Personalized Route Description Based On Historical Trajectories

Han Su
Hansu@uestc.edu.cn
University of Electronic Science and Technology of China Chengdu, China
Chengdu, China

Guanglin Cong
nofloat@163.com
University of Electronic Science and Technology of China Chengdu, China
Chengdu, China

Wei Chen
chenwei19920109@uestc.edu.cn
University of Electronic Science and Technology of China Chengdu, China
Chengdu, China

Bolong Zheng
bolongzheng@hust.edu.cn
Huazhong University of Science and Technology
Wuhan, China

Kai Zheng *
zhengkai@uestc.edu.cn
University of Electronic Science and Technology of China Chengdu, China
Chengdu, China

ABSTRACT

The turn-by-turn route descriptions provided in the existing navigation applications are exclusively derived from underlying road network topology information, i.e., the connectivity of edges to each other. Therefore, the turn-by-turn route descriptions are simplified as metric translation of physical world (e.g. distance/time to turn) to spoken language. Such translation that ignores human cognition of the geographic space, is frequently verbose and redundant for the drivers who have knowledge of the geographical areas. In this paper, we study a Personalized Route Description system dubbed PerRD - with which the goal is to generate more customized and intuitive route descriptions based on user generated content. PerRD utilizes a wealth of user generated historical trajectory data to extract frequently visited routes in the road network. The extracted information is used to make cognitive customized route description for each user. We formalize this task as a problem of finding the optimal partition for a given route that maximizes the familiarity while minimizing the number of partitions, and finding a proper sentence to describe each partition. For empirical study, our solution is applied to three trajectory datasets and users' real experiences to evaluate the performance and effectiveness of PerRD.

1 INTRODUCTION

Navigation applications that determine optimal routes and corresponding turn-by-turn directions in road networks are one of the most used applications in a wide variety of domains. While the problem of computing optimal route has been extensively studied and many efficient techniques have been developed over the past several decades, the turn-by-turn direction computation techniques have have remained unchanged. With the fast-paced development of mobile Internet, the applications related to route description are increasingly used in people's daily lives, and these application providers will record a large number of user's driving data, i.e, trajectory data, every day. These data are particularly useful for producing more effective and cognitive turn-by-turn directions. Figure 1(a) shows an example of a user who expects to go to '9

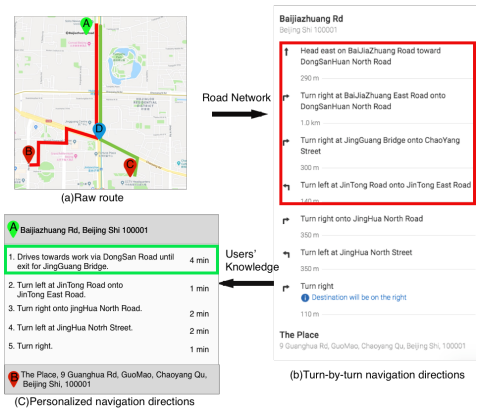


Figure 1: Personalized route descriptions.

Guanghua Rd, Guomao' (marked as B) from her home (marked as A). The route from A to B is denoted by the red line, while the route from A to her work (marked as C) is indicated by the green line. We can see that the two routes share most parts, route from A to D. Figure 1(b) shows the description of route A to B provided by navigation systems. The route description is of the turn-by-turn route description strategy which is computed by taking into account inherent cost measures (i.e., distance and/or travel time) and turn angles (e.g., left or right) between the edges of the underlying road network. Thus the turn-by-turn directions are inevitably verbose even if the details are quite familiar to a particular driver. Moreover, no cognitive summary (e.g., landmarks leading to main turns in the route) is available to the driver, which may be tremendously helpful to perceive the route at once. One way to generate more laconic and intuitive route description is to present the routing information using higher level objects such as routes and landmarks that are frequently traveled by the driver while omitting details which are already familiar to her, such as location C.

*Corresponding author

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3357877

Typically, experienced urban commuters have good knowledge of the city and are familiar with certain parts of the city such as their living regions and working regions. Continuing with our example shown in Figure 1, the driver frequently travels on the green route and hence the navigation system should replace redundant navigation instructions in red rectangle of Figure 1(b) with a single sentence 'Drive towards ***work*** via DongSan Road until exit for Jingguang Bridge', demonstrated in Figure 1(c). Since familiar routes and more cognitive information are involved to describe the routing directions, navigation information becomes easier for the driver to understand. We call the route description strategy which describes routes utilizing user-familiar semantic locations 'personalized route description'. With these two kinds of description strategies compared, it is easy to find out that the target route created by personalized route description is closer to user's travel habits than turn-by-turn route description. Furthermore, personalized route description requires less resources (e.g., screen space) to explain the entire route in a more intuitive way with high level information. Despite the clear advantage shown by personalized route descriptions, there remain few papers working on it. [10] is the only paper that has addressed the problem. Nevertheless, it has some strict requirements, as a result of which not many navigation routes can be personalized described. Besides, [10] only the nearest POI is used to describe route partitions, which is not a good solution in some cases.

Thus in this paper, we propose a new personalized route description system called PerRD based on users' historical trajectory data. In order to achieve personalized route description for the users, there are some challenges we have to overcome. The first one is how to convert the user's history trajectories to the available knowledge appropriately. The second one is how to find an optimally partitioning a given route based on the knowledge. The third one is how to generate a good personalized description. To address these challenges, we propose a two-phrase framework, i.e., route partition and partition description. In route partition, we first measure users' knowledge of the network by utilizing users' historical trajectories. The routes that a user know well will be extracted. Based on the routes known to a user, we divide a given route into several partitions. In partition describing phase, we first develop the template of description and then transform the route description problem into that of finding destination marker problem. In order to solve the problem, we propose a two-step method which firstly detect the types of regions. Then based on the region type, we select a proper landmark to be the destination marker.

In summary, the contributions of this paper are as follows:

- We propose a system PerRD to provide higher-quality personalized route descriptions for urban commuters by leveraging historical trajectory data. We formalize the route description problem, and then propose efficient algorithms to solve them optimally.
- We propose an approach to finding the most suitable location word to describe a route partition. To the best of our knowledge, this paper is the first to use location words to describe routes.
- We conduct extensive experiments based on both real dataset, and as demonstrated by the results, with a proper amount of known routes, PerRD can reduce the number of route descriptions while still providing sufficient information for user to follow the route.

The remainder of the paper is structured as follows. We review the related work in Section 2. Section 3 introduces the preliminary concepts and the work-flow of the proposed system. The route partitioning phase is described in Section 4. In Section 5, we elaborate the detail of the proposed route summarization algorithms. The experimental studies are presented in Section 6. Section 7 concludes the paper.

2 RELATED WORK

The problem referred to in this work is relevant to trajectory process- ing and trajectory querying issues, including trajectory summarization, trajectory compression with semantic meaning, personalized travel recommendation.

Trajectory Summarization. Given a set of trajectories, [7] a solution is proposed cluster the trajectories into several groups, and represent each group is represented by its most central trajectory. [1] summarized a set of trajectories are summarized by providing a symbolic route to represent the cardinal trajectory directions. The output of both [7] and [1] is a trajectory rather than sentences. [17, 18] proposed a framework to construct summary for raw trajectories to describe behaviors of drivers while our work focuses on describing a route based on the drivers' path preferences.

Trajectory Compression with Semantic Meaning. Our work is also related to trajectory compression [14, 24]. A series of researches consider trajectory compression with the constraints of transportation networks [9, 15]. We can simplify the description as long as the moving object is traveling on the shortest path. Though this approach is effective in trajectory compression, the shortest path may not be intuitive or even confusing to users, making it unfit for our application.

Personalized Travel Recommendation. The method to discover personalized routes from trajectories is proposed in [2], while the approach to finding popular routes is investigated in [4]. In [5, 11, 19, 20], the algorithms to recommend routes based on historical trajectories are described. In these applications, the input usually consists of a source and destination and the output is the best route from source to destination based on some criterion, e.g., popularity, safety, while in the input is a route and the output is a route summary. Personalized travel recommendation is a specialized navigation system that aims at finding a route that best matches a user's historical preference. On the contrary, our work is orthogonal to the underlying navigation system and thus can be integrated with any navigation system. That is to say, our system can take an arbitrary route generated by a navigation system and map it to a summary based on a user's historical travel patterns.

[10] is the only work which also attempts to use the routes familiar to users to describe routes. However, [10] is too ideal that the given route can only utilize a user's knowledge when the starting point and ending point of the given route are precisely in the knowledge route; while in this work the given route can utilize a user's knowledge when the given route shares the same sub-route with the knowledge route. In other words, this work can increase

the using probability of a user's knowledge. Moreover, in summarization phase [10] only uses the nearest POI to describe a route segment, while in this work we mine semantic locations for each user and utilize a more proper semantic location to describe a route segment even though the semantic location may be distant from the route partition.

3 PROBLEM STATEMENT

In this section, we introduce some preliminary concepts, and formally define the personalized route description process. Table 1 summarizes the major notations involved in the rest of the paper.

Table 1: Summarize of notations

Notation	Definition
T	a trajectory
l	a landmark in the space
$l.s$	the significance of a landmark
R	a route in a road network
$\overline{RS}$	a route segment
$\overline{RP}$	a route partition
$d(\overline{RP})$	a description of a route partition $\overline{RP}$
$d(R)$	a description of a route R

3.1 Preliminary

Definition 1 (Landmark). *A landmark l is a geographical point in the space, which is stable and independent of trajectories.*

A landmark can be either a Point Of Interest (POI) or a turning point of the road network.

Definition 2 (Trajectory). *A trajectory T is a finite sequence of locations sampled from the original route of a moving object and their associated time-stamps, i.e., $T = [(l_1, t_1), (l_2, t_2), \cdots, (l_n, t_n)]$.*

A raw trajectory T is a finite sequence of locations sampled from the original route of a moving object and their associated time-stamps, i.e., $T = (l_1, t_1), (l_2, t_2), \cdots, (l_n, t_n)$, where p_i denotes a location specified by latitude and longitude and t_i indicates the corresponding timestamp. To facilitate analysis of the trajectories, we use map-matching and anchor-based trajectory calibration [16] to transform the raw trajectory T into a landmark-based trajectory, by treating landmarks as anchor points. After calibration, these trajectories are aligned based on landmarks, fow which a calibrated trajectory can be considered as a sequence of landmarks, i.e., a route.

Definition 3 (Route). *A route R in a road network is defined as a sequence of landmarks $R = [l_1, l_2, \cdots l_n]$ which connect a starting point and destination.*

A route segment $\overline{RS}_i$ of a route R is a sub-route which connects two consecutive landmarks l_i and l_{i+1} of R. For a given route R, it includes $|R| - 1$ segments $R_1, R_2, \cdots, R_{n-1}$. These segments are the basic atoms constructing R. Two segments are named *contiguous segments* if they share the same landmark as the start and the destination respectively, i.e., R_2 and R_3 sharing landmark l_3.

Definition 4 (Route Partition). *A partition of a route R is $\mathbb{P}_R$ such that*

- *Each partition $\overline{RP} \in \mathbb{P}_R$ is a sub-route of R made up of contiguous route segments, i.e.,*
$$\overline{RP} = [\overline{RS}_i, \overline{RS}_{i+1}, \cdots, \overline{RS}_{i+j}]$$
- $\bigcup_{\overline{RP} \in \mathbb{P}_R} \overline{RP} = R$
- $\overline{RP}_i \cap \overline{RP}_j = \emptyset, \forall i, j.$

Clearly, each route segment of a route is covered by exactly one partition. $|\mathbb{P}_R|$ indicates how many partitions are in $\mathbb{P}_R$. Obviously, navigation systems provide route descriptions by partitioning a route into route partitions and then describe each partition, e.g., shown in Fig. 1(b). Thus we can formally define *route description* as follows:

Definition 5 (Route Description). *A description of a route R is $d(R)$ so that*

- *A description $d(\overline{RP})$ of a route partition $\overline{RP}$ is a human readable sentence which describes the adjacent relations between roads covered by $\overline{RP}$.*
- *$d(P)$ is list of sequences that $d(P) = \sum_{\overline{RP} \in \mathbb{P}_R} d(\overline{RP})$.*

All navigation systems only utilize features and topologies of roads without personal information to generate route descriptions. In a road network, a route segment has multiple road features, e.g., street name, grade of road, road width and traffic direction. Navigation systems split a route into route partitions, and within each partition route segments share similar road features. Then navigation systems describe a route in a *turn-by-turn manner* which is computed by taking into account inherent cost measures (i.e., distance and/or travel time) and turn angles (e.g., left or right) between the edges of the underlying road network. In this way, the description of every route is independent of specific commuter, and we define it as *turn-by-turn description* d_{turn}. For example, in Figure 1(b), '*Head east on Baijiazhuang Road toward DongSanHuan North Road*' is an example of turn-by-turn description. For a given route $R = [l_1, l_2, \cdots l_n]$ and its corresponding route partition $\mathbb{P}_R$, its route partition can be denoted by $d(R) = d_{turn}(\overline{RP}_1) + d_{turn}(\overline{RP}_2) + \cdots + d_{turn}(\overline{RP}_{|\mathbb{P}_R|})$. Thus the regular expression of a turn-by-turn manner route description is $d_{turn}(\cdot)^*$.

In reality, a urban commuter is always familiar with certain landmarks and routes, e.g., routes from her home to work place or routes from her work place to a shopping center. We call such user familiar routes *knowledge routes*. In Figure 1(a), the route from home to the work is an example of a knowledge route. Our system splits a route into route partitions, and within each partition route segments belong to a knowledge route or share similar road features. Then our system describes a route in a *personalized manner* if there is a route partition belonging to a knowledge route. In doing so, the description of every route is dependent on specific commuter. We define it as *personalized description* d_{per}. For example, in Figure 1(c), '*Drive towards work via DongSan Road until exit for Jing-Guang Bridge*' is an example of personalized description. For a given route $R = [l_1, l_2, \cdots l_n]$ and its corresponding route partition $\mathbb{P}_R$, the regular expression of a personalized manner route description is $(d_{turn}(\cdot)|d_{per}(\cdot))^*$. As a personalized partition always contains

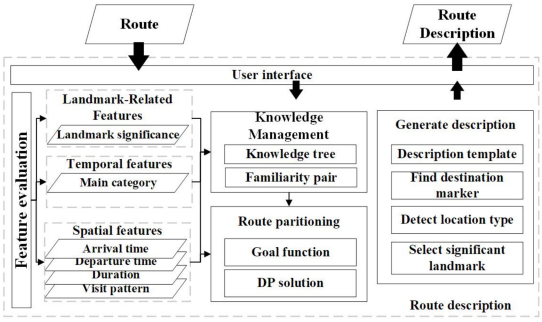


Figure 2: System Overview

more than one turn-by-turn partition, the personalized manner becomes more concise, i.e., less number of route partitions, and more familiarity to drivers than the turn-by-turn manner.

3.2 PerRD System

Figure 2 presents the overview of the proposed PerRD system, which basically consists of two parts: route partitioning and route description. In route partitioning phase, we first measure users' knowledge trees of the network by utilizing users' historical trajectories. Knowledge trees can be constructed off-line as it is independent of data input. Based on users' knowledge trees, we split a given route into multiple partitions. In route description phase, we first devise the template of description and then transform the route description problem into that of finding destination marker problem. In order to address the problem, we propose a two-step method which firstly detects the types of regions and based on the region type we select a proper landmark to be the destination marker. We will detail the discussion on each part in the next two sections.

4 ROUTE PARTITIONING

In this section, we first present our algorithm to explain how to extract users' knowledge based on historical trajectories. The output will be several tree structures which capture the relationship of known routes for each user with corresponding familiarity score. Then we will partition a given route based on a user's knowledge trees.

4.1 User Knowledge Measurement

It is a common sense that if a user visits a POI or travels through a route several times, the user has a high likehood to be familiar with the POI or the route. A user's historical trajectories capture her travel history. Thus, by mining her historical trajectories, we can detect her familiar POIs and routes, i.e., user knowledge. We represent this user knowledge as another metric onto the map. Besides its route distance, traffic condition, etc., we associate each route segment $\overline{RS}$ with scores measuring a user's familiarity with it. Before we give the formal definition of the familiarity measure, we would like to emphasize an important observation about the property of such a familiarity measure. For instance, imagining a user takeing a specific road to commute to work every day so is very familiar with it, however, she may not know how to get to a place she has never visited via that road. From this observation, we can see that a user's familiarity with a route segment is depending on which route we are talking about. Besides the familiarity score is not additive, i.e., getting more familiar with a route segment for a certain route does not mean an increase in familiarity when talking about a different route. Therefore, it is argued that the familiarity score should take the form $f_u(\overline{RS}|R)$, i.e., the familiarity of a particular user u with a particular route segment $\overline{RS}$ is subject to a particular route R. In this paper, we simply define $f_u(\overline{RS}|R)$ as the frequency of user u to visit $\overline{RS}$ in R. Clearly, given a user u, for each $\overline{RS}$ there are multiple familiarity scores, $\{f_u(\overline{RS}|R), \forall R, \overline{RS} \in R\}$, attached to it. In the rest of this subsection we will introduce how to measure user knowledge of a user u by giving her historical trajectories $\mathbb{T}$. We first cluster all the trajectories of a user u and manage them in a succinct data structure—knowledge trees. Then we traverse the collection of knowledge trees built to extract the familiarity scores, and annotate them onto the map.

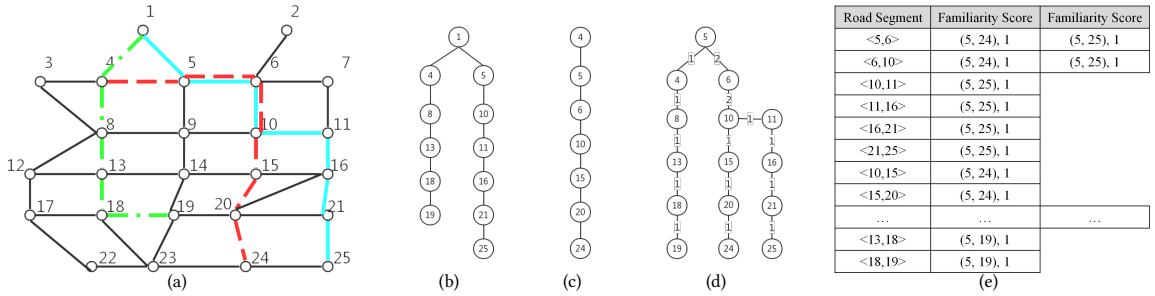
$T_1 = [l_1, l_4, l_8, l_{13}, l_{18}, l_{19}]$ and $T_2 = [l_1, l_5, l_6, l_{10}, l_{11}, l_{16}, l_{21}, l_{25}]$ are two trajectories of $\mathbb{T}$, denoted by the green line and blue line in Figure 3(a) respectively. For trajectories with the same starting point, i.e., l_1, we can easily construct a knowledge tree with the root l_1 in the following steps:

- Add each landmark of trajectories T_1 and T_2 to the vertex set V.
- Add the starting point l_1 as the root and a directed edge from l_i to l_j, denoted by $e(l_i, l_j)$, if there exists a road segment connecting l_i to l_j directly in T_1 and T_2.
- Each edge $e(l_i, l_j)$ is annotated with how many times the user passes $e(l_i, l_j)$ in T_1 and T_2. The annotated number is named familiarity score.

The constructed tree $\mathcal{TR}_{l_1}$ of landmark l_1 is illustrated in Figure 3(b). Similarly, given a trajectory $T_3 = [l_4, l_5, l_6, l_{10}, l_{15}, l_{20}, l_{24}]$ denoted by the red line in Figure 3(a), we can construct a tree $\mathcal{TR}_{l_4}$ shown in Figure 3(c). However, an experienced driver, who has traveled a region or a city many times, may have plenty of starting points. It may cause the construction of many knowledge trees, i.e., high space complexity, even through some starting points are very close spatially, e.g., l_1 and l_4 in Figure 3(a). In order to reduce the space complexity, we need to make changes to the raw trajectories in the following steps:

- Cluster all the starting points and determine a centering POI for each starting point cluster. For a starting point cluster $\mathbb{L}$ and its corresponding center l_c, we construct a new tree $\mathcal{TR}_{l_c}$ and add l_c as the root.
- For a starting point cluster $\mathbb{L}$ and its corresponding center l_c, $\mathbb{T}_{l_c}$ denotes all the trajectories with a starting point l where l belongs to cluster $\mathbb{L}$. For every trajectory $T = [l_1, l_2, \cdots, l_n] \in \mathbb{T}_{l_c}$, we insert l_c to T utilizing the insert method introduced by [8]. This method ensures that its result trajectory is shorter than any other inserted trajectory. Thus the inserted trajectory can be denoted as $T = [l_1, l_2, \cdots, l_i, l_c, l_j, \cdots, l_n]$. We retain the landmarks from l_c to l_n, which in other words the retained trajectory is $T = [l_c, l_j, \cdots, l_n]$.

With the method above, we can change all the trajectories, of which starting points belong to the same cluster, to have the same starting



Road Segment	Familiarity Score	Familiarity Score
<5,6>	(5, 24), 1	(5, 25), 1
<6,10>	(5, 24), 1	(5, 25), 1
<10,11>	(5, 25), 1	
<11,16>	(5, 25), 1	
<16,21>	(5, 25), 1	
<21,25>	(5, 25), 1	
<10,15>	(5, 24), 1	
<15,20>	(5, 24), 1	
…	…	…
<13,18>	(5, 19), 1	
<18,19>	(5, 19), 1	

Figure 3: Example of conducting a knowledge tree

point, i.e., the cluster center. Then we can utilize the tree constructing method introduced at the beginning of this subsection to construct a new knowledge tree. Figure 3(d) demonstrates the new knowledge tree of Figure 3(a).

By using the above explained method, we can establish a collection of knowledge trees $\{\mathcal{TR}_1, \cdots, \mathcal{TR}_k\}$ for a user u, each $\mathcal{TR}_i$ representing a group of trajectories with a shared starting point src as the tree root, and the leaf nodes as different destination points $dest$, and each inner node as a route segment. Then for each $\mathcal{TR}_i$, we perfrom a post-order traverse on the tree. During the traversal for each inner node we visit, we maintain the list of trajectories $(src, dest)$ it belongs to and the corresponding frequencies. It is noteworthy that this can be done easily, as the contained trajectory destinations of a route segment are simply all the leaf nodes in the subtree rooted at the corresponding inner node. In each way, we can annotate each route segment on the map with a list of route and familiarity score pairs $\{(R = (src, dest), f_u(\overline{RS}|R))\}$, as shown in Figure 3(e).

4.2 Partitioning Algorithm

In this section, we introduce the route partition algorithm in PerRD. Although any partition of a route R can lead to a summary, not all of them are suited to a good one. Firstly, it is better for each partition to be intuitive and easy for a user to understand. For example, if a route partition is comprised of route segments with high familiarity scores, a user can have a clear view of how to travel along the route. Secondly, it is easier to generate more compact descriptions if the number of partition size is small. Based on the above intuitions, we define the quality of a partition $\mathbb{P}$ as follows:

$$Q_u(\mathbb{P}(R)) = \sum_{\overline{RS}_i \in \mathbb{P}(R)} f_u(\overline{RS}_i|R) - \lambda|\mathbb{P}(R)|$$

where $|\mathbb{P}(R)|$ indicates the number of route segments in this partition, λ denotes a non-negative constant specified by users, with penalty added to the number of segments in the generated route partition. Thus, finding the optimal route partition $\mathbb{P}^*(R)$ is to find the partitioning scheme that maximizes $Q_u(\mathbb{P}(R))$.

$$\mathbb{P}^*(R) = \underset{\mathbb{P}(R) \in \text{all partitions}}{\text{argmax}} \quad Q_u(\mathbb{P}(R))$$

One naïve approach to find $\mathbb{P}^*(R)$ is to enumerate all the possible combinations of route segments and choose the one with the highest quality. Nevertheless, the time complexity of this naïve approach is exponential with the number of links in the route. In the following part, we will demonstrate that dynamic programming can be used to find the optimal route partition with polynomial time complexity.

Considering a route $R = l_1, l_2, \cdots, l_n$, we use $R(1, k))$ to denote its prefix subsequence $l_1, \cdots, l_k$. Clearly, $\mathbb{P}^*(R) = \mathbb{P}^*(R(1, n))$. Given a prefix subsequence $R(1, k)$, we define its conditional optimal route partition $\mathbb{P}^*(R|R')$ as the optimal route partition of R with the last route segment equals to R'. The optimal partition of $R(1, k)$ is the best conditional optimal route partition among all candidates and belongs to the route of the last route segment. That is,

$$\mathbb{P}^*(R(1,k)) = \underset{\forall R' \in \mathbb{R}(R(k-1,k))}{\text{argmax}} \quad Q\left(\mathbb{P}^*(R(1,k)|R'\right)$$

We can recursively derive $\mathbb{P}^*(R(1, k))$ from $\mathbb{P}^*(R(1, k-1))$, and ultimately derive $\mathbb{P}^*(R(1, n))$ and thus $\mathbb{P}^*(R)$. The derivation can be summarized as follows:

$$Q(\mathbb{P}^*(R(1,k)|R')) = \underset{\forall R'' \in \mathbb{R}(R(k-2,k))}{max}$$

$$\begin{cases} Q(\mathbb{P}^*(R(1,k-1)|R'')) + f_u(R(k-1,k)|R') & \\ \qquad \textbf{if } R' = R'' & (1) \\ Q(\mathbb{P}^*(R(1,k-1)|R'')) + f_u(R(k-1,k)|R'') - \lambda & \\ \qquad \textbf{if } R' \neq R'' & (2) \end{cases}$$

Complexity Analysis: there will be totally N sub-problems, where $N = |R|$ is the number of links in R. Each subproblem can be solved by considering $|\mathbb{R}|$ candidates from the previous sub-problem, where $|\mathbb{R}|$ is the average number of candidate routes for a given link that is usually quite small (≤ 5). Therefore, the time complexity is $O(N \cdot |\mathbb{R}|)$.

Space Complexity: as the algorithm only has to store the information on the conditional optimal partitions of two adjacent steps, thus the total space complexity is $O(|\mathbb{R}|)$.

5 ROUTE SUMMARIZATION

After discussing the route partitioning strategy, we now shift focus to how to generate its corresponding personalized route description in this section. As mentioned above, the personalized route description will describe a turn-by-turn partition with turn-by-turn description d_{turn} and describe a personalized partition with personalized description d_{per}. Thus, for a given route $R = [l_1, l_2, \cdots l_n]$ and its corresponding route partition $\mathbb{P}_R$, the regular expression of a personalized route description is $(d_{turn}(\cdot)|d_{per}(\cdot))^*$.

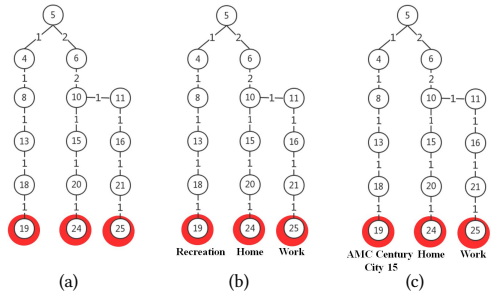


Figure 4: Example of finding a proper destination marker.

For a route description template shown below, there are three types of key information, i.e., 'action', 'current location' and 'destination marker'.

'action' on 'source location' to 'destination marker'

Action is used to indicate the next step of the user's behavior such as 'turn left', 'turn right', 'make a U-turn' and so on. Source location reveals the current location (POIs or roads) of a user. Destination marker represents the destination which can be a POI or a road. As shown in Figure 1(b), 'head east' is an action, 'Baijiazhuang Road' is the source location and 'Dongsanhuan North Road' is the destination marker. Apparently, action and source location are the inherent attributes of topologies of routes, so that there is no difference between descriptions of action and source location in both turn-by-turn description and personalized description. On the contrary, destination markers between turn-by-turn description and personalized description can be starkly different. For instance, destination markers of turn-by-turn description are mostly street names, while destination markers of personalized description are semantic locations such as prominent landmarks and even aliases indicating locations (e.g., home, work, school and so on). The well-known landmarks (e.g., the Times Square) and users' location aliases often make urban commuters recall routes leading to these destinations. Thus, the route description process is filling the route description template by finding proper 'action', 'source location' and 'destination marker' for each route partition. **As finding proper 'action' and 'source location' has been well studied, our work will focus on how to find a proper 'destination marker' for each route partition in personalized route description.**

In practice, there are some *location words* that users are fully aware of without giving detailed geographic information such as location word 'home'. It is the best way to use these personal location words as destination markers cause using location words that can avoid using detailed geographic description. It is worth noting that a user always parks her car near her destination l_{des} not exactly l_{des}. Besides, a user does not always park her car in the same place every time she arrive l_{des}. For example, a user may park her car in a different parking lot not far from her workplace. Therefore, we cannot simply use the ending point of a trajectory as a destination marker. Based on the idea that a user's subsequent behavior patterns are similar when the stay points are within a certain range, we cluster a user's destinations into clusters $[cs_1, cs_2, \ldots, cs_m]$. The red circles of Figure 4(a) denote the destination clusters of Fig. 3(d).

As there are many roads and landmarks within each cluster, we propose a two-step method to find a proper location word $w(cs)$ to represent each cluster. The first step is to detect the most likely location word type of a cluster and the second step is to select the most significant landmark falling into the location word type. In the following section we will detail the two-step algorithm.

5.1 Detect Location Type

To facilitate practical use, we define five types of locations: *Home*, *Work*, *Restaurant*, *Recreation* and *Others*. There are several works [21–23] as attempt to detect the activities of a trajectory. For the activity-oriented detecting purpose, these methods are designed to have a higher probability to classify a cluster as the type of recreation. Besides, our application scenario is detecting the type of a destination cluster instead of a sole destination. Thus, to solve this problem, we propose a new classification model which leverage the of power of both spatio-temporal features of destination clusters and sequential features of trajectories:

5.1.1 Spatio-temporal Features of Destination Clusters . The user's behavior pattern has a close relationship with the spatio-temporal characteristics. For instance, people have a great tendency to stay at home for a long time in the evening, and go to work in the weekday morning. Moreover, people may be *shopping* or *eating* in a business intensive place, and visiting a friend's home in a residential area. Thus we endow every cluster cs_i with a temporal feature vector to capture a user's living habits in the region and a spatial feature vector to describe region characteristics. We list some features as follows:

Temporal feature:

- Arrival time and departure time: a certain visit of a region has its concrete arrival time and departure time. In this paper, we use the time of a trajectory entering a region as the arrival time and the time of a trajectory leaving a region as the departure time.
- Duration: the visit duration indicates how long a user has stayed in a region. It can be calculated from arrival time and departure time. Notably, a user does not record location at any time. Therefore, when system does not get a user's location for a long time, like one day, the duration will not be calculated.
- Visit pattern: the visit pattern of a region indicates the frequency and distribution of visits. We extract the most frequent visits in a week (Monday, Tuesday, $\cdots$, Sunday), the most frequent visits in a day (1:00am , 2:00am, $\cdots$, 12:00pm), the distribution of visits of a week and the distribution of visits of a day in this paper.

Spatial feature: The spatial feature used in paper is the function of a region (e.g., residence, supermarket, etc.), which in other words is the main category of POIs within a region. The TF-IDF idea is conducive to exploring the main categories of POIs within a region. A POI category vector is defined as $< v_1, v_2, \ldots, v_k >$, where k represents total number of categories and v_i indicates tf-idf value of i-th POI category. v_i is defined as follows:

$$v_i = \frac{n_i}{N} \times log\frac{T}{t_i}$$

where n_i denotes the number of POIs belonging to category i, N indicates the total number of POIs in this region, T refers to the total number of POIs in the POI database, and t_i indicates the total number of i-th category POIs. The greater v_i is, the more significant the i-th category POIs are in this region.

We choose XGBoost [3] as the classifier in this step, which is a concrete implementation of Gradient Boost Decision Tree (GBDT) algorithm, in view of its excellent classification performance and the great modeling ability for the missing values in the features. For each cluster cs_i, in order to make the train data more robust, we discretize some temporal features (e.g., duration is divided into high, medium, and low), and calculate the corresponding statistical characteristics such as mean, variance, maximum and minimum. We take both spatial features and temporal features as the input of the classification. Once the XGBoost is trained, from a input vector we can obtain a probability vector $\vec{\Pr} = [\Pr_1, \Pr_2, \ldots, \Pr_t]$, where $\Pr_i$ indicates the probability of a cluster cs belonging to the i-th predefine type. If cs is in a continuous visit sequence, then in step (2), sequence constraint is applied to further determine the type of cs_i, otherwise the category is the result in $\vec{\Pr}_i$ with the highest probability.

5.1.2 Sequential Features of Trajectories. All of the features described above are independent attributes of the clustering region. However, human's behavior is usually sequential, e.g., people usually go *shopping* before going back *home*, and they probably won't go *eating* next from a restaurant gathering area. Thus, the type of each cluster in the visit sequence is affected by its front and rear elements.

In order to achieve the desired effect, we decide to use Bi-directional Long Short-Term Memory (BILSTM) to trajectory sequential pattern problem. LSTM [6] is a highly popular neural network, and it has achieved remarkable results in voice recognition, sequence labeling and other tasks. The basic idea of BILSTM is to merge two LSTM networks (one is forward, another is backward) by connecting them to the same input layer and output layer. So this structure is capable to provide the complete past and future context information to predict each point in the input sequence, that is, it can naturally help us achieve the effect of sequence constraint.

Inspired by the idea of model stacking, we take the probability vector $\vec{\Pr}$ as the new input data in this part. The reason is that it can preserve the first step classifier's effect while reducing the dimensionality of the feature vector to BILSTM.

Firstly, we use a user's daily behavior to create a vector sequence $[\chi_1, \chi_2, \ldots, \chi_t]$ as the input of neural network. Here t indicates the number of clusters which a user visits continuously, and χ_i denotes the feature vector of i-th cluster. Then. the back-propagation through time algorithm (BPTT) can be applied to optimize parameters in the BILSTM. With softmax layer as the output layer, once the BILSTM is trained, from an input vector sequence, we can obtain a probability vector $\vec{\Pr}_j = [\Pr_1, \Pr_2, \ldots, \Pr_t]$, in each step, where $\Pr_i$ is the probability of j-th cluster in the user's visit sequence falling into the i-th pre-define type. We choose the maximum one as the result. Neverthless, the same cluster cs_i may be classified into different pre-types in different sequences. We adopt the voting method to select the most frequently occurring pre-type as its final forecast type. Figure 4(b) illuystrates the location types of Figure 4(a).

5.2 Select Significant Landmark

In route description, two aliases *home* and *work* are sufficiently intuitive to use straightaway, while $restaurant$, $recreation$ and $others$ need to be replaced with certain POIs (e.g., recreation is replaced with Times Square). Therefore, in this paper, we need to select a proper landmark for type restaurant, recreation and others. It is a common sense that landmarks have a different significances. For instance, the White House is famous globally, but Pennsylvania Ave, where the White House is located, is only known to the locals in Washington DC. The selected landmark should have a high significance, so that users can recall the route to it accurately. In this work, we utilize trajectories of cars in the target city to infer the significance of landmarks. By regarding the travelers as authorities, landmarks as hubs, and check-ins/visits as hyperlinks, we can leverage a HITS-like algorithm such as [27] to infer the significance $l.s$ of a landmark l. Thus for cluster cs and its location type $cs.type$, it is easily to obtain the landmarks (denoted by $\mathbb{L}_{cs}$) which are of type $cs.type$ within region of cs. The selected landmark l_{opt} is defined as follows:

$$l_{opt} = \arg\max_{l \in \mathbb{L}_{cs}} l.s$$

Figure 4(c) presents the selected landmark of Figure 4(b). Using this two-step method, we can take aliases *home* and *work* or l_{opt} as destination marker in personalized route description.

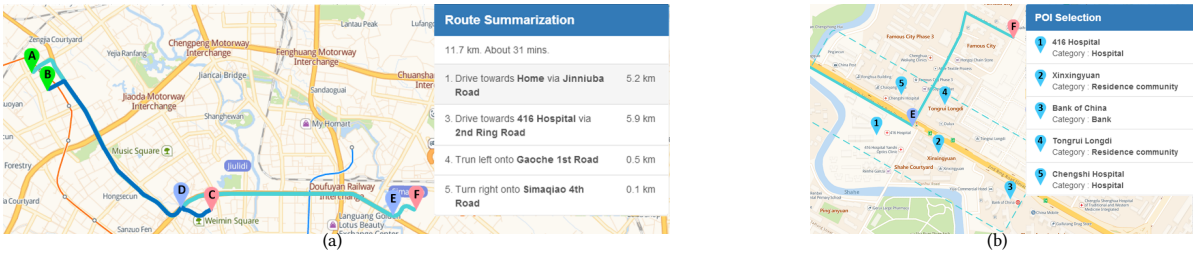
6 EXPERIMENT

In this section, our experimental results are presented to evaluate the performance of proposed route description system PerRD. PerRD is performed using Java and Python on Ubuntu 16.04. All the experiments are conducted on a computer with Intel Core i7-4770K(3.9GHz) CPU and 16GB memory.

6.1 Experimental Setup

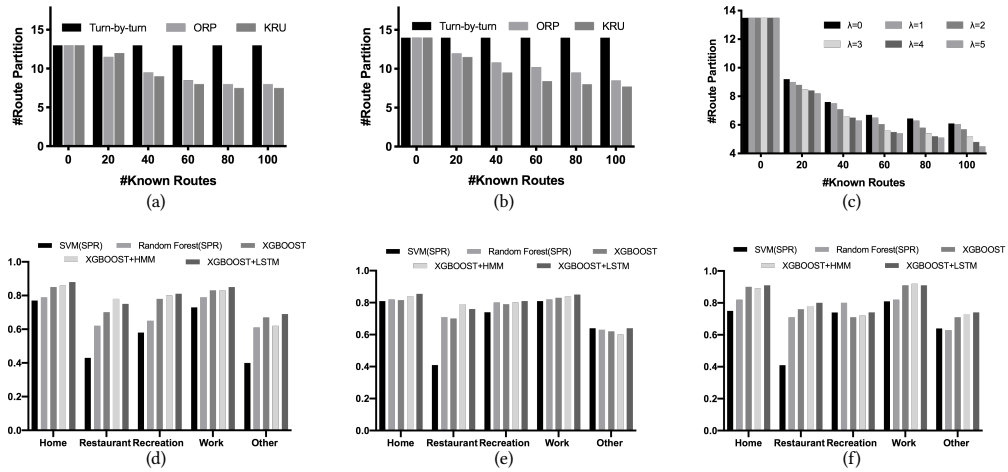
Road Network: we use two road networks (Beijing and Chengdu) from OpenStreetMap to carry out the experiments. The first road network contains 71,646 vertices and 107,396 edges, while the other contains 29,214 vertices and 66,541 edges. Meanwhile, for each city, we have a POI database. There are approximately 1 million POIs in Beijing and 200 thousand in Chengdu.

Trajectory dataset: three groups of real trajectory datasets have been used in these experiments: (1) Geolife dataset [25**?** , 26]: This is a Microsoft Research Asia project which collects 182 users' trajectories in a period of over three years. We only select the GPS sequences whose label is "Car", "Taxi" or "Bus" here. (2) Beijing taxi trajectory dataset: a real-world trajectory dataset that is generated by more than 33,000 taxis in Beijing over four months and contains about 100,000 trajectories. (3) Chengdu trajectory dataset: we collected nearly 800 navigation trajectories from 20 volunteers within one month in Chengdu. As each volunteer carried a mobile phone with the GPS recording application, the program was set to record a time-stamped latitude-longitude pair every 5 seconds. Besides, in order to simulate the real data samples collected by navigation system in the actual using process, they just activate the software to record the GPS trajectory when navigation is in use. Eventually, they are required to label a predefined type for each stay point. In addition, we also use the routing engine OSRM [12] to generate



(a) (b)

Figure 5: (a) An example of personizlied route description, and (b) landmark ranking in finding proper destination marker.



(a) (b) (c) (d) (e) (f)

Figure 6: (a)(b) Information compression results on Geolife dataset and Taxi dataset, respectively. (c) Effect of parameter in route summarization. (d)(e)(f) Comparison of location type detection algorithms on precision, recall and F1-score, respectively.

about 1000 virtual trajectories to expand the dataset, and label a predefined type for each stay point.

Baselines: we compare our proposed Knowledge-based RoUte partitioning method (KRU) and destination marker selecting method with the following algorithms:

- Optimal Route Partition (ORP) [**?**] uses natural routes, which are constructed by connecting adjacent road segments with the same characteristics, and mines routes known users so as to reduce the number of segments in the path by dynamic programming.
- Semantic Place Recognition (SPR) [13] is a comprehensive recognition approach that only exploits the temporal features to train a classifier in the first place. The prediction result of the classifier and the spatial features are weighted to calculate the probability that the current place belongs to each type. Then according to the sequence of the places visited by user in the trajectory, the category probabilities of these places are treated as new features for further prediction by HMM.

Parameters Setting: the penalty for new route segment λ is set to 3 in our method, while it is set to 0.2 in ORP. For SPR, considering the difference of the actual application scenarios, we continue useing the temporal features mentioned in this paper to train classifier. In the following experiments, parameters are set to default values if unspecified.

6.2 Case Study

Before assessing the performance of the system, we peform a case study to show the effectiveness of the personalized route description system. Figure 5(a) presents a case study where known routes are used to generate more concise route descriptions. It is assumed that the route represented by the bluish line from the point A to the point F is a route navigated by navigation systems for the user. By analyzing historical trajectory data, PerRD finds that the user is very familiar with the route R_{AD} from A to D and route R_{DE} from D to E. R_{AD} represents the route connecting A to the user's 'home', R_{DE} denotes the route connecting D to E and the region E is identified as type 'others'. As a result, instead of providing verbose turn-by-turn directions, PerRD generates the description "Drive towards *Home*" to be the description of R_{AD}. Figure 5(b) shows the top five landmarks belonging to type of 'others', which are *416 Hospital*, *Xinxingyuan* and so on, ranked according to the significance of the landmarks. All of these buildings are conspicuous enough to make

driver recall the roads around there and know where she needs to go in this step. When to describe R_{DE}, PerRD finds the most significant landmark belonging to the type of 'others', that is '416 Hospital'. It can be seen that the personalized route description is concise and intuitive to users. As for the remaining parts of the route from E to F that the driver is unfamiliar with, PerRD still provides the detailed turn-by-turn description.

6.3 Performance Evaluation

In this section, we test the methods proposed previously by collecting users' subjective feelings and validating the objective trajectory data. To make the evaluation purpose, we use the first two datasets (Geolife and Taxi) to perform the same experiments to test the effect of route partitioning and route description, while the third dataset (Chengdu) is used to examine the result of destination marker selecting. Users' historical trajectories are split into two parts. The first 80% of trajectory data is taken as training data to mining the users' familiarity with each road while the rest 20% is treated as new input routes to test the effect of system.

6.3.1 ***Effectiveness of route partitioning.*** **Information Compression** As discussed above, using users' knowledge can make the route description more concisely with fewer sentences. Generally, each route partition needs one sentence to describe it, and we use the number of route partitions to approximate the number of sentences. We randomly choose 1000 source and destination pairs from the testing data, then using OSRM routing engine to generate routes. The average length of the route returned by the navigation system is around 15km. To demonstrate the effect of the information compression, for each dataset, we combine trajectories created by different users into one trajectory dataset and treat it as new commuter's trajectory to ensure that each person has at least 100 historical trajectories. Figure 6(a) and 6(b) show the average number of route segments generated by the turn-by-turn approach, ORP and our algorithm in the case of different number of user's known routes. To simplify the description, we use $|\mathbb{P}_t^*|$, $|\mathbb{P}_o^*|$and $|\mathbb{P}_b^*|$ to represent the number of route partitions of 'turn-by-turn', 'ORP' and 'KRU' respectively. From Figure 6, we have the following three main observations: (1) all $|\mathbb{P}_o^*|$ and $|\mathbb{P}_b^*|$ are smaller than $|\mathbb{P}_t^*|$, which means that user's knowledge can really help us build more concise route descriptions. When the number of known routes reaches 100, $|\mathbb{P}_b^*|$ is less about 1/3 of $|\mathbb{P}_t^*|$. (2) $|\mathbb{P}_b^*|$ is significantly smaller than $|\mathbb{P}_o^*|$, which is because ORP can use knowledge routes only when the starting point and the destination of a knowledge route must be on the given route. Thus the probability of utilizing knowledge route is low as compared to our method. (3) with the increase of known routes, the rates of descent of $|\mathbb{P}_b^*|$ are gradually slowing down.

Effect of λ: λ is used to constrain the number of route segments in a route. Figure 6(c) shows the relationship between the known routes and the number of route partitions, which is the average of route partition results on both two datasets with different λ. In this experiment, it is observe that: with the increase of λ, the number of route partitions decreases and the magnitude of the decrease is gradually reduced, which is because when balancing the user's familiarity score with the route and the number of segments, the constraining force of the latter gradually reaches its peak. If λ is overly large, the algorithm will ignore the familiarity of routes.

6.3.2 Effectiveness of destination marker selecting. In this section, we use the volunteers' navigation trajectories and virtual trajectories to perform the experiment. We obtain 2,804 stay points from these trajectories, and 253 clusters are obtained ultimately. To prevent the test set containing too little data, we use the 5-fold cross validation to calculate the effect of destination marker selection. Destination marker selecting consist of two steps, detecting location type and selecting significant landmark. As the selection of significant landmark utilizes existing algorithm, we only test the effectiveness of location type detection.

We first compare location type detecting method only utilizing spatio-temporal features. We compare different variations of our algorithm (XGBoost, XGBoost+HMM, XGBoost+LSTM) with baselines (SVM and random forest). Figure 6(d)-6(f) show the precision, recall and F1 score ($F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$) of the labeling results in different cases. It can be seen that all XGBoost methods have higher precision and F1 in most cases. Thus it indicates that XGBoost is more suitable in this application scenario. This is because the tree-based models are usually superior in dealing with missing values in features and historical trajectory data is sparse and of low-quality. Furthermore, XGBoost+LSTM outperforms others in most cases with precisions consistently 70%. For the type of 'home' and 'work', its precision can reach 90%. This is because the forward and backward LSTM can well capture the contextual information.

Feature importance: Figure 7(a) plots the top eight most important features for location type detecting, by computing how many times a feature is used for dividing the training samples by XGBoost. As shown, all the top three are temporal features. It is expected that $frequency$ can reflect the importance of a location to the user, while $duration$ and $arrive\ time$ can reveal the user's living habits in this place. As spatial features, the TF-IDF values of $restaurant$ and $residence$ can be used to infer the user's possible activities here, so as to assist the type detection in making the correct judgment. The remaining three features are also useful indicators, providing considerable weight for place classification.

6.3.3 Efficiency Evaluation. We also evaluate the time cost of our route description algorithm, which is especially important for online description systems. The time cost mainly depends on the size (number of landmarks) of the given route. Thus we tune the size of the given route, and record the average time cost for describing a single route. The result is shown in Figure 7(b), from which it is observed that the route can be described within tens of milliseconds. With the increase of the size, the time cost increases slightly.

6.3.4 Subjective Experiments. Based on users' historical trajectories, our personalized route summarization aims at providing them with a more intuitive view of their trips. Thus, to make a comparison between the personalized route description and turn-by-turn description manner, we perform a subjective experiment where we choose 20 volunteers residing in Chengdu, then generate their historical trajectories and label the personally semantic places with the assistance of routing engine. It is noteworthy that during the experiment, volunteers use the personalized route description and turn-by-turn route description manner simultaneously, and their

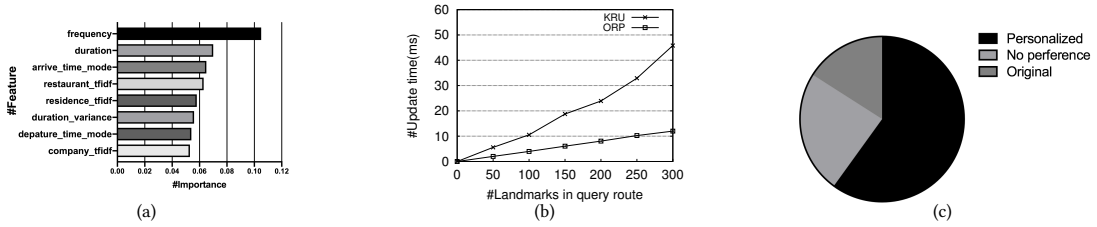


Figure 7: (a) Importance of different features in location type detecting. (b) Time cost. (c) User preference.

preferences ("personalized route description", "traditional turn-by-turn description manner" or "no strong preference") will be recorded. According to these records, as Figure 7(c) shows, we can find out that a majority of the volunteers prefer our personalized route description, while only 15% volunteers prefer the original ones, and about 25% volunteers have no preference. This experiment result suggests the tremendous convenience brought by the personalized route description system in guiding the users to their destination so that percolating to most volunteers' satisfaction. It's because compared with turn-by-turn instructions, using personally semantic places and POIs are more intuitive for route description.

7 CONCLUSIONS

In this paper we have proposed a new personalized route description system based on users' historical trajectory data is proposed, which is capable to make navigation instructions more customized and laconic. The system first extracts knowledge trees from historical trajectory, then uses these knowledge trees to partition a given route, select a proper destination marker for each partition and finally generates the corresponding route description. We have conducted extensive experiments on three real trajectories data and the user's real experience. As demonstrated by the experimental results, in most cases, compared with traditional turn-by-turn route description, the system framework is effective in providing users with clearer and better route description experience.

ACKNOWLEDGMENTS

This research is supported by the NSFC (Grant No. 61802054, 61972069, 61836007, 61832017, 61532018, 61902134), and the Central Universities (UESTC: Grants No: ZYGX2016KYQD135, HUST: Grants No. 2019kfyXKJC021, 2019kfyXJJS091).

REFERENCES

[1] S. Andrae and S. Winter. Summarizing gps trajectories by salient patterns. In *Angewandte Geoinformatik*, 2005.

[2] K.-P. Chang, L.-Y. Wei, M.-Y. Yeh, and W.-C. Peng. Discovering personalized routes from trajectories. In *ACM SIGSPATIAL Workshop on Location-Based Social Networks*, pages 33–40. ACM, 2011.

[3] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.

[4] Z. Chen, H. T. Shen, and X. Zhou. Discovering popular routes from trajectories. In *ICDE*, pages 900–911. IEEE, 2011.

[5] D. Delling, A. V. Goldberg, M. Goldszmidt, J. Krumm, K. Talwar, and R. F. Werneck. Navigation made personal: Inferring driving preferences from gps traces. In *ACM GIS*. ACM, 2015.

[6] D. E. D'Informatique, N. Ese, P. Esent, E. Au, F. Gers, P. R. Hersch, P. Esident, and P. P. Frasconi. Long short-term memory in recurrent neural networks. *Epfl*, 9(8):1735 – 1780, 2001.

[7] M. R. Evans, D. Oliver, S. Shekhar, and F. Harvey. Summarizing trajectories into k-primary corridors: a summary of results. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 454–457. ACM, 2012.

[8] Y. Huang, R. Jin, F. Bastani, and X. S. Wang. Large scale real-time ridesharing with service guarantee on road networks. *Proceedings of the Vldb Endowment*, 7(14), 2013.

[9] G. Kellaris, N. Pelekis, and Y. Theodoridis. Trajectory compression under network constraints. In *Advances in Spatial and Temporal Databases*, pages 392–398. Springer, 2009.

[10] Y. Li, H. Su, U. Demiryurek, B. Zheng, T. He, and C. Shahabi. Pare: A system for personalized route guidance. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 637–646, Republic and Canton of Geneva, Switzerland, 2017. International World Wide Web Conferences Steering Committee.

[11] K. Liu, B. Yang, S. Shang, Y. Li, and Z. Ding. MOIR/uots: Trip recommendation with user oriented trajectory search. In *Mobile Data Management (MDM), 2013 IEEE 14th International Conference on*, volume 1, pages 335–337. IEEE, 2013.

[12] D. Luxen and C. Vetter. Real-time routing with openstreetmap data. In *ACM Sigspatial International Conference on Advances in Geographic Information Systems*, pages 513–516, 2011.

[13] M. Lv, L. Chen, Z. Xu, Y. Li, and G. Chen. The discovery of personally semantic places based on trajectory data mining. *Neurocomputing*, 173:1142–1153, 2016.

[14] K.-F. Richter, F. Schmid, and P. Laube. Semantic trajectory compression: Representing urban movement in a nutshell. *Journal of Spatial Information Science*, 2012(4):3–30, 2012.

[15] R. Song, W. Sun, B. Zheng, and Y. Zheng. Press: A novel framework of trajectory compression in road networks. *Proceedings of the VLDB Endowment*, 7(9):661–672, 2014.

[16] H. Su, K. Zheng, H. Wang, J. Huang, and X. Zhou. Calibrating trajectory data for similarity-based analysis. In *SIGMOD*, pages 833–844. ACM, 2013.

[17] H. Su, K. Zheng, K. Zeng, J. Huang, S. Sadiq, N. J. Yuan, and X. Zhou. Making sense of trajectory data: A partition-and-summarization approach. In *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*, pages 963–974. IEEE, 2015.

[18] H. Su, K. Zheng, K. Zeng, J. Huang, and X. Zhou. STMaker: a system to make sense of trajectory data. *VLDB*, 7(13):1701–1704, 2014.

[19] H. Yoon, Y. Zheng, X. Xie, and W. Woo. Social itinerary recommendation from user-generated digital trails. *Personal and Ubiquitous Computing*, 16(5):469–484, 2012.

[20] J. Yuan, Y. Zheng, X. Xie, and G. Sun. Driving with knowledge from the physical world. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 316–324. ACM, 2011.

[21] N. J. Yuan, Y. Zheng, X. Xie, Y. Wang, K. Zheng, and H. Xiong. Discovering urban functional zones using latent activity trajectories. *IEEE Transactions on Knowledge and Data Engineering*, 27(3):712–725, 2015.

[22] K. Zheng, S. Shang, N. J. Yuan, and Y. Yang. Towards efficient search for activity trajectories. In *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*, pages 230–241. IEEE, 2013.

[23] K. Zheng, Y. Zheng, N. J. Yuan, S. Shang, and X. Zhou. Online discovery of gathering patterns over trajectories. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1974–1988, 2014.

[24] Y. Zheng. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(3):29, 2015.

[25] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W. Y. Ma. Understanding mobility based on gps data. In *International Conference on Ubiquitous Computing*, pages 312–321, 2008.

[26] Y. Zheng, X. Xie, and W. Y. Ma. Geolife: A collaborative social networking service among user, location and trajectory. *Bulletin of the Technical Committee on Data Engineering*, 33(2):32–39, 2010.

[27] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining interesting locations and travel sequences from gps trajectories. In *World Wide Web*, pages 791–800. ACM, 2009.