Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 4488781, 13 pages
https://doi.org/10.1155/2021/4488781

Research Article

Efficient Semantic Enrichment Process for Spatiotemporal Trajectories

Bin Zhao,[1] **Mingyu Liu**,[1] **Jingjing Han**,[2] **Genlin Ji**,[1] **and Xintao Liu**[3]

[1]*School of Computer and Electronic Information/School of Artificial Intelligence, Nanjing Normal University, Nanjing, China*
[2]*Quality Assurance Office, Jiangsu Open University, Nanjing, China*
[3]*Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University, China*

Correspondence should be addressed to Bin Zhao; zhaobin@njnu.edu.cn and Jingjing Han; hanjj@jsou.edu.cn

Received 5 August 2021; Accepted 1 October 2021; Published 12 November 2021

Academic Editor: Fa Zhu

The increasing availability of location-acquisition technologies has enabled collecting large-scale spatiotemporal trajectories, from which we can derive semantic information in urban environments, including location, time, direction, speed, and point of interest. Such semantic information can give us a semantic interpretation of movement behaviors of moving objects. However, existing semantic enrichment process approaches, which can produce semantic trajectories, are generally time-consuming. In this paper, we propose an efficient semantic enrichment process framework to annotate spatiotemporal trajectories by using geographic and application domain knowledge. The framework mainly includes preannotated semantic trajectory storage phase, spatiotemporal similarity measurement phase, and semantic information matching phase. Having observed the common trajectories in the same geospatial object scenes, we propose a semantic information matching algorithm to match semantic information in preannotated semantic trajectories to new spatiotemporal trajectories. In order to improve the efficiency of this approach, we build a spatial index to enhance the preannotated semantic trajectories. Finally, the experimental results based on a real dataset demonstrate the effectiveness and efficiency of our proposed approaches.

1. Introduction

Spatiotemporal trajectories record the spatiotemporal position sequences of moving objects. The increasing access to positioning device technologies, such as smartphones, GPS-enabled cameras and sensors, results in vast volumes of collected spatiotemporal trajectories. Analyzing and mining spatiotemporal trajectories can study in depth various fields such as traffic coordination and management (e.g., road flow monitoring), tourist route recommendation, and natural disaster early warning (e.g., typhoon prediction). However, many applications in the mobility domain require a semantic interpretation of movement information. This semantic interpretation is usually obtained by mining semantic trajectories, which is the fusion of spatiotemporal trajectories and semantic information. Location-based social networks (LBSN), such as Twitter and Weibo, produce multifaceted semantic information, which contains the moving state of moving objects (e.g., speed and direction) and environment information (e.g., air temperature and spatial topological relationship) [1]. Combing semantic information, such as user's personalized characteristics, landmark names, user's interest, and occupation into the user's spatiotemporal trajectories, will contribute to the recommendation of nearby hot spots of interest for users [2, 3]. It can be seen that mining semantic trajectories [4] can better meet the needs of decision analysis applications.

Different from spatiotemporal trajectories obtained by position-aware devices, semantic trajectories must be generated through semantic trajectory modeling. Semantic trajectory modeling includes trajectory data preprocessing, trajectory segmentation, and semantic enrichment. Among them, the semantic enrichment process is the key stage, which annotates appropriate semantic information (e.g., behavior attributes, environment information, and domain knowledge) in spatiotemporal trajectories. With different

sources, complex types, and diverse forms of semantic information, there are different semantic enrichment process approaches.

The existing semantic enrichment process approaches can be divided into three categories: (1) Early approaches directly annotate velocity and direction in spatiotemporal trajectories. Due to lacking rich semantic information, the results of mining semantic trajectories annotated by early approaches have a low semantic interpretation. (2) Part approaches annotate domain knowledge in spatiotemporal trajectories through ontology. However, approaches based on ontology transform a semantic trajectory into RDF graph description [5], which causes the finding and reasoning semantic trajectories time-consuming. (3) Typical approaches annotate geographical object information, including areas of interest (ROIs), lines of interest (LOIs), and points of interest (POIs), through the *spatial join* [6] algorithm and *map matching* [7] algorithm. The execution time of [6, 7] is linearly correlated with the number of geospatial objects, which results in high time consumption. It can be seen that the existing semantic enrichment process approaches have the disadvantage of high time consumption.

On the other hand, given movement trajectories limited by topological relationship of urban road networks, there are common movement trajectories in the same geospatial object scenes. For example, commuters departing from the Tsinghua Park residential usually take Metro Line 4 to Beijing Zhongguancun SOHO Building. Due to traffic restrictions, it is easy to collect a large number of identical commuting trajectories. Obviously, new commuting trajectory information can be directly attached to historical commuting trajectories. Similarly, it is possible to directly annotate the semantic information in a preannotated semantic trajectory to new spatiotemporal trajectories. Using preannotated semantic trajectories for enrichment does not need a complicated computation and annotation process, which may avoid an inefficient semantic enrichment process.

In this paper, we propose a new semantic enrichment process approach named Efficient Semantic Enrichment Process for Spatiotemporal Trajectories based on Semantic Information Matching (SEPSIM), which firstly uses semantic information in preannotated semantic trajectories for annotating spatiotemporal trajectories. We first store preannotated semantic trajectories in the form of episodes. In this phase, we segment semantic trajectories into stop or move episodes. Then, we measure the spatiotemporal similarity between subtrajectories and episodes. The similarity of stop subtrajectories and move subtrajectories is measured, respectively. Finally, we propose a new algorithm named Semantic Information Matching Algorithm based on Similar Episodes (SESIM), which can match semantic information of episodes to a new trajectory. In order to put down the search cost of metrics and matching, we build a spatial index to store episodes of preannotated semantic trajectories.

In summary, this article makes the following contributions:

(i) We propose an efficient semantic enrichment process framework (SEPSIM) for spatiotemporal trajectories based on semantic information matching. It includes three phases: preannotated semantic trajectory storage, spatiotemporal similarity measurement, and semantic information matching. In order to improve the efficiency of the SEPSIM approach, we establish a spatial index

(ii) We propose a new standard to measure the effectiveness of semantic enrichment process approaches. Also, we compared different semantic enrichment process approaches in efficiency

(iii) In order to verify the effectiveness and efficiency of the SEPSIM approach, experiments were performed by using the real trajectory dataset. The results prove the high effectiveness and efficiency of the SEPSIM approach

2. Related Work

There are different semantic enrichment process approaches with different sources, complex types, and various forms of semantic information. Early semantic enrichment process approaches directly annotate velocity and direction in spatiotemporal trajectories, which generate semantic trajectories as stop and move subtrajectory sequences. Ashbrook and Starner [8] calculated the moving speed (whether the speed is zero) to identify stop subtrajectories. Due to poor speed measurement and other reasons, semantic trajectory stop segments do not match actual situation. Krumm and Horvitz [9] calculated the speed and direction to identify stop subtrajectories; Palma et al. [10] set the subtrajectory below the average speed as a stop subtrajectory, generating the semantic trajectory consisting of stop and move subtrajectories. In addition to calculating the moving speed, Zheng et al. [11] also calculated the acceleration and speed change rate to discover move subtrajectories with different modes of transportation (e.g., bicycles, buses, and self-driving) to enrich the semantic trajectory. Although early semantic enrichment process approaches were fast in annotation, the semantic information was not rich enough.

Part semantic enrichment process approaches annotate domain knowledge as semantic information through ontology. Spaccapietra et al. [12] first proposed an ontological method for semantic trajectory modeling. Based on the concepts of "stop" and "move," the ontology was used to define semantic trajectories, and the semantic information of trajectories was further enhanced using the reasoning ability of ontology. Baglioni et al. [13] extended the definition of Baglioni's ontology and proposed the concept of core ontology, which formally describes the concepts of stop, move, time, place, and mode in human mobile behavior, further enriching the definition of semantic trajectories. In 2014, Vandecasteele et al. [14] combined semantic trajectories with semantic events. Nogueira et al. [15] proposed the QualiTraj ontology to describe the various motion characteristics of original trajectories, especially the derivative characteristics, such as speed, acceleration, and direction. Nogueira

and Martin [16] proposed a new ontology based on QualiTraj ontology with stronger information description ability, namely, Semantic Trajectory Episodes (STEP) ontology. It can not only describe basic motion characteristics but also describe environmental characteristics of moving trajectories on a higher semantic level. In 2018, Nogueira et al. [17] proposed the FrameSTEP, a semantic trajectory labeling framework based on STEP ontology. This method can calculate various physical movements and spatial geometric features of trajectory segments and use external reliable resources (such as OSM and LinkedGeoData geographic knowledge base) to label the environmental features of trajectories. However, approaches based on ontology need to represent semantic trajectories as RDF graphical descriptions, which results in time consumption.

The main source of information on semantic enrichment is geospatial objects with geometric features in geographical objects, including regions of interest (ROI), lines of interest (LOI), and points of interest (POI) [18]. At present, the typical semantic enrichment processing method uses the spatial join algorithm [6] to find the regions of interest (ROI) that have a topological relationship with spatiotemporal trajectories and label the regions of interest associated with spatiotemporal trajectories and the corresponding topological relationship. This algorithm needs to combine the external environment information (e.g., OSM map and Baidu map) to select the regions of interest associated with spatiotemporal trajectories. The execution time of the algorithm is linearly related to the number of geospatial objects, resulting in high time complexity and low semantic enrichment performance in the spatial connection process. For points of interest (POI), Sun et al. used an implicit Markov model [19] to label the POI categories for staying segments of spatiotemporal trajectories, but in the regions with intensive POI, staying segments may be related to multiple interest points. Coupled with the low GPS sampling rate, it is difficult to identify effective POIs. On the other hand, the LOI labeling method often uses a global map matching algorithm [7] to determine the location of spatiotemporal trajectories. Parent et al. proposed a "point-segment distance" measurement method [7] to replace the original distance function in the global map matching algorithm, which is suitable for labeling lines of interest in geographical scenarios such as dense road networks, parallel roads, and intersections. The global matching algorithm needs to perform metric matching on trajectory segments where spatiotemporal trajectories are located, which easily results in high time complexity of algorithm execution and low semantic enrichment performance.

3. Preliminaries

In this section, we will present definitions of all necessary concepts used in this paper and formally state the problem.

3.1. Basic Concepts. The SEPSIM approach proposed in this paper is aimed at annotating semantic information of preannotated semantic trajectories in spatiotemporal trajectories. The input of this problem is a trajectory, short for a spatiotemporal trajectory. Thus, we provide the definition of "trajectory" at first.

Definition 1 (trajectory). A trajectory T is a sequence of sampling points in the form $T = \{p_1, p_2, \cdots p_{|T|}\}$, $p_i = (\text{tid}, x_i, y_i, t_i)$, where tid is an object identifier and x, y and t are spatial coordinates and a time stamp, respectively. $|T|$ records the number of sampling points in trajectory T.

Definition 2 (subtrajectory). A subtrajectory is a substring of a trajectory, i.e., $T_s = \{p_{i+1}, p_{i+2}, p_{i+3}, \cdots, p_{i+m}\}$, where $0 \le i \le |T| - m, m \ge 0$.

Definition 3 (stop subtrajectory and move subtrajectory). Given the distance threshold ε and the number of point threshold minpts, a DBSCAN cluster [20] analyzes the trajectory T. Each cluster is a stop subtrajectory of the trajectory. If each p_i in $T_s = \{p_{i+1}, p_{i+2}, p_{i+3}, \cdots, p_{i+m}\}$ is an outlier, T_s is a stop subtrajectory (stop T_s). If point p_i is in the end of a stop subtrajectory and point p_{i+m+1} is in the beginning of another stop subtrajectory, $i + m < |T|$, T_s is a move subtrajectory (move T_s).

Then, we define "semantic trajectory" as the output of this problem. The main source of information on semantic enrichment is geospatial objects in geographical environment. For this reason, the semantic information matching in this paper refers to geospatial object information matching. First, we give the basic related to semantic information.

Definition 4 (geospatial object). According to geometric shapes, geographical objects are divided into three categories: region of interest (ROI), line of interest (LOI), and point of interest (POI). In this paper, we refer to ROIs, LOIs, and POIs collectively as geospatial objects. A geospatial object Go is defined as a uniquely identified specific space site (e.g., a park, a road, or a cinema). A Go is a quad (id, cat, loc, con), where id represents a geospatial object identifier and cat denotes the category of it (e.g., ROI, LOI, and POI), and loc denotes its corresponding location attribute in terms of longitude and latitude coordinates and con denotes its name.

Definition 5 (topological relation). For different types of geospatial objects, the topological relationship between subtrajectory T_s and the geospatial object Go is defined as the following seven types: T_s pass by Go (Go is a LOI), T_s pass by Go (Go is a POI), T_s pass by Go (Go is a ROI), T_s across Go (Go is a ROI), T_s enter Go (Go is a ROI), T_s leave Go (Go is a ROI), and T_s stop inside Go (Go is a ROI).

Definition 6 (episode). An episode [21] is a subtrajectory of semantically homogeneous sections of a trajectory, such as move episode and stop episode. We define an episode as a multilayered semantic sequence aligned in accordance with the time of a subtrajectory, i.e., episode = (T_s, sp, dir, geoinf), where T_s denotes the episode corresponding to trajectory segments, sp denotes the average speed of an episode, dir denotes the direction of an episode, and geoinf denotes the

episode corresponding geospatial information. The form of a specific episode is shown in Figure 1.

Definition 7 (semantic trajectory). A semantic trajectory ST is a sequence of episodes in a spatiotemporal order of a moving object, i.e., $ST = \{episode_1, episode_2, \cdots, episode_{|ST|}\}$.

The list of major symbols and notations in this paper is summarized in Table 1.

3.2. Problem Statement. Given a trajectory T, a preannotated semantic trajectory dataset OST, two clustering thresholds ε and minpts, four radii r_1, r_2, r_3, r_4, and a similarity threshold σ, our goal is to annotate semantic information of preannotated semantic trajectories in trajectory T, which can transform trajectory T to semantic trajectory ST.

4. Framework

In this section, we will present the SEPSIM framework including preannotated semantic trajectory storage phase, spatiotemporal similarity measurement phase, and semantic information matching phase. Figure 2 outlines this framework.

Preannotated Semantic Trajectory Storage. Given the preannotated semantic trajectory dataset OST, the first step is to store them. In order to prevent reducing the semantic information matching accuracy, preannotated semantic trajectories are stored in the form of episodes, which are representative and diverse. Semantic trajectories are segmented into episodes by the moving state (stop/move) of the moving object. The output of this phase is a set of stop episodes and move episodes, which can represent and describe a certain region.

Spatiotemporal Similarity Measurement. Given a trajectory T, the spatial-temporal similarity is measured between T and episodes obtained in the first phase. We first segment trajectory T into stop/move subtrajectories by DBSCAN clustering. Then, there are two subproblems that need to be solved: how to measure the similarity between the stop subtrajectory and stop episode and how to measure the similarity between the move subtrajectory and move episode. To solve the problem above, we propose the algorithms based on the Hausdorff distance [22] and based on the Longest Common Subsequence (LCS) [23], respectively. The output of this phase is stop and move episodes, which satisfy the specified similarity condition.

Semantic Information Matching. Semantic information of similar stop/move episodes is matched to trajectory Tin this phase, through the proposed semantic information matching algorithm (SESIM). The algorithm consists of two subphases: candidate episode sorting and semantic information mapping. We aim to generate a semantic trajectory ST that contains the most semantic information. For part subtrajectories which have no matching information, we complete the semantic enrichment process of ST by using the typical approach.

4.1. Preannotated Semantic Trajectory Storage. After we get the preannotated semantic trajectory dataset OST, the first task is to store them for the matching phase. Storing all preannotated semantic trajectories can reduce the workload of storage and search, but the effectiveness and efficiency of the matching phase between complete trajectories are poor. And storing complete preannotated semantic trajectories with corresponding episodes causes data redundancy. In order to ensure complete semantic information and avoid data redundancy, we choose to store preannotated semantic trajectories in forms of episodes. However, episodes can only be obtained through trajectory segmentation. There are two kinds of trajectory segmentation methods: segment according to geospatial objects and segment according to the moving state of the moving object. With complex and irregular distribution and a large number of geospatial objects, segmentation according to geospatial objects is easy to cause trajectory fragments and time consumption. Meanwhile, segmentation according to the moving state of moving objects has the advantages of high segmentation efficiency and clear segmentation rules. So, we choose to segment spatiotemporal trajectories by the moving state of moving objects. For the reason that the stop of the moving object produces trajectory point gathering, we segment preannotated semantic trajectories into stop/move episodes by DBSCAN clustering.

Given a preannotated semantic trajectory dataset OST and a new coming preannotated episode, there are three situations to compare with the episodes in the dataset OST. The first case is the newly episode not repeated in the dataset OST at all, the second case is partial repetition but not complete repetition compared with the dataset OST, and the third case is complete repetition. If all preannotated episodes were stored, it will cause querying multiple repeated episodes with the increasing dataset, which reduces the efficiency of the similarity measurement and matching phase. Therefore, there is a challenge: which preannotated episodes stored can guarantee to avoid redundancy and ensure the effectiveness and efficiency of matching.

To solve the challenge above, we choose to store representative and diverse preannotated episodes to build the dataset OST. The semantic information of the semantic episodes (spatial information and geospatial environment information) represents the geospatial environment characteristics of a certain region. Therefore, the representative semantic episode of a certain region is defined as the episode with the same or partial spatial information and incomplete semantic information compared with preannotated episodes in the set semantic trajectories OST. The diversity of episodes is reflected in the diversity of geospatial environment information, which can enrich the characteristics of a certain region. So, we define the diverse episode as an episode with new geospatial environment information compared with preannotated episodes in the dataset OST. In this paper, the representative and diverse episodes are obtained through trajectory classification in Figure 3. For a given preannotated episode dataset, we first classify it according to spatial information and then classify it according to geospatial information and topological relationship, and finally, the leaf nodes store fine-grained representative and diverse preannotated episodes for matching. The

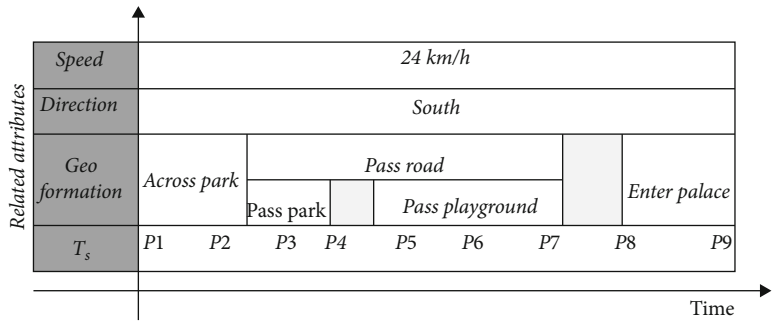


Figure 1: Example of an episode.

Table 1: Table of notations.

Notations	Definition
T	A spatio-temporal trajectory
T_s	A subtrajectory of T
stop T_s	A stop subtrajectory of the trajectory T
move T_s	A move subtrajectory of the trajectory T
Go	A geospatial object
episode	A subtrajectory of a semantic trajectory
stop episode	A stop subtrajectory of a semantic trajectory
move episode	A move subtrajectory of a semantic trajectory
ST	A semantic trajectory
OST $\{ST_1, ST_2, \cdots\}$	The set of semantic trajectories
ε	The DBSCAN clustering distance threshold
minpts	The DBSCAN clustering point number threshold
r_1, r_2, r_3, r_4	Four similar region radii
σ	The similarity threshold
stop $T_{s\,set}$	The set of stop T_s of T
move $T_{s\,set}$	The set of move T_s of T
stop Episode_{Set}	The set of stop episodes of OST
move Episode_{Set}	The set of move episodes of OST

output of this phase is a set of representative and diverse stop/move episodes of the set semantic trajectory OST, which represent a certain region.

4.2. Spatiotemporal Similarity Measurement. For an incoming trajectory T, we compare it with episodes to find similar episodes. Once we find the similar episodes, we can match the semantic information of episodes to trajectory T. Giving the limitation of topological relationship of urban road networks, there are many similar or the same trajectory segments. So, we first segment trajectory T into stop/move T_s by DBSCAN clustering. Then, we solve the two problems: the similarity between stop subtrajectory and stop episode (stop trajectories) measurements and the similarity between move subtrajectory and move episode (move trajectories) measurements. Next, we will discuss the algorithm to solve these two problems, respectively, in the following algorithms.

The Algorithm to Determine the Similarity between Stop Trajectories. To our knowledge, there is no basic method for measuring the similarity of stop trajectories in the Euclidean space. In this paper, the stop T_s and stop episode are clusters of trajectory points obtained by DBSCAN clustering. The similarity measurement of the stop T_s and stop episode can be regarded as similarity measurement of point sets. Therefore, we view each stop trajectory, which is a stop T_s or a stop episode, as point sets. The algorithm proposed in this paper consists of two steps: (1) similar region determination and (2) similarity measurement based on the Hausdorff distance. Given the fact that the closer the space, the more similar the trajectories, we first narrow the metric range of stop episodes down and remain stop T_s with greater likelihood of similarity. Then, we calculate the Hausdorff distance between each stop T_s in stop $T_{s\,set}$ of T and stop episodes in stop Episode_{Set} of OST sequentially. Finally, stop episodes meeting similar conditions are remained.

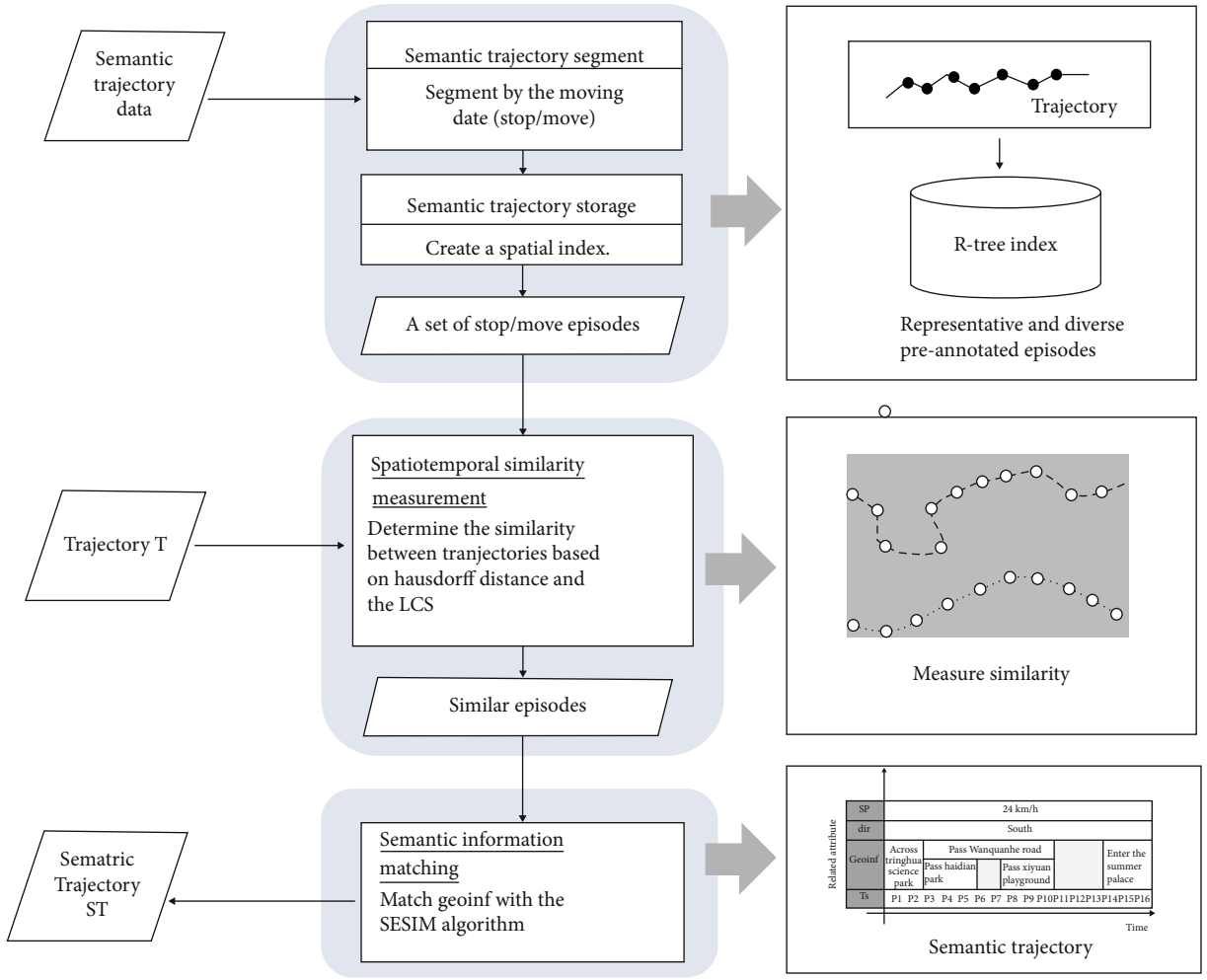


Figure 2: The framework of the SEPSIM process.

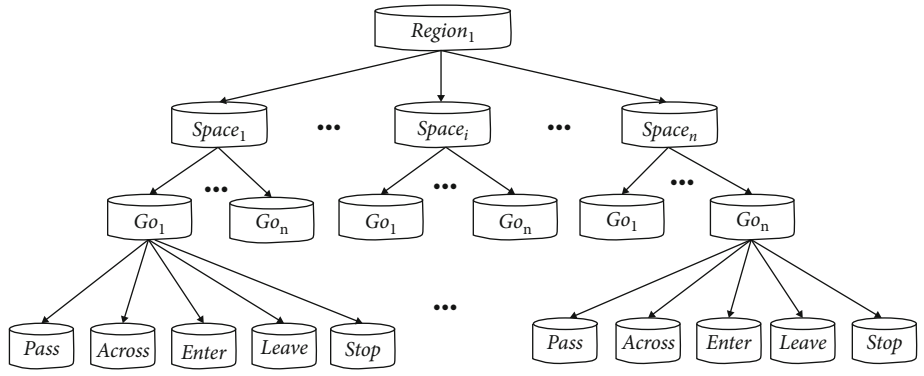


Figure 3: Principle of the trajectory classification.

In the first step, we narrow the number of stop episodes down and remain stop episodes with high similar probability to each stop T_s of T. Firstly, we convert each stop T_s to a point set P by assigning the latitude and longitude coordinates of each stop T_s to the x, y coordinates of the point set O (lines 1-5). According to the minimum circumscribed point o in point set P and the given radius r_1, we draw a circular area Circle_1 as the similar region of the stop T_s (line 6). All the stop episodes that intersect with or are inside Circle_1 are extracted for similarity measurement. If there are no stop episodes in a similar region, there is no similar stop episode to the stop T_s. Otherwise, we convert stop episodes extracted in a similar region to point sets $E_{\text{set}}(E_1(\text{stop Episode}_{1,}), \cdots, E_n(\text{stop Episode}_{n,}))$ in the second step

(lines 7-10). Figure 4 shows the similar region determination of each stop T_s.

Then, we calculate the Hausdorff distance between P and each point set E_i in a similar region (lines 11-13). Finally, the point set E_i, which has the minimum Hausdorff distance to point set P, was returned. The stop episode corresponding to the point set E_i is the most similar episode to the stop T_s (lines 14-16).

The Algorithm to Determine the Similarity between Move Trajectories. Generally, move episodes are not completely similar to the entire subtrajectory. In academia, this kind of similarity measurement is called the local matching of the trajectories. Existing local matching methods include the Frechet distance [24], Longest Common Subsequence (LCS) [23], and K Best Connected Trajectories [25]. The Frechet distance method is sensitive to a noise trajectory point; the K Best Connected Trajectory method can only query a few elements and is mainly used for recommending tourist routes. The Longest Common Subsequence (LCS) method is different from the previous similarity measurement methods. The previous methods focus on calculating the distance between point pairs of trajectories. The LCS method takes into account the movement of vehicles, which is restricted by the road network. If vehicles travel on the same road segment, the trajectories passing through the road segment may completely overlap, which is consistent with the thought of the SEPSIM approach. Therefore, the degree of overlap between trajectories can be used as a criterion for similarity.

The LCS method is only suitable for trajectory data generated on the road network, and the time complexity is $O(m * n)$. However, the LCS method has the advantage of not considering departure time and driving speed of trajectories and is robust to noise, which is consistent with the situation of the experimental data in this paper. Therefore, we propose the algorithm to determine the similarity between move trajectories based on the LCS. The detail of the LCS method can be found in [23].

This algorithm consists of three steps: (1) similar region determination, (2) measurement range determination, and (3) similarity measurement based on LCS. First, we filter move episodes that are likely similar in each move T_s similar region [26]. Then, the subtrajectory part of the move episode that is similar to T is determined. Finally, we calculate the similarity between move episodes and the corresponding similar subtrajectory of T based on the LCS method. The long common subsequence obtains the similarity and retains move episodes that meet the similarity threshold. The same operation is performed on each move T_s.

We use the same way to draw the similar region of each move T_s in move $T_{s\,set}$ of T. In the first step, we draw a circular area Circle_2 with a given radius r_2 and a circle point o, which is the center of each move T_s, as the similar region of each move T_s (lines 1 and 2). Each move episode that intersects with or is inside the circle is extracted for measurement range determination, which is the candidate move episode set E_{set} (lines 3-5). For each move episode in a similar region, we draw two circular areas Circle_3 and Circle_4 with the given

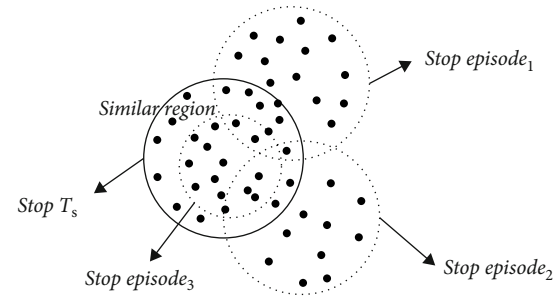


Figure 4: Similar region determination of each stop T_s of T.

radii r_3 and r_4 and two circle points, which are the beginning and end point of each move episode (lines 6-9). Given that the trajectories are partially similar, we then confirm the measurement ranges of trajectory T, where each move episode measures the similarity. The part of trajectory T, which is tangent to the two circles Circle_3 and Circle_4, is the measurement range corresponding to each move episode. Figure 5 shows similar region determination and measurement range of each stop T_s of T.

In the third step, we calculate the similarity simSeq(move episode, move T_s) based on the Longest Common Subsequence (LCS) method (lines 9 and 10). If the simSeq is greater than or equal to the given similarity threshold σ, the move episode is similar to the part trajectory T. We remain the move episodes as simMove $\text{Episode}_{\text{Set}}$, which meet the similarity threshold (lines 11-13).

4.3. Semantic Information Matching. In this phase, we aim to match semantic information of episodes $\text{Episode}_{\text{Set}}$ remained in the spatiotemporal similarity measurement phase to the trajectory T. The simStop $\text{Episode}_{\text{Set}}$ remained are the most similar ones corresponding to the part trajectory T, and all the similarities of simMove $\text{Episode}_{\text{Set}}$ are greater than or equal to 95%, which are identical to T in spatial information. Given a trajectory T and a set of similar episodes $\text{Episode}_{\text{Set}}$, the episodes corresponding to T have the following three matching ways shown in Figure 6. Obviously, there is a problem that needs to be solved: how to determine if the selected episodes are the best combination in the similar episode set for matching T to ST, which has the most semantic information.

To solve the problem, we propose a Semantic Information Matching Algorithm based on Similar Episodes (SESIM). This algorithm consists of two steps: (1) similar episode sorting and (2) semantic information matching. According to measurement range determination in the second phase, we first sort similar episodes meeting similar conditions by the spatial coordinate sequence of the trajectory T. Then, we model the problem as a knapsack problem to match semantic information.

Similar Episode Sorting. Given a trajectory T and a set of similar episodes simStop $\text{Episode}_{\text{Set}}$, we first measure the similar range of the trajectory T corresponding to similar episodes with the same solution in the step of measurement range determination. In this step, we convert the set of

Input : $stop\ T_{s\,set}$, $stop\ Episode_{Set}$, r_1
Output : $simStop\ Episode_{Set}$
1 **for** each $stop\ T_s \in stop\ T_{s\,set}$ **do**
2 **for** each $P_i(x, y) \in stop\ T_s$ **do**
3 $O_i(x, y) \longleftarrow P_i(x, y)$;
4 $i++$
5 $Insert\ O_i\ into\ O_{set}(stop\ T_s)$;
6 $Circle\ c = \text{minCircle}(O_{set}(stop\ T_s), r_1)$;
7 **for** each $stop\ Episode_i \in stop\ Episode_{Set}$ **do**
8 **if** $stop\ Episode_i$ in or insert $Circle\ c$ **then**
9 $E_i(stop\ Episode_i) \longleftarrow \text{EpisodeTransferPoint}(stop\ Episode_i)$
10 $Insert\ E_i\ into\ E_{set}(E_1(stop\ Episode_{1,}), \cdots, E_n(stop\ Episode_{n,})$;
11 **for** each $E_i(stop\ Episode_i) \in E_{set}$ **do**
12 $\text{distance}(stop\ Episode_i) \longleftarrow \text{Hausdorff}(stop\ Episode_i, stop\ T_s)$
13 $insert\ \text{distance}(stop\ Episode_i)\ into\ distance_{set}$;
14 **for** each $\text{distance}(stop\ Episode_i) \in distance_{set}$ **do**
15 $simStop\ Episode_i = MinDistance(\text{distance}(stop\ Episode_i))$;
16 return $simStop\ Episode_{Set}$;

Algorithm 1: Similarity measurement of the stop trajectory (SMST).

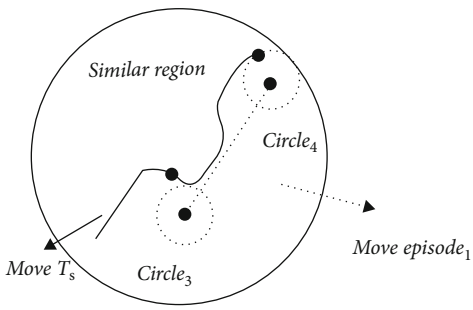


Figure 5: Similar region determination and measurement range of each stop T_s of T.

similar episodes to the candidate set $\text{Episode}_i(\text{Episode}_i, (P_{\text{begin}}, P_{\text{end}}), V(i), L(i))$, where P_{begin} and P_{end} are the beginning and end trajectory points of subtrajectory T_s, respectively, corresponding to $\text{simStop Episode}_{\text{Set}}$, $L(i)$ is the number of sampling points in T_s, and $V(i)$ is the number of geospatial information in $\text{simStop Episode}_{\text{Set}}$ (lines 1-5). Then, we sort the set E by the position of P_{begin} in trajectory T (lines 6-10).

Semantic Information Matching. In this step, we aim to select the best combination of episodes in set E for matching the semantic trajectory with most semantic information. We extend a knapsack algorithm, considering the number of sampling points of the trajectory T as the capacity of the backpack W and the number of geospatial information in E as the value of the episode. In start matching from the end sampling point P_{end} of the trajectory T, we aim to maximize the total value of the entire backpack. Given the candidate set $E\{\text{Episode}_i(\text{Episode}_i, (P_{\text{begin}}, P_{\text{end}}), V(i), L(i))\}$, we define the value of the trajectory T using the following formula: $\text{SemScore}(|T|) = \text{Max}\ (\text{SemScore}(|T| - 1), \text{SemScore}(|T| - L(i)) + V(i))$ (lines 11-18).

4.4. Space Index Establishment. To quickly get the preannotated semantic episodes similar to trajectory T, we use the space attribute of trajectory data to establish a space index for saving and querying episodes quickly, which will improve the efficiency of the SEPSIM approach.

The establishment of the space index is related to the query target. The index in this section is used to query episodes similar to the trajectory T_s. Therefore, the elements stored in the space index should be trajectory edge data. The common space index includes R-tree index [27], quadtree index [28], and grid index [29]. The elements stored in the spatial index are episodes, which are essentially trajectory edge data. The quad-tree index is only adapted to query a trajectory point. The large number of unevenly distributed geospatial objects causes the grid index to be inefficient. Meanwhile, the R-tree index can be efficient in the unevenly distributed dataset in this paper by ensuring the balance of the tree. Therefore, we create and maintain an R-tree index for preannotated episodes. With this index, we can compare an incoming subtrajectory T_s with preannotated episodes in the index, which are inside or intersect with the subtrajectory T_s.

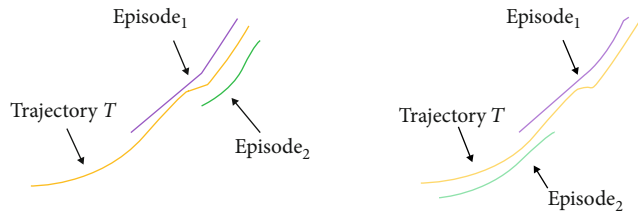
5. Experiments

In this section, we conduct extensive experiments on real trajectory datasets to compare the effectiveness and efficiency between the proposed approach SEPSIM in this paper and the typical approach based on the *spatial join* algorithm and *map matching* algorithm as the baseline approach.

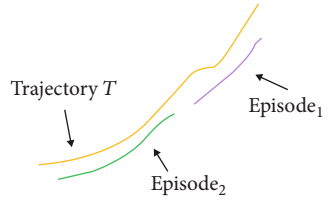
5.1. Experimental Settings. We evaluate our approach on the GeoLife dataset. This trajectory dataset was collected in (Microsoft Research Asia) GeoLife project by 182 users in a period of over five years (from April 2007 to August 2012), which contains 17,621 trajectories with a total distance of 1,292,951 kilometers and a total duration of

Input: $move\ T_{s\,set}$, $move\ Episode_{Set}$, d, r_2, r_3, r_4, σ
Output: $simMove\ Episode_{Set}$
1 **for** each $move\ T_s \in move\ T_{S\,set}$ **do**
2 $Circle\ C = \text{minCircle}(move\ T_s, r_2)$;
3 **for** each $move\ Episode_i \in \text{move}\ Episode_{Set}$ **do**
4 **if** $move\ Episode_i$ in or insert $Circle\ c$ **then**
5 $insert\ move\ Episode_i\ into\ E_{set}(move\ Episode_1, \cdots, move\ Episode_n)$;
6 **for** each $move\ Episode_i \in E_{set}$ **do**
7 $Circle\ C_1 = \text{minCircle}(move\ Episode_i, P_{begin}, r_3)$;
8 $Circle\ C_2 = \text{minCircle}(move\ Episode_i, P_{end}, r_4)$;
9 **if**$move\ T_S$ tangent $Circle\ C_1$ and $Circle\ C_2$ **then**
10 $SimSeq(move\ T_S, move\ Episode_i) = LCS(move\ T_S, move\ Episode_i)$;
11 **if**$SimSeq(move\ T_S, move\ Episode_i) \geq \sigma$ **then**
12 $insert\ move\ Episode_i\ into\ simMove\ Episode_{Set}$; ;
13 return $simMove\ Episode_{Set}$;

Algorithm 2: Similarity measurement of the move trajectory (SMMT).



(a) Similar episodes with large overlaps (b) Similar episodes with few overlaps

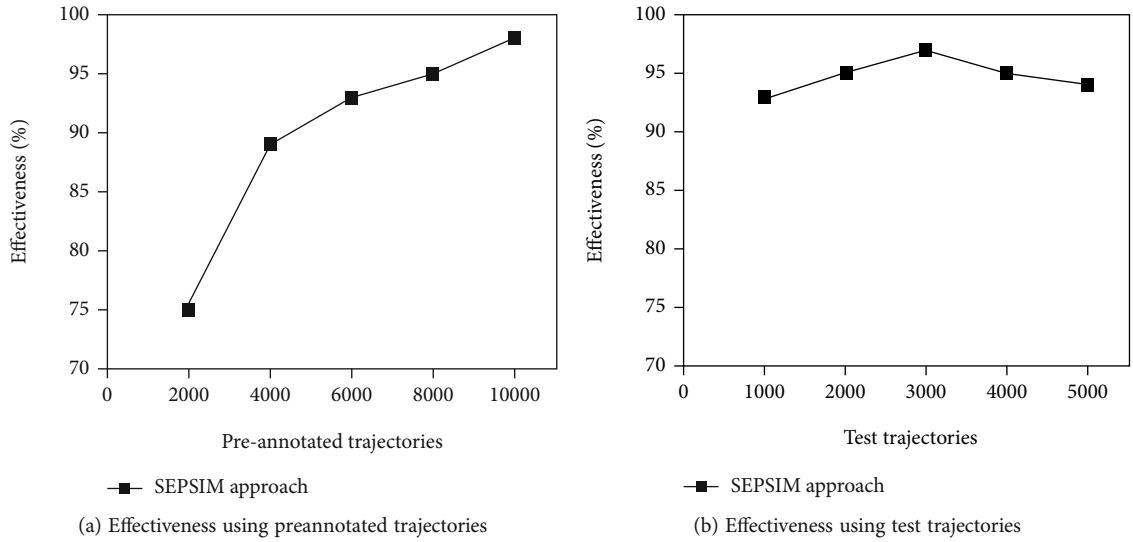


(c) Similar episodes with no overlaps

Figure 6: Tree matching types of similar episodes.

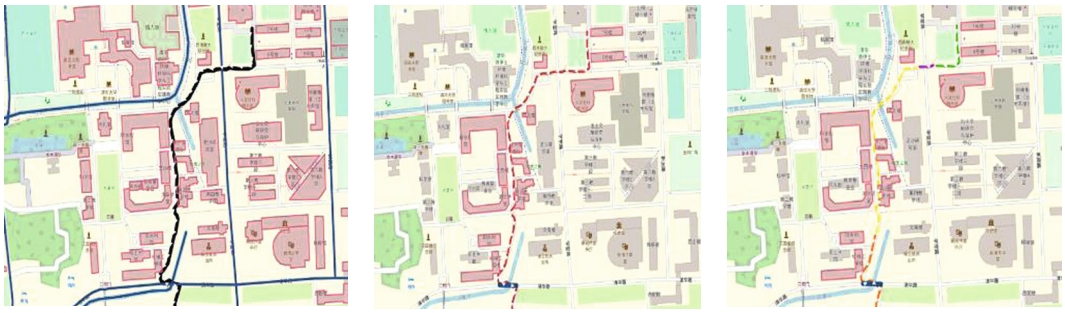
Input: T, $Episode_{Set}\{simStop\ Episode_{Set}, simMove\ Episode_{Set}\}$
Output: ST
1 **for** each $Episode_i \in Episode_{Set}$ **do**
2 $P_{begin}, P_{end} \longleftarrow MeasurementRangesDetermination\ (Episode_i, T)$;
3 $\text{V(i)} \longleftarrow GetGeoinfNumber(Episode_i)$;
4 $\text{L(i)} \longleftarrow GetNumberTrajPoint(Episode_i, T)$;
5 $insert\ P_{begin}, P_{end}, \text{V(i)}, \text{L(i)}\ into\ Episode_i(Episode_i, (P_{begin}, P_{end}), \text{V(i)}, \text{L(i)})$;
6 **for** each $Episode_i$ **do**
7 $Insert\ Episode_i(\ Episode_i, (P_{begin}, P_{end}), \text{V(i)}, \text{L(i)})into\ set\ E$;
8 $SortByTrajSpatial(E)$;
9 **for** $i = 0$ to $|T|$ **do**
10 $SemScore(0) = 0$;
11 **if**P_{end}in $Episode_i$ equal P_{end}in T **do**
12 $SemScore(|T|) = \text{Max}\ (SemScore(|T| - 1), SemScore(|T| - \text{L(i)} + \text{V(i)}))$
13 **for** each $Episode_i$ in $SemScore(|T|)$
14 $\text{ST} \longleftarrow MatchSemanticInf((Episode_i)$
15 return ST ;

Algorithm 3: Semantic information matching algorithm based on similar episodes (SIM).



(a) Effectiveness using preannotated trajectories (b) Effectiveness using test trajectories

Figure 7: Effectiveness of the SEPSIM approach.



(a) Trajectory in the OSM map (b) Result of the baseline approach (c) Result of the SEPSIM approach

Figure 8: Trajectory enrichment results using different approaches.

50,176 hours. These trajectories were recorded by different GPS loggers and GPS phones and have a variety of sampling rates. The majority of the data was created in Beijing, China, and the data size is 1.87 GB. In this paper, all the preannotated semantic trajectories are generated by the typical approach. Both algorithms are implemented in Java and on computers with Intel(R) Xeon(R) CPU E5-2620 (2.10 GHz) and 32 GB memory.

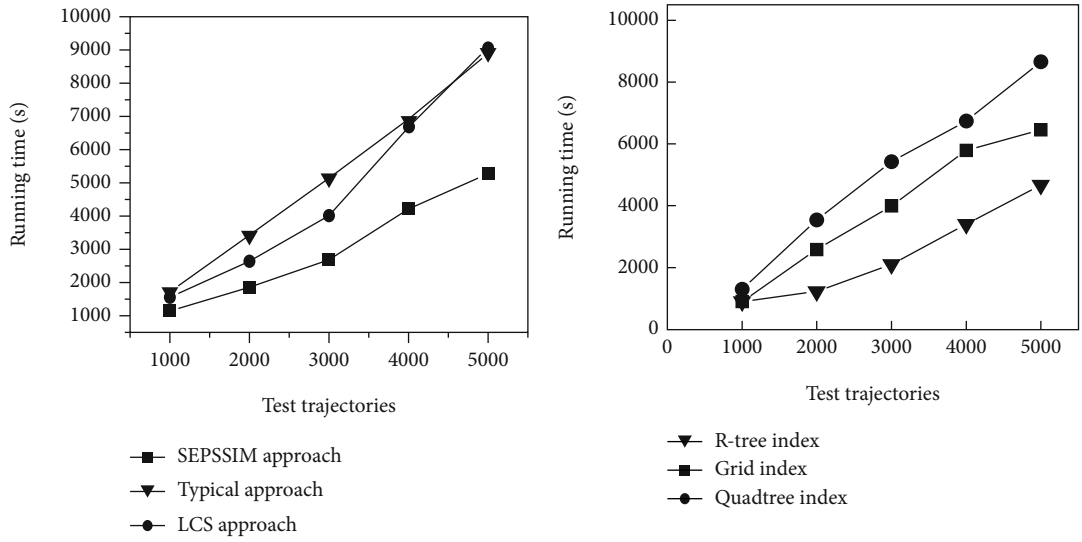
5.2. Effectiveness. There is no clear and unified definition for the effectiveness of the semantic enrichment process. In this paper, we propose a new standard to measure the effectiveness of the algorithm proposed in this paper. For a trajectory T, we view the semantic trajectory ST_1 generated by the typical approach as the standard one and compare the semantic trajectory ST_2 generated by the SEPSIM approach with its difference. Firstly, we segment ST_1 and ST_2 by the move state. Then, we compared the accuracy of each pair of subtrajectories T_{s1} and T_{s2} between ST_1 and ST_2. The effectiveness of ST_2 generated by the SEPSIM process approach is defined as the average accuracy of matched semantic information.

$$\text{Effectiveness}(ST_2) = \frac{\sum T_{s2}.\text{Accuracy} * T_{s2}.\text{Count}}{ST_2.\text{Count}}, \quad (1)$$

$$T_{s2}.\text{Accuracy} = \frac{\text{matchedGeoInf of } T_{s2}.\text{Count}}{\text{standardGeoInf of } T_{s1}.\text{Count}}, \quad (2)$$

where semantic trajectory ST_2 is generated by the SEPSIM approach of a given trajectory, T_{s2}. Accuracy means the correct matched semantic information accuracy of the subtrajectory T_{s2} compared to corresponding subtrajectory T_{s1} in ST_1, which is defined as the ratio of correct matched semantic information quantity in T_{s2} (matchedGeoInf of T_{s2}.Count) to the standard semantic information quantity in T_{s2} (standardGeoInf of T_{s1}.Count); T_{s2}.Count and ST_2.Count represent the number of sampling points contained in T_{s2} and semantic trajectory ST_2. Obviously, the higher the average accuracy of a matched subtrajectory, the more effective our proposed algorithm will be.

Figure 7(a) shows the change in effectiveness with the increasing preannotated trajectories. Obviously, after processing more and more preannotated trajectories, the



(a) Runtime comparison under different approaches

(b) Runtime comparison using different indexes

Figure 9: Performance comparisons among different approaches.

effectiveness of trajectories that need to be enriched is gradually increasing. When the number of preannotated trajectories reaches 4000, the effectiveness exceeds 90% and keeps increasing steadily. Figure 7(b) shows the change in effectiveness with the increasing test trajectories. It can be seen that the effectiveness of test trajectories keeps above 90%.

On the other hand, to evaluate the effectiveness of the SEPSIM algorithm, we compare the semantic trajectories generated by the baseline approach and by the SEPSIM approach in the form of visualization. Figure 8(a) shows the geographical object information represented by red boxes and corresponding topological relationships of a given trajectory in OSM map. Figure 8(b) shows the geographical object information and corresponding topological relationships enriched in the given trajectory by the baseline approach, which annotate all relevant and reasonable geographical object information. Figure 8(c) shows that trajectory matched with different episodes represented by different colors annotates the same geographical object information. It can be seen that the algorithm proposed in this paper can annotate reasonable semantic information for spatiotemporal trajectories in geospatial environment.

5.3. Efficiency. In this section, we study the efficiency of our proposed algorithms. We compare it with the baseline approach and the LCS approach, which can annotate the semantic information on the similar trajectories. For each trajectory in the GeoLife dataset, we generate the semantic trajectory by the SEPSIM approach, the baseline approach, and the LCS approach, respectively, to retrieve the running time. The results of comparison are shown in Figure 9(a). We can see that the baseline approach and the LCS approach take more time annotating the same number of test trajectories than the SEPSIM approach. With the increasing test trajectories, the time spent by the typical approach and the LCS approach and the time spent by the SEPSIM approach gradually become more time-consuming.

Figure 9(b) shows the efficiency of the SEPSIM approach with different spatial indexes. Obviously, the time spent by the SEPSIM with the *R*-tree index is much less than that of the other two spatial indexes in the SEPSIM approach, which means the *R*-tree index is appropriate to the dataset in this paper. Meanwhile, the SEPSIM approaches with the three indexes are faster than the typical approach and the LCS approach, which represents the high efficiency of our proposed SEPSIM approach.

6. Conclusion

In this paper, we study the problem of the semantic enrichment process for spatiotemporal trajectories in geospatial environments. We first directly use semantic information in preannotated semantic trajectories for annotating spatiotemporal trajectories by the SEPSIM approach. It includes three phases: preannotated semantic trajectory storage, spatiotemporal similarity measurement, and semantic information matching. We propose an algorithm named Semantic Information Matching Algorithm based on Similar Episodes (SIM) for matching semantic information. In order to improve the performance of efficient enrichment processing, we establish an *R*-tree index to query preannotated semantic trajectories. Finally, we conduct extensive experiments over a real dataset. The experimental results verify the superiority of our proposed approach in terms of effectiveness and efficiency.

Data Availability

The trajectory dataset used to support the findings of this study can be made available at https://www.microsoft.com/en-us/download/details.aspx?id=52367.

Disclosure

This paper expands on the short paper "Efficient Semantic Enrichment Process for Spatiotemporal Trajectories," which was published in 4th Asia-Pacific Web and Web-Age Information Management, Joint Conference on Web and Big Data, APWeb-WAIM 2020.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Funding

This study was supported by NSFC41971343 and NSFC61702271.

Acknowledgments

This study was supported by the NSF of Jiangsu Province (BK20200725) and the Postgraduate Research Innovation Program of Jiangsu Province (KYCX201258).

References

[1] D. Daowd and S. Mallappa, "Semantic analysis techniques using twitter datasets on big data: comparative analysis study," *Computer Systems Science and Engineering*, vol. 35, no. 6, pp. 495–512, 2020.

[2] L. H. Qi, R. G. Chen, and X. Wen, "Research on the LBS matching based on stay point of the semantic trajectory," *Journal of Geo-Information Science*, vol. 16, no. 5, pp. 720–726, 2014.

[3] A. Hussain, B. N. Keshavamurthy, and R. Prasad, "Accurate location prediction of social-users using mHMM," *Intelligent Automation & Soft Computing*, vol. 25, no. 3, pp. 473–486, 2019.

[4] F. Zhu, J. Gao, and C. Xu, "On selecting effective patterns for fast support vector regression training," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 8, pp. 3610–3622, 2018.

[5] T. Bry and T. Fureche, "Web and semantic web query languages: a survey," *Reasoning Web, Msida, Malta: Computer Science*, vol. 3564, pp. 35–133, 2005.

[6] Z. X. Yan, D. Chakraborty, and C. Parent, "Semantic trajectories," *ACM TIST*, vol. 4, no. 3, pp. 1–38, 2013.

[7] C. Parent, S. Spaccapietra, C. Renso et al., "Semantic trajectories modeling and analysis," *ACM Computing Surveys*, vol. 45, no. 4, pp. 1–32, 2013.

[8] A. Daniel and S. Thad, "Using GPS to learn significant locations and predict movement across multiple users," *Personal and Ubiquitous Computing*, vol. 7, no. 5, pp. 275–286, 2003.

[9] K. John and H. Eric, "Predestination: inferring destinations from partial trajectories," in *International Conference on Ubiquitous Computing*, pp. 243–260, Orange County, CA, USA, 2006.

[10] A. T. Palma, V. Bogorny, B. Kuijpers, and L. O. Alvares, "A clustering-based approach for discovering interesting places in trajectories," in *Proceedings of the 2008 ACM symposium on Applied computing - SAC '08*, pp. 863–868, Fortaleza, Ceara, Brazil, 2008.

[11] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W.-Y. Ma, "Recommending friends and locations based on individual location history," *ACM Transactions on the Web*, vol. 5, no. 1, pp. 1–44, 2011.

[12] S. Spaccapietra, C. Parent, M. L. Damiani, J. A. de Macedo, F. Porto, and C. Vangenot, "A conceptual view on trajectories," *Data & Knowledge Engineering*, vol. 65, no. 1, pp. 126–146, 2008.

[13] M. Baglioni, J. Macêdo, and C. Renso, "Towards semantic interpretation of movement behavior," in *12th AGILE Conference Advances in GIS*, pp. 271–288, Hannover, 2009.

[14] A. Vandecasteele, R. Devillers, and A. Napoli, "From movement data to objects behavior using semantic trajectory and semantic events," *Marine Geodesy*, vol. 37, no. 2, pp. 126–144, 2014.

[15] P. T. Nogueira, R. B. Braga, and H. Martin, "An ontology-based approach to represent trajectory characteristics," in *The 5th International Conference on Computing for Geospatial Research and Application*, pp. 102–107, USA, 2014.

[16] T. P. Nogueira and H. Martin, "Qualitative representation of dynamic attributes of trajectories," in *17th AGILE Conference on Geographic Information Science*, Castellón, Spain, 2014.

[17] T. P. Nogueira, R. B. Braga, and C. T. Oliveira, "FrameSTEP: a framework for annotating semantic trajectories based on episodes," *Expert Systems with Applications*, vol. 92, pp. 533–545, 2018.

[18] L. G. Xiang, T. Wu, and J. Y. Gong, "A geo-spatial information oriented trajectory model and spatio-temporal pattern quering," *Acta Geodactica et Catographica Sinica*, vol. 43, no. 9, pp. 982–988, 2014.

[19] T. Sun, Z. Huang, H. Zhu, Y. Huang, and P. Zheng, "Congestion pattern prediction for a busy traffic zone based on the hidden Markov model," *IEEE Access*, vol. 9, pp. 2390–2400, 2021.

[20] E. Martin, K. P. Hans, and S. Jorg, "A density-based algorithm for discovering clusters in large spatial databases with noise," KDD, Portland, Oregon, USA, 1996.

[21] D. Mountain and J. Raper, "Modelling human spatio-temporal behaviour: a challenge for location-based services," in *Proceedings of 6th International Conference on Geocomputation*, The University of Queensland, Brisbane, Australia, 2001.

[22] M. P. Dubuisson and A. K. Jain, "A modified Hausdorff distance for object matching," in *Proceedings of 12th international conference on pattern recognition*, pp. 566–568, Jerusalem, Israel, 1994.

[23] J. Kima and S. Mahmassanibhan, "Spatial and temporal characterization of travel patterns in a traffic network using vehicle trajectories," *Symposium on Transportation and Traffic Theory*, vol. 9, pp. 164–184, 2015.

[24] M. M. Fréchet, "Sur quelques points du calcul fonctionnel," *Rendiconti del Circolo Matematico di Palermo (1884-1940)*, vol. 22, no. 1, pp. 1–72, 1906.

[25] Z. Chen and H. T. Shen, "Searching trajectories by locations: an efficiency study," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pp. 255–266, Indianapolis, Indiana, USA, 2010.

[26] F. Zhu, J. Yang, J. Gao, C. Xu, S. Xu, and C. Gao, "Finding the samples near the decision plane for support vector learning," *Information Sciences*, vol. 382-383, pp. 292–307, 2017.

[27] A. Guttman, "R-trees: a dynamic index structure for spatial searching," in *Proceedings of the 1984 ACM SIGMOD*

international conference on Management of data, pp. 47–57, Boston, Massachusetts, USA, 1984.

[28] R. Kanth, S. Ravada, and D. Abugov, "Quadtree and R-tree indexes in oracle spatial: a comparison using GIS data," in *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pp. 546–557, Madison, Wisconsin, USA, 2002.

[29] X. F. Xu, L. Xiong, and V. S. Sunderam, "D-Grid: Abn in-memory dual space grid index for moving object databases," in *2016 17th IEEE International Conference on Mobile Data Management (MDM)*, pp. 252–261, Porto, Portugal, 2016.