).

The graph below in Figure 10 shows the training and validation loss comparison of the predicted results for both the proposed LSTM and GRU models. The length of the location sequence was considered between a minimum of five locations and a maximum of 10 locations. The number of epochs varied between 0 and 200. The test was performed on a total of 26,618 records. The results show that the performance of the proposed LSTM model was better compared to the GRU as the loss value for both the training and validation sets was less.

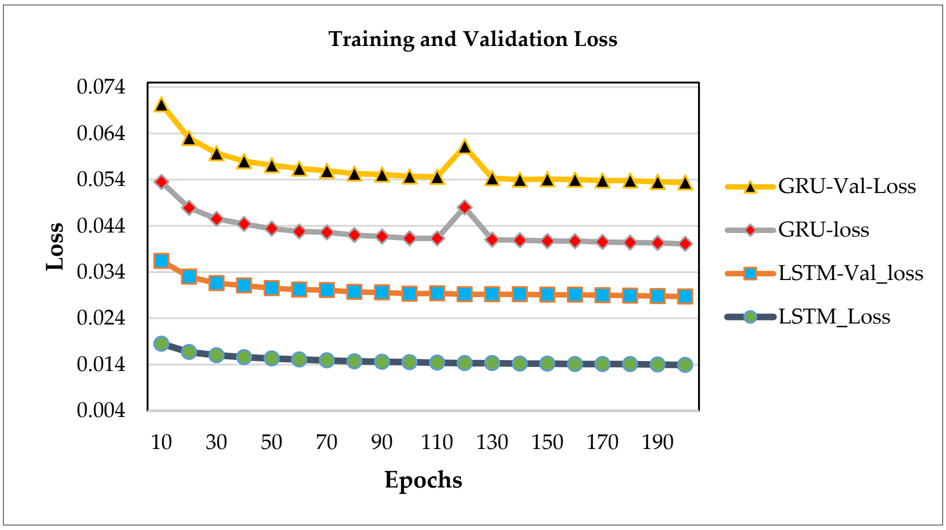


Figure 10. Epochs vs. training and validation loss (for a trajectory length from 5 to 10) in the different models.

The graph below in Figure 11 shows the training and validation loss comparison of the predicted results for both the proposed LSTM and GRU models. The length of the location sequence was considered between a minimum of eight locations and a maximum of 10 locations. The number of epochs varied between 0 and 200. The test was performed on a total of 18,893 records. The results show that the performance of the proposed LSTM model was better compared to the GRU as the loss value for both the training and validation sets was less.

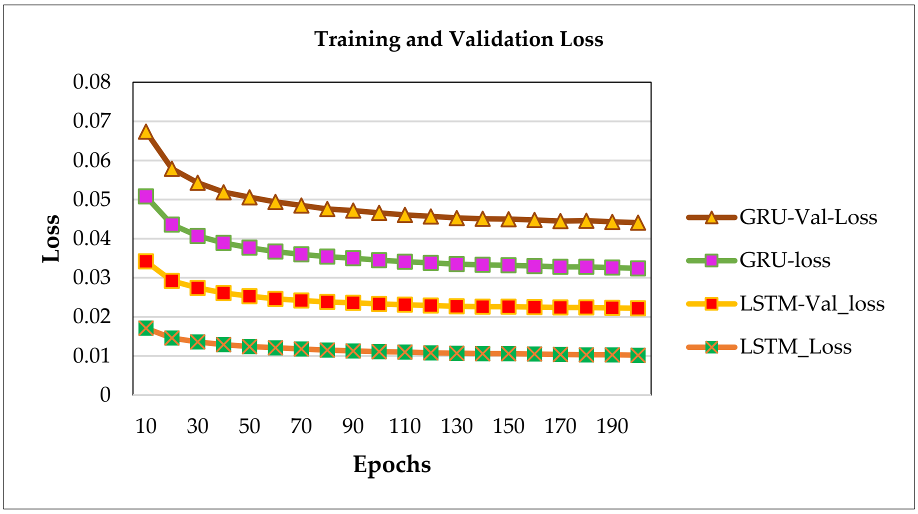


Figure 11. Epochs vs. training and validation loss (for a trajectory length from 8 to 10) in the different models.

As can be observed in the above graphs, the performance of the proposed model was better than the GRU in all three cases, so next we compared the LSTM model results against each other in each scenario to find the optimal route/travel history size. We found that the size of the number of visited locations did matter initially, as shown in the graph in Figure 12, such as when the size is three the loss is higher at the initial epochs, but later it becomes stable. The sequences with a higher number for the user's past locations, show a better performance in the recommendation systems. However, the sequence lengths of five and 10 show more or less the same performance with an average of 4–5 recent locations, which are as good as a higher number of locations such as 8 or 10.

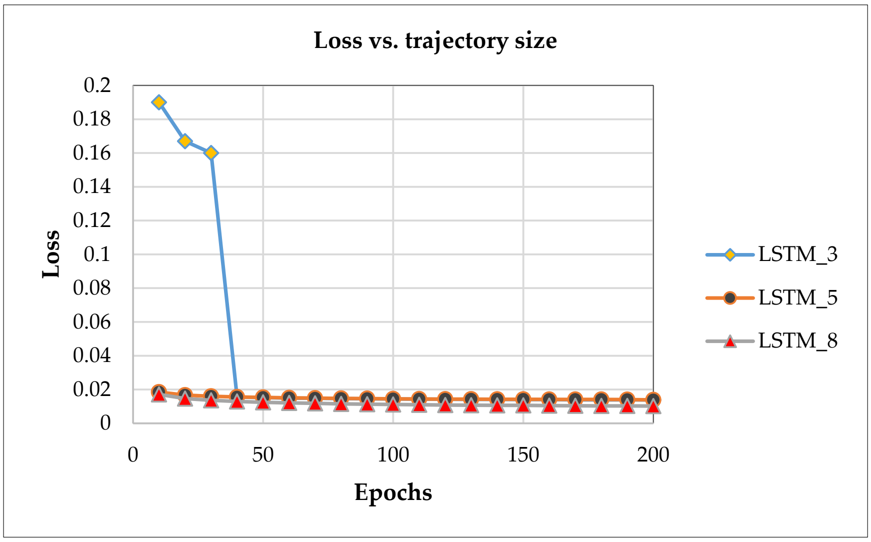


Figure 12. Impact of the trajectory size on the model performance.

6.2. Performance Comparisons with Other Algorithms

We compared the results of our next location prediction task with other very simple yet powerful machine-learning algorithms. As the prediction of the next location is a simple task, we therefore checked and compared the performance of the LSTM used in our model with other basic algorithms. The most commonly used approach for location prediction is the Markov model. However, it only incorporates the user's location sequence and might not perform well for the time-based predictions.

Therefore, we used simple models such as the Naïve Bayes, the Markov Model (MM) and the Weighted Markov Model (WMM) and compared them with deep learning models such as LSTM and GRU. The graph above in Figure 13 shows the validation loss for the different number of epochs for Naïve Bayes, WMM, MM, GRU and LSTM. We can observe that the LSTM and GRU are comparatively performing better on our data set.

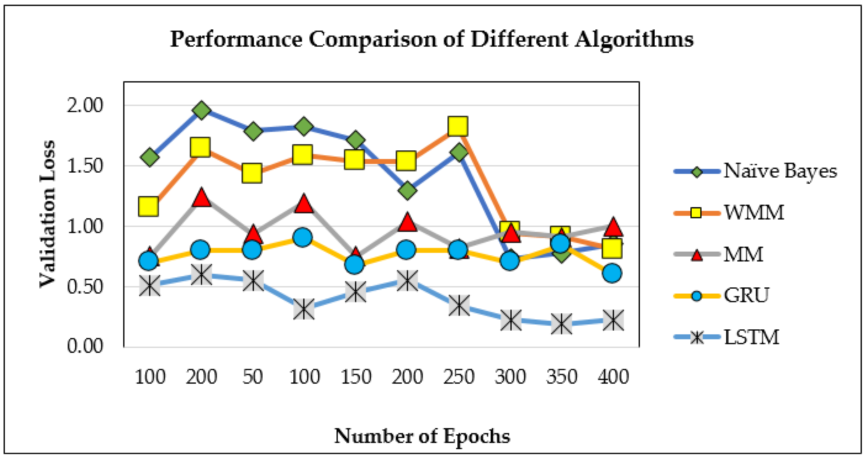


Figure 13. Performance comparison in terms of validation loss of the different algorithms. (MM: Markov Model; WMM: Weighted Markov Model).

6.3. Accuracy

We computed the accuracy of our proposed model and then compared it with other deep learning models such as GRU and bidirectional LSTM, as these two models are also variations of traditional RNN models. The GRU performed well and showed very close performance results with our proposed model. The graph below in Figure 14 shows the comparisons of the accuracies. LSTM achieved the highest, around 97%.

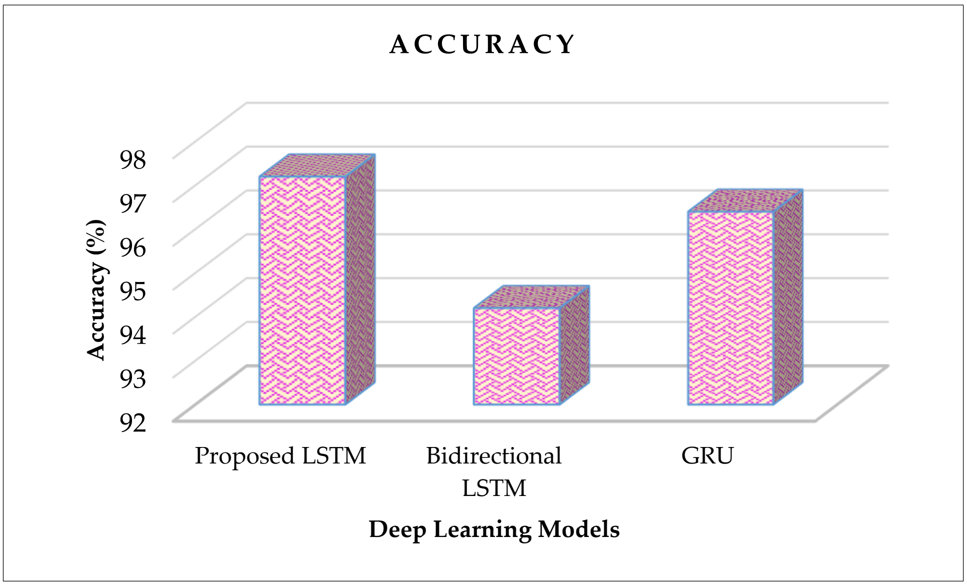


Figure 14. Accuracy of the different deep learning models.

7. Discussion

We proposed a method to predict individual mobility traces of short-term foreign tourists leveraging the collective large-scale motion behavior of people and a deep learning-based methodology adapted to process motion trajectories. The model relies on a recurrent neural network architecture composed of embedding and LSTM layers. We assessed the feasibility of such methodology on short, non-repetitive traces, revealing its potentiality for human mobility studies and applications. We proposed a model to recommend top places to tourists based on their travel patterns and incorporated several different features, primarily including ratings, reviews, distance, time, weather, risks, temperature and popularity. A deep learning-based approach was developed to handle the user trajectories. The proposed model is capable of predicting the probability of a next visit. It is also capable of learning contexts and recommending places that are in the most suitable and desired range of the traveler by considering all the relevant parameters.

Taking everything into account, the introduced deep learning approach showed favorable outcomes in the location prediction and recommendations for a tourist. This fits into the field of smart tourism as it is a deep learning-based artificial intelligence technique. It also enables tourists to improve their experiences and helps in the decision-making process. Recently, such approaches have gained a lot of interest from the researcher community. This study contributes in the development of smart tourism and analytical tools that are actively helping users in decision making based on their patterns of interactions. In general, the proposed approach unlocks an extensive variation of possibly appropriate uses such as customized location-based services, travel planning, time management, risk assessment and popularity estimations. The most relevant and direct implementation choice is linked to enhancing and optimizing the overall travel quality and the experience of individuals. The proposed model enables tourists with customized recommendations comprising of the top ranked places to visit, highlighting the nearby spots at a specific attraction, nearby restaurants, highlighting any risks such as air quality in terms of smoke or dust quantity in the air. Moreover, the assessment of floating tourists at any attraction can help in warning other tourists to be aware of congestion issues. Consolidating individual forecasts can undoubtedly be utilized to consider the future spatial aggregate circulation of travelers, which is unquestionably significant for a few errands, including the modification of the supply of offices and administrations, and economical countermeasures consenting to constant tourist control.

The results show that the LSTM-based approach performed better when compared with other variations of RNN such as the GRU. However, increasing epochs and adding more layers in GRU can enhance its performance. For any recommendation system, the vital and most critical part is to assess the quantity of data suitable for effective and efficient recommendations. It is considered that the more data you have, the more you know about the user and can recommend better. In the case of tourist recommendation systems, user travel history is of key importance. However, the question rises as to how many previous records are enough. To answer this question, we had different experimental settings and for each setting we used a different size of travel history data. We added a few of these settings in previous sections. We used three different sizes of travel history such as (1) a travel history of length ≥ 3≤ 10 (2) a travel history length ≥ 5≤ 10 and travel history of ≥ 8≤ 10. The results show that for travel histories of lengths greater than three have same impact on the final output. As you increase the length, you are going back in the past, while in this fast moving era travelers' choices and decisions also change very rapidly. Therefore, in such systems, the thing that matters most is how recent the records are rather than the length of past records. The more recent activity you have recorded, the better performance and contextual awareness the recommendation system will have.

In summary, this fits in the field of location prediction and recommendations based on user trajectories by utilizing AI systems, especially adding the features that are capable of the deep learning of traveler's movement patterns, revealing that RNN models are a promising technique for design acknowledgment in trajectory study. Moreover, the use of features such as environmental factors, atmospheric risks, location's popularity assessment and the intelligent selection of features based on their importance are also promising tasks to enhance the recommendation results. It also reflects the importance of user-relevant and location-relevant contexts.

8. Conclusions

We proposed a hierarchical deep learning-based recommendation system for tourists. In this paper, the significance of utilizing the potential factors and features in combination with the generally used location data were discussed. We also discussed how the use of factors such as weather, time, distance, climate, user reviews and the impact of nearby places, atmospheric risks and environmental conditions can add significant value to the recommendation systems for tourists. We considered that LSTM as a specific kind of recurrent neural network. The LSTM model and its various variations have accomplished noteworthy performances in various sequence-learning matters in image, speech, music and content analysis, where it is valuable in catching long-run dependencies in sequences of data. LSTMs significantly improve our capacity to deal with long-range dependencies. The proposed model is a hierarchal model that first predicts the probability of a user's next visit to a tourist spot and then incorporates the contextual data to recommend the top places to the user.

We showed that by incorporating the context features such as the location's rating, the atmosphere and the environmental factors, that they can be beneficial for prediction and the next location selection for the tourists. Due to the current Covid-19 global crisis, the tourism industry is globally negatively affected. We included this current context in our feature of environment risks category. In South Korea, the international travel ban has resulted in an increase in domestic tourism. However, it is indeed a point of concern for everyone to be aware of which places are risky to visit and what places are safe to spend some time.

Our results show that the proposed model performed better on our dataset when compared with basic machine learning models, Markov models such as WMM and other deep learning models such as GRU or bidirectional LSTM. The accuracy of the results achieved with the proposed model was the highest (i.e., >97%), followed by the GRU with an accuracy of around 96% and the bidirectional LSTM with an accuracy of around 94%.

9. Contributions and Future Directions

In this paper we illustrated the significance of capitalizing on the relevant contextual data to a tourist spot in accordance with a traveler's next probable visit when recommending a tourist spot. The proposed model is enriched with all relevant contexts and the significant contribution is that it incorporated the atmospheric risks as well. Currently, the global tourist industry is in crisis because of the coronavirus pandemic. Cooperative and context-aware services and applications in tourism will endure to increase in importance in the future. Here, we divided our model into different modules and each module was responsible for capturing useful insights relevant to tourist spot recommendation. The first module extracted tourist spots from given data and extracted the relevant features from different sources, for example, the ratings and reviews of a tourist spot were collected from VisitJeju, Tripadvisor and Google. Similarly, other features such as weather, environmental risks, popularity index and atmospheric elements were learnt. The second module was developed using LSTM that used location sequence embedding and generated the probabilities of the next location to be visited. The third module incorporated the input from the second module and combined it with data from the contextual unit that was responsible for extract all the relevant features to a tourist spot. This third module then recommended the top n places to visit. To the best of our knowledge, our work is the first to explore and highlight the importance of contexts for recommendations such as atmosphere risks and the location popularity. The contexts are enriched with both user movement patterns and current local and global circumstances that can impact the visitor preferences to visit a location.

There are numerous promising opportunities for future work. The proposed model can be applied in other recommendation systems such as movie, book or online product recommendation where we can first learn a user's click pattern and then add other context vectors such as price, season, brand, range, reviews and location. It can also be interesting to see how tourists' behaviors and preferences change or develop over time under specific circumstances. We would like to evaluate our proposed model in a more realistic setting e.g., how users interact or make preferences based on a specific condition, situation or feature. Moreover, for the current coronavirus pandemic, we would like to evaluate our model on large data collected from different tourist spots during this period and critically analyze its impact on tourism and tourist preferences.

Author Contributions: W.S. conceived the idea for this paper, designed the experiments, wrote the paper, assisted in the algorithms' implementation, and assisted with the design and simulation; Y.-C.B. proof-read the manuscript and supervised the work. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) [2019-0-00203, The Development of Predictive Visual Security Technology for Preemptive Threat Response]. And, This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2019S1A5C2A04083374).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Adomavicius, G.; Tuzhilin, A.; Berkovsky, S.; De Luca, E.W.; Said, A. Context-awareness in recommender systems: Research workshop and movie recommendation challenge. In Proceedings of the fourth ACM Conference on Recommender Systems, Barcelona, Spain, 26–30 September 2010; pp. 385–386.
2. Open Data Portal. Available online: https://www.data.go.kr/dataset/3084438/fileData.do?lang=en (accessed on 12 April 2020).
3. The Jeju Weekly. Available online: http://www.jejuweekly.com/news/articleView.html?idxno=6123 (accessed on 12 April 2020).

4. Gretzel, U.; Yoo, K.H. Use and impact of online travel reviews. *Inf. Commun. Technol. Tour.* **2008**, 35–46. [CrossRef]
5. Bennett, D.; Yábar, D.P.B.; Saura, J.R. *University Incubators May Be Socially Valuable, but How Effective Are They? A Case Study on Business Incubators at Universities; Entrepreneurial Universities*; Springer: Cham, IL, USA, 2016; pp. 165–177.
6. Resnick, P.; Varian, H.R. Recommender systems. *Commun. ACM* **1997**, *40*, 56–58. [CrossRef]
7. Bobadilla, J.; Ortega, F.; Hernando, A.; Gutiérrez, A. Recommender Systems Survey. *Knowl. Based Syst.* **2013**, *46*, 109–132.
8. Wang, D.; Liang, Y.; Xu, D.; Feng, X.; Guan, R. A content-based recommender system for computer science publications. *Knowl. Based Syst.* **2018**, *157*, 1–9. [CrossRef]
9. Van den Oord, A.; Dieleman, S.; Schrauwen, B. Deep content-based music recommendation. In Proceedings of the Conference Workshop Neural Information Processing System (NIPS), Lake Tahoe, NV, USA, 5–10 December 2013; pp. 265–2643.
10. Huang, W.; Wu, Z.; Liang, C.; Mitra, P.; Giles, C.L. A neural probabilistic model for context based citation recommendation. In Proceedings of the AAAI Conference Artificial Intelligence (AAAI), Austin, TX, USA, 25–30 January 2015; pp. 2404–2410.
11. Alptekin, G.I.; Büyüközkan, G. An integrated case-based reasoning and MCDM system for Web based tourism destination planning. *Expert Syst. Appl.* **2011**, *38*, 2125–2132. [CrossRef]
12. Hsu, F.-M.; Lin, Y.-T.; Ho, T.-K. Design and implementation of an intelligent recommendation system for tourist attractions: The integration of EBM model Bayesian network and Google Maps. *Expert Syst. Appl.* **2012**, *39*, 3257–3264. [CrossRef]
13. Nilashi, M.; Ibrahim, O.b.; Ithnin, N.; Sarmin, N.H. A multi-criteria collaborative filtering recommender system for the tourism domain using expectation maximization (EM) and PCA–ANFIS. *Electron. Commer. Res. Appl.* **2015**, *14*, 542–562. [CrossRef]
14. Ravi, L.; Vairavasundaram, S. A collaborative location based travel recommendation system through enhanced rating prediction for the group of users. *Comput. Intell. Neurosci.* **2016**, *2016*, 1–28. [CrossRef] [PubMed]
15. Fenza, G.; Fischetti, E.; Furno, D.; Loia, V. A hybrid context aware system for tourist guidance based on collaborative filtering. In Proceedings of the 2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011), Taipei, Taiwan, 27–30 June 2011; pp. 131–138.
16. Li, G.L.; Zhu, T.; Hua, J.; Qiu, D.D.; Wu, R.Z.; Zhang, H.B.; Ji, D.H. Hybrid recommendation system for tourist spots based on hierarchical sampling statistics and Bayesian personalized ranking. *J. Cent. China Normal Univ. Nat. Sci.* **2019**, *53*, 214–221.
17. Li, G.; Zhu, T.; Hua, J.; Yuan, T.; Niu, Z.; Li, T.; Zhang, H. Asking images: Hybrid recommendation system for tourist spots by hierarchical sampling statistics and multimodal visual bayesian personalized ranking. *IEEE Access* **2019**, *7*, 126539–126560. [CrossRef]
18. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the CVPR, Las Vegas, NV, USA, June 26–July 1 2016.
19. Kim, Y. Convolutional neural networks for sentence classification. *arXiv* **2014**, arXiv:1408.5882.
20. Zhang, H.; Yang, Y.; Luan, H.; Yang, S.; Chua, T. Start from scratch: Towards automatically identifying, modeling, and naming visual attributes. In Proceedings of the MM 22nd ACM International Conference on Multimedia, Orlando, FL, USA, 3–7 November 2014.
21. Cheng, H.-T.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.; Aradhye, H.; Anderson, G.; Corrado, G.; Chai, W.; Ispir, M.; et al. Wide & deep learning for recommender systems. In Proceedings of the DLRS, Boston, MA, USA, 15 September 2016.
22. He, X.; Chua, T.-S. Neural Factorization Machines for Sparse Predictive Analytics. In Proceedings of the 40th International ACM SIGIR Conference, Tokyo, Japan, 7–11 August 2017.
23. He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; Chua, Y.-S. Neural collaborative filtering. In Proceedings of the 2017 International World Wide Web Conference Committee, Perth, Australia, 3–7 April 2017.
24. Liu, Q.; Wu, S.; Wang, D.; Li, Z.; Wang, L. Context-aware sequential recommendation. In Proceedings of the 2016 IEEE 16th International Conference on Data Mining (ICDM), Barcelona, Spain, 12–15 December 2016.
25. Liu, Q.; Wu, S.; Wang, L.; Tan, T. Predicting the Next Location: A recurrent model with spatial and temporal contexts. In Proceedings of the AAAI, Phoenix, AZ, USA, 12–17 February 2016.

26. Manotumruksa, A.; Macdonald, C.; Ounis, I. A deep recurrent collaborative filtering framework for venue recommendation. In Proceedings of the CIKM, Singapore, 6–10 November 2017.
27. Yu, F.; Liu, Q.; Wu, S.; Wang, L.; Tan, T. A dynamic recurrent model for next basket recommendation. In Proceedings of the SIGIR, Pisa, Italy, 17–21 July 2016.
28. Tan, Y.K.; Xu, X.; Liu, Y. Improved recurrent neural networks for session-based recommendations. In Proceedings of the DLRS, Boston, MA, USA, 15 September 2016.
29. Koren, Y.; Bell, R.; Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **2009**, *42*, 30–37. [CrossRef]
30. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. BPR: Bayesian personalized ranking from implicit feedback. In Proceedings of the UAI, Montreal, QC, Canada, 18–21 June 2009.
31. Deveaud, R.; Albakour, M.-D.; Macdonald, C.; Ounis, I. Experiments with a venue-centric model for personalized and time-aware venue suggestion. In Proceedings of the CIKM, Melbourne, Australia, 19–23 October 2015.
32. Manotumruksa, A.; Macdonald, C.; Ounis, I. A Personalised Ranking Framework with Multiple Sampling Criteria for Venue Recommendation. In Proceedings of the CIKM, Singapore, 6–10 November 2017.
33. Yao, L.; Sheng, Q.Z.; Qin, Y.; Wang, X.; Shemshadi, A.; He, Q. Context-aware Point-of-Interest Recommendation Using Tensor Factorization with Social Regularization. In Proceedings of the SIGIR, Santiago, Chile, 9–13 August 2015.
34. Yuan, F.; Guo, G.; Jose, J.; Chen, L.; Yu, H. Joint Geo-Spatial Preference and Pairwise Ranking for Point-of-Interest Recommendation. In Proceedings of the ICTAI, San Jose, CA, USA, 6–8 November 2016.
35. Zhang, J.D.; Chow, C.-Y. GeoSoCa: Exploiting geographical, social and categorical correlations for point-of-interest recommendations. In Proceedings of the SIGIR, Santiago, Chile, 9–13 August 2015.
36. Zhao, S.; Zhao, T.; Yang, H.; Lyu, M.R.; King, I. STELLAR: Spatial-Temporal latent ranking for successive point-of-interest recommendation. In Proceedings of the AAAI, Phoenix, AZ, USA, 12–17 February 2016.
37. Yang, L.; Zheng, Y.; Cai, X.; Dai, H.; Mu, D.; Guo, L.; Dai, T. A LSTM based model for personalized context-aware citation recommendation. *IEEE Access* **2018**, *6*, 59618–59627. [CrossRef]
38. Bartoli, F.; Lisanti, G.; Ballan, L.; Del Bimbo, A. Context-aware trajectory prediction. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 1941–1946.
39. Liu, J.; Wang, G.; Hu, P.; Duan, L.Y.; Kot, A.C. Global context-aware attention LSTM networks for 3D action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1647–1656.