Research Article

Hierarchical Matching of Traffic Information Services Using Semantic Similarity

Zongtao Duan, Lei Tang, Zhiliang Kou, and Yishui Zhu

School of Information Engineering, Chang’an University, Xian, Shaanxi, China

Correspondence should be addressed to Lei Tang; tanglei24@gmail.com

Received 13 April 2018; Accepted 14 May 2018; Published 7 June 2018

Copyright © 2018 Zongtao Duan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Service matching aims to find the information similar to a given query, which has numerous applications in web search. Although existing methods yield promising results, they are not applicable for transportation. In this paper, we propose a multilevel matching method based on semantic technology, towards efficiently searching the traffic information requested. Our approach is divided into two stages: service clustering, which prunes candidate services that are not promising, and functional matching. The similarity at function level between services is computed by grouping the connections between the services into inheritance and noninheritance relationships. We also developed a three-layer framework with a semantic similarity measure that requires less time and space cost than existing methods since the scale of candidate services is significantly smaller than the whole transportation network. The OWL_TCI4 based service set was used to verify the proposed approach. The accuracy of offline service clustering reached 93.80%, and it reduced the response time to 651 ms when the total number of candidate services was 1000. Moreover, given the different thresholds for the semantic similarity measure, the proposed mixed matching model did better in terms of recall and precision (i.e., up to 72.7% and 80%, respectively, for more than 1000 services) compared to the compared models based on information theory and taxonomic distance. These experimental results confirmed the effectiveness and validity of service matching for responding quickly and accurately to user queries.

1. Introduction

For transportation systems to advance strongly in China, greater financial investment in infrastructure, e.g., roads, docks, and power stations, is important. In addition, we must look beyond such measures to find new ways to expedite transportation development. With the increased availability of intelligent technology, urban areas (i.e., cities and their surrounding areas) in China have considerably accelerated their drive towards modernization, thereby bringing about benefits to the transportation system of the entire country [1]. In particular, these advances have facilitated the dissemination of traffic information and have helped to promote compliance with its requirements. The rapid growth of the Internet and mobile Internet of things (IoT) devices relies on diversification, magnitude, and heterogeneity as their dominant measures of quality for traffic services [2]. However, considering real-world scenarios, particularly in China, different data formats, languages, and dialects will produce diverse travel demands. The data generated by these services often prove that the communication of diverse needs is difficult. Such an understanding demands more than the ability to share data and improve the maximum use of the services. Clearly, a strategic key for the development of transportation systems is the creation of traffic information services (TIS) that meet the wide variety of needs and preferences of travelers, providing them with prompt and accurate access to a variety of services by merging multiple distinct technologies. Service matching focuses on finding the most similar information to a given query. For example, the system can match the TISs which tell a user where a restaurant is, what time the bus comes, etc.

To meet the unique needs of commuters, tourists, business travelers, and others, the massive amounts of information available on the traffic environment are organized into a variety of TIS offerings that involve varying degrees of complexity. These services provide drivers with information, e.g., alerts about road conditions, unexpected incidents, or
roadwork. To make public transit more acceptable, TIS may also cover information about ticketing and current operations, as well as the availability of park-and-ride facilities [3]. Although a single search engine can intelligently categorize and classify thousands of pieces of traffic information, service matching for transportation has incomparable advantages. First, service matching can provide an association between two or more applications to provide information that goes beyond anything that a single application can provide. For example, a complete travel solution for an individual using public transportation could cover ticketing, timetables, and information about accommodations, as well as the location, hours, and cost of particular attractions. Second, many mobile devices, e.g., in-car systems, provide dynamic value-added services, such as recommending only nearby restaurants, that are different from the types of information that web-based search engines offer. Therefore, faced with abundant traffic services, accurate and rapid service matching can provide robust solutions to meet a variety of requirements. Service matching can be improved by the addition of semantic features, such as traffic information, behaviors, and preferences of travelers [4, 5], that can be applied for a better understanding of an individual’s travel goals.

Therefore, it is particularly important to maximize service matching for TIS solutions and reduce their search space. Moreover, it is absolutely critical that the behavior and presentation of these services reflect how users consider a travel in the real world as closely as possible. In our research, we developed a hierarchical matching method with a semantic similarity measure. This approach increased the accuracy of service matching and reduced the response time. The proposed search engine can serve as the basis for creating a new TIS in order to better comply with travelers’ needs, which can be taken by transport planners and operators.

Our contributions are as follows:

1. To reduce the search space for matching, each TIS was classified automatically by using K-means clustering and was structured as an ontology tree.

2. The semantic similarity between different TIS concepts was calculated using the proposed model that included aspects of information theory and taxonomic distance.

3. The improved bipartite graph, with nodes that had two types of attributes, was applied to calculate the semantic similarity.

2. Related Work

Paolucci et al. [6] proposed a DAML-S-based service matching method based on the relationships between the concepts in a taxonomy tree. Their algorithm suggested only four rough degrees of matching, an approach that needed improvement to achieve fine-grained matching for a large number of services. Nonetheless, their work took a step forward towards matching characterized by semantic features.

The measure of the semantic similarity between services is a key issue [7]. To establish a measure of similarity, three major models have been followed [8]: one based on taxonomic distance [9], another based on information content (IC) [10], and the last based on the concepts’ properties. Harispe et al. presented a framework for assessing similarity [11]. They also suggested the criteria for associating semantic similarity measures and provided a method for optimizing the configuration parameters. The framework proposed could facilitate applications in a biomedical context by improving the understanding of semantic measures.

Sánchez et al. [12] classified the existing approaches to ontology-based semantic similarity in terms of precision, computational complexity, prior knowledge needed, and adjustable parameters. They also defined the distance between concepts as the length of the path connecting two terms in a taxonomy and incorporated taxonomic distance into a set of features for similarity assessment. Usually, methods that utilize semantic matching on the basis of the taxonomic distance attempt to quantify the similarity between two services by considering a quantifiable structure in which the similarity decreases with an increase in the distance, and vice versa. However, service matching that uses this approach is susceptible to factors such as the depth and density of the domain ontology tree and the symmetry of the relationship between the two ontology concepts. Furthermore, IC, an indicator of the amount of information provided by an ontology concept, enables the assessment of the degree of semantic similarity of words referring to these concepts [13]. In the work cited above [13], Sánchez et al. developed some ontologies to improve the measure of IC-based semantic similarity. They detailed two strategies that considered the compared concepts as belonging to the same or to two or more different ontologies, respectively. Their results showed that the accuracy improved significantly when multiple ontologies were included.

Meng et al. defined a concept’s topology and incorporated it into the similarity assessment of WordNet [14]. Different from the existing work, they introduced the depth and structure of every given concept, along with the quantity of their hyponyms, in an ontology tree as parameters in their IC model. The good performance of their measure was demonstrated by solving the problem caused by sparse data. Similarly, considering WordNet, an approach was proposed by Gao et al. for determining the semantic similarity associated with edge-counting [15]. IC was also used to define the shortest length of links treated as unequal distances between adjacent concepts. Then, the similarity was given by weighting various combinations of information sources nonlinearly or linearly. However, this approach resulted in a loss of useful information because it did not consider other paths in the net (i.e., WordNet).

The above illustrates the significance of similarity measure in service matching and discusses the drawback of existing methods, which motivates our research. Although we have learned that an IC-based model is not sensitive to the problem of varying link distance, the similarity does not depend strongly on the ontology tree and is simply recognized using term frequency-inverse document frequency (TF-IDF). Such textual statistical methods have caused problems such as data sparseness and ambiguous concepts that affect the calculation of the frequency of concepts and the assessment of further similarity, particularly for service matching in a traffic environment.
3. Service Matching for Transportation

Given a service request, the purpose of traffic information service matching is to search the service library efficiently and quickly for a candidate service subset. Currently, there are a variety of ontology languages for describing semantic-based TIS, such as OWL, WSMO, SWSO, and SAWSDL, each of which has certain advantages. Because of the standards for the Web Ontology Language for Services (OWL-S) [16], we used it to describe the traffic information services in this study. The basic framework of OWL-S [17] is made of three parts: the service profile, service model, and service grounding. Thus, any given service is represented using these three parts. The “service profile” describes what the service does, the “service model” states how it works, and the “service grounding” describes how to interact with the service.

In our research, with the help of the OWL-S profile, the problem of service matching amounted to finding a way to measure the semantic similarity of the services’ functional and nonfunctional attributes. In a traffic environment, TIS applications are offered mainly to the traveling public. Real-time dynamic information is sent to users through wireless or wired communication technologies. Information comes in the form of texts, pictures, videos, and other communications that provide dynamic, real-time access to the best traffic information anytime and anywhere. Although the development of web services matching began decades ago, these matching processes have faced the following five challenges in a traffic environment:

(1) Integration: when a traveler requests information, frequently, various functional services must be integrated to provide a complete answer to the query [18]. For example, when traveling to another city, you might require the weather forecast from a meteorological service, timetables from the suburban transportation service, and road conditions from the traffic services. To meet the user’s request fully, all of these services must work together well to provide a complete response.

(2) Dynamic heterogeneity: the information provided by TIS must consider the possibility that the traveler may leave the original geographic area after sending the request for information. Therefore, the results might be returned, in effect making the service unavailable. In addition, TIS data can be delivered by various types of mobile devices or via the Internet, as well as by roads or other means. This requirement implies that TIS data need to be considerably different from web-only data in terms of their structure [19]. Therefore, service matching must support dynamic and homogeneous access to heterogeneous services in traffic environments.

(3) Robustness: communication links between traffic facilities, and connections to mobile devices, are not as reliable as the web, so the issues inherent to mobile devices can cause traffic services to be unavailable temporarily. Therefore, priority should be given to the robustness of service matching.

(4) Real-time use: because of the rapid development of mobile IoT, there has been a significant growth in the number and variety of heterogeneous traffic information services in a wide variety of formats [20], e.g., speech-based navigation, visualization queries for traffic flows, and dissemination of information about traffic incidents. For services to facilitate travel, it is important to reduce the search space, making matching more efficient and creating a better, safer travel experience. Therefore, it is necessary to focus on the real-time performance of matching TIS data.

(5) User-friendliness: TIS search results must be flexible to meet the needs of a variety of audiences (e.g., drivers, passengers, and traffic-controllers). Each type of user looks at the same data in a different way. Therefore, service matching is required to be highly user-friendly. Semantic-based matching must have the ability to understand human thinking and present findings in a manner that parallels human reasoning.

With respect to traffic information, we found that service matching faces the following challenges:

(1) A relatively large search space leads to inefficient matching.

(2) The resources of various road infrastructures are limited.

(3) Service matching must be acquired dynamically in real time because vehicles are in motion.

To meet these challenges in transportation, it is vital to construct an efficient, real-time, lightweight model for TIS matching that can handle a large search space, heterogeneous dynamic data, and limited resources.

4. Semantic-Based Matching Model of TISs

Because of the difficulty of characterizing domain-specific knowledge using XML and WSDL, we used the OWL-S Editor to work with OWL-S to structure the traffic information services and allow for semantic reasoning. Given the description of services, the service profile (one part of the OWL-S-based services) was adopted as the base. In particular, three terms were extracted from a service profile: the name, short-text description, and functions of the TIS. In this study, the first two were used to group services automatically; the “function of the TIS” was used in input/output- (IO-) based matching. Hereafter, for convenience, we have referred to the service profile and defined the description-related set of services provided and that of service requests.

First, we defined the profile of traffic information services as follows.

Definition 1. Service = \{Description, Input-Output\}, where these attributes are drawn from the service profile of the OWL-S model, and the description consists of the name and short-text description. Input = \{input1, input2, ...\} and Output = \{output1, output2, ...\} are regarded as the input and the output function sets of a service, respectively.

The description set (DS), a set of descriptions as described above, is given as follows.

Definition 2. DS = \{Description1, Description2, ...\}. The service request is defined as Req = \{RDescription, RInput-Output\}, where RDescription denotes the description of services expected by a requestor; RInput = \{rInput1, rInput2, ...\} and ROutput = \{rOutput1, rOutput2, ...\} are the separate input and output, respectively, according to the individual’s needs.
The approach proposed in this paper involves the clustering and matching of the TISs in the library, which include the following:

1. Dividing the TISs in the library into different categories
2. Measuring the semantic similarities between the functions requested and those provided
3. Returning the candidate TIS solutions to the individual traveler after ranking the calculated similarities

4.1. Grouping TIS Using K-Means. Because there are many services for transportation, it was necessary to reduce the search space before matching to significantly improve the efficiency and the precision of the search results. Many popular text clustering methods use text modeling algorithms to reduce the query space [21]. They usually assess the similarity among the services and tag the optimal candidate category, which is then iteratively refined as more detail is developed to update the centers. Text clustering compresses the traffic information service matching space quickly, thereby improving the accuracy of the matching algorithm and reducing the time that the users must wait for a response [22]. Rajagopal et al. [17] applied clustering techniques to group a massive number of services. Services falling into the same category belonged to the same cluster, and thus, the similarity among these services was the highest. However, considering that different services were assigned to different clusters, these researchers did not further refine the process of service matching in this situation.

To accelerate the search speed for web-service queries, Wenjing et al. [23] restructured the services for clustering them hierarchically in terms of their inputs, outputs, names, and text-based descriptions. Similarly, considering the inter-cluster distance, hierarchical agglomerative clustering was performed by Surianarayanan et al., depending on both the inputs and the outputs of services with prioritization. The similarities were then identified as different degrees of match for meeting disparate queries from clients [24]. Yisong et al. [25] utilized nonfunctional descriptions to group services and provided the service-to-service matches in terms of semantic similarity.

Although OWL-S-based descriptions have been used to automate text clustering and classification in searches for the best services, the abovementioned methods failed for traffic information services. The unknown number of centers and the coarser-grained classification were identified as the reasons for the low achievement of this approach for TIS matching and the consequent failure of service delivery. Moreover, the textual descriptions may be extended in the area of information retrieval. The description of a TIS is shown below with the vector space model.

\[ \text{Description}_j = \{d_{1j}, d_{2j}, d_{3j}, \ldots, d_{nj}\} \]  

(1) The data preprocessing of the TIS in-service libraries was performed using word segmentation. Keywords were generated and traversed to create a customized dictionary.

(2) The vectorization of keywords was achieved using the vector space model and TF-IDF within the area of information retrieval. The description of a TIS is shown below with the vector space model.

\[ d_{i,j} = \frac{TF_{i,j} \times \log \left( \frac{N}{n_j} \right)}{\sqrt{\sum_{j=1}^{n} [TF_{i,j} \times \log \left( \frac{N}{n_j} \right)]^2}} \]  

where \( d_{i,j} \) denotes how frequently a word obtained from the dictionary appears in a special description of the TIS; \( d_{i,j} = 1 \) if the word appears and 0 otherwise. However, the incompleteness of data could not be captured in this way with the increasing number of TISs in the library, and the contribution of each word was ignored. Consequently, following the preprocessing operation, all of the words in the Description\(_j\) were weighted using the TF-IDF [26] as follows:

\[ D_{\text{vec}} = \frac{\sum_{j=1}^{J} \text{Description}_{ij} - \bar{u}_k}{\sum_{j=1}^{J} \text{Description}_{ij}} \]  

(3) The clustering of all of the TISs in the library by using K-means was accomplished as follows:

\[ U = \sum_{k=1}^{K} \sum_{i \in \text{DSS}} \left( \text{Description}_j - u_k \right)^2 \]  

where \( K \) denotes the number of service categories in the traffic environment and \( J \) implies the number of services belonging to the same categories. \( u_k \) denotes the cluster center, which is a vector generated by an iterative means of averaging a set of descriptions from the same category of services. \( K \) was determined experimentally in this study, an approach that facilitated more rigorous classification than that found in other existing works [27, 28]. The clustering process was used to classify a large number of TISs and keep each center’s information separate. The iterative process was continued until \( U \) in formula (3) for convergence was met. When receiving a new request for a travel service, the category of the request was determined by analyzing the request’s taxonomic distance from the centers. This method allowed the search space to be reduced quickly and improved the efficiency and the precision of TIS matching. The clustering (Algorithm 1) used in this study was as follows.
The function-level matching in our work was based on the following four assumptions:

1. All TISs are characterized on the basis of OWL-S.
2. All TISs of the same category are defined in the same domain ontology.
3. Only the inheritance relationships between nodes in the ontology tree are considered.
4. Only the input and output functions are taken into consideration for matching.

The proposed function-level matching was divided into two stages. In the first stage, we calculated the similarity between the nodes. In the second stage, we focused on the similarity between sets of nodes. Thus, the match optimization problem could be represented as a maximum-weight bipartite matching problem.

If we were to limit our method to the use of a geometric model where all of the weights of each edge were the same, the result would be a loss of accuracy for TIS matching. Therefore, to measure the similarity, we referred to Zhang et al.’s work [29] to present mixed matching by integrating a geometric model and a model derived from information theory. We observed that the similarity increased with a decrease in the semantic distance between the nodes in the ontology tree. As we went down into the deep layers of the tree, the similarity increased as well, illustrating that two nodes were similar even if they were irrelevant on inheritance. That is, it was difficult to identify the services found as meeting the personal demands of travelers without considering the relationships between the nodes in the hierarchy during matching. For example, given a tree-based library for traveling, “Hiking” and “Surfing” would be considered two sibling nodes of the parent node “Activity.” When a request delivered relates to “Surfing,” the service matching could fail because unexpected “Hiking” might be returned by the matching process if we were to compute only the similarities without thinking about the relationships of nodes, particularly inheritance.

Therefore, the calculation of similarity was revised, and the relations were captured by parameter \( \beta \), which was expressed as 1 between a given node and its children (e.g., “Activity” and “Sports”) and between 1 and \( \mu \) for a given node and its children’s descendant (e.g., “Activity” and “Swimming”) otherwise. In addition, \( \beta \) was assigned the value above \( \mu \) when two nodes were not connected on inheritance (e.g., “sightseeing” and “Swimming”). \( \mu \) was experimentally determined in this study. The following formula was used:

\[
SemSim(\text{func}_i, \text{func}_j) = 1 - \sqrt{\frac{\beta - 1}{\alpha}} \times \frac{\text{Dep}(\text{func}_j)}{\text{Dep}(\text{func}_i) + \text{Dep}(\text{func}_j)} \times \text{SemDist}(\text{func}_i, \text{func}_j)
\] (4)

where \( \text{func}_i \) and \( \text{func}_j \) provided are considered two separate nodes in tree-structured ontologies. \( \text{SemDist}(\text{func}_i, \text{func}_j) \) gives their semantic distance derived using information theory. \( \text{Dep}(\text{func}_j) \) is the depth of \( \text{func}_j \) in a tree. \( \alpha \) and \( \beta \) are used to maximize the traffic service matches.

To determine the similarity between each set of functions, a set of nodes related to the TIS was created using a bipartite graph that consisted of two disjoint sets of vertices, a set representing input functions that \( R_{\text{input}} \) requested, and a set representing input functions that \( \text{input} \) provided. An edge
Input:
\[ R_{\text{Input}} = \{ r_{\text{input1}}, r_{\text{input2}}, \ldots \} \]: input function sets requested
\[ \text{Input} = \{ \text{input1, input2}, \ldots \} \]: input function set of a service provided.
\[ R_{\text{Output}} = \{ r_{\text{output1}}, r_{\text{output2}}, \ldots \} \]: output function sets requested
Output = \{ output1, output2, \ldots \} : output function set of a service
\[ \sigma \]: similarity threshold
\[ \lambda_1, \lambda_2 \]: weights of input and output functions

Output:
Sim\_func: similarity of function sets from travelers and providers

Begin
(1) \[ \text{sim\_func} = \lambda_1 \times \text{sim}(R_{\text{Input}}, \text{Input}) + \lambda_2 \times \text{sim}(R_{\text{Output}}, \text{Output}) \]

\[ \text{sim(func}_a, \text{func}_b) \]

(1) for \( i \leftarrow 1 \) to \( m \) do
(2) for \( j \leftarrow 1 \) to \( n \) do
(3) \( e_{ij} \leftarrow \text{SemSim(func}_a[i], \text{func}_b[j]) \)
(4) if \( e_{ij} > \sigma \)
(5) \( \text{service}\_\text{set} \leftarrow \text{service}\_\text{set} \cup e_{ij} \)
(6) end if
(7) end for
(8) end for
(9) return service\_\text{set}

End

Algorithm 2: Functional matching.

between the functions requested and the functions provided existed if the match was feasible, with a weight \( e_{ij} \) that represents their similarities computed using formula (5), where \( I \) and \( J \) are the number of functions requested and provided, respectively.

\[ e_{ij} = \text{SemSim}(x_i, y_j) \]

(\( i = 1, 2, \ldots, I; \ j = 1, 2, \ldots, J \)) (5)

where \( x_i \in R_{\text{Input}}, \ y_j \in \text{Input} \)

Thus, the matching process was transformed into a problem to solve for maximizing the sum of weights. The calculation of similarity is presented in this paper using formula (6).

\[ \text{Sim}(R_{\text{Input}}, \text{Input}) = \frac{\sum_{i=1}^{I} \sum_{j=1}^{J} \text{max}(e_{ij})}{IJ} \] (6)

However, in the case of the typical weighted bipartite graph model, the number of vertices in two separate sets (e.g., input/output functions requested and provided, expressed as “\( I = J \)”) into which all functions obtained from the tree were divided must be the same. Such rigorous conditions further restricted the use of a typical model in the real world. Considering the failed matching associated with different numbers (expressed as “\( I \neq J \)”), we improved the similarities as shown in formula (7).

\[ \text{Sim}(R_{\text{Input}}, \text{Input}) = I \times \left[ \sum_{i=1}^{I} \sum_{j=1}^{J} \text{max}(e_{ij}) \right] + \frac{I \times \left( \sum_{i=1}^{I} \text{max}(e_{ii}) \right)}{2IJ} \] (7)

The calculation of similarity for a set of output functions was similar to formula (7), as previously deduced. Therefore, the semantic similarity for the function-level matching of travel services is as follows:

\[ \text{Sim}(\text{Req, Service}) = \lambda_1 \text{Sim}(R_{\text{Input}}, \text{Input}) + \lambda_2 \text{Sim}(R_{\text{Output}}, \text{Output}) \] (8)

where \( \lambda_1 + \lambda_2 = 1 \), and \( \lambda_1 \) and \( \lambda_2 \) imply whether travelers find the input or output functions more or less attractive.

The function-level semantic matching is illustrated in Algorithm 2. This algorithm determines the similarities using formula (7) with a generated bipartite graph model and then gives the final similarities using formula (8). It has a function that calculates the similarity between two nodes according to formula (4).

4.3. Hierarchical Matching-Based Process. The layered matching process proposed in this study includes the following steps.

(1) The TIS are divided into different categories using clustering in the registration center. The requestor then identifies the categories associated with his/her query through the text mining technology.
(2) A tree-structured ontology is created for function-level matching based on an evaluation of the similarities between the functions requested and provided. The candidate sets that fit within the functional constraints are selected.

(3) Finally, the similarity values are ranked to provide the requestor with the desired TIS solution. The pseudo code of the algorithm is given in Algorithm 3. Line (3) applies K-means clustering to group the TISs. Lines (4)–(8) extract and process the function and textual contents of the service requested. Lines (9)–(14) perform the layered matching. Line (15) screens the services that meet the requirements, ranks them, and returns them to the requestor.

### Algorithm 3: Semantic matching using mixed model of TIS.

```plaintext
Input:
Req = [R_Description, R_Input-Output]; service request
θ: similarity threshold
Output:
result_set: service sets with similar functions
Begin
(1) service_set ← null
(2) allServices ← getServices()
(3) cluster ← getServiceCluster(service_set)
(4) serviceCategory ← getServiceCategory(service_set)
(5) R_Des ← ServiceDescriptionExtraction(Rq)
(6) R_func ← ServiceFunctionExtraction(Rq)
(7) R_cate ← ServiceClassification(Rq)
(8) Services ← getServicesFromSpecifiedCategories(Rcate, cluster, serviceCategory)
(9) for S in Services do
(10) sim_func ← sim(ServiceFunctionExtraction(S), R_func)
(11) if sim_func > θ
(12) result_set ← result_set ∪ S
(13) end if
(14) end for
(15) SortedBySimilarity(result_set)
End
```

5. Architecture for Semantic Matching of TISs

While a TIS placed on web servers tends to be available consistently for reference, other TIS applications developed for mobile use may be unavailable at various times as a result of location issues. Because of the diverse conditions encountered while traveling (e.g., while commuting or during long journeys), the main issue for TIS matching is the assignment of requirements that arise dynamically over time at various locations relative to the service providers. First, we illustrated the hierarchical framework and classified it as the three-layer structure shown in Figure 1. Then, we described the characteristics of some important functions and proposed an architectural implementation.

(i) The user layer serves as an interface between users and the matching module. Users include service providers who present local information, such as news and weather, as well as travelers querying the TIS. By interacting with a website, a traveler submits a request indicating the desired functions bounded by time and space. The textual descriptions of the TIS can be registered and found in the data layer.

(ii) The matching agency layer operates the distributed TIS matching module associated with the request integrated by the user layer. The matching agency layer includes three core parts of the proposed matching model: the feature extraction module, the clustering module for the automatic grouping of the TISs having similar functions and facilitating fast service discovery, and the I/O-level functional matching module. Figure 2 illustrates how the three modules work together. The tasks of the last two modules are performed sequentially, where a match fails if one of them does not work under a certain reasonable threshold. Thus, we can provide an automated process for picking up services that satisfy a traveler’s demands with a weighted sum of modules.

(iii) The data layer provides the underlying data support, such as an in-built domain ontology library and a set of TIS solutions semantically extended and registered. Because of the data loss that can occur while using an integrated data structure, the data in the registration center should be backed up.

(a) Data Preprocessing. The descriptions of TISs in the library, represented using OWL-S, should be used for further matching with data preprocessing. This includes the following steps:

(1) The interfaces, i.e., OWLOntology and OWLKnowledgeBase, are adopted to parse the OWL-S-based description for the names and textual descriptions of the TIS. For example, given a TIS for hiking, we can give its name and textual description as “Hiking Urban Area Service” and “this service returns the best urban areas for a given hiking type.” Table 1 gives
8 Journal of Advanced Transportation

User layer

Service Requestor

Service Provider

return

submit

describe

Matching agency layer

Sorter

Matching Module

IO-level Functional Matching

Clustering

Feature Extraction

Data layer

Service

Publication

OWL-S based Description

Domain Ontology Library

Figure 1: Proposed architecture.

Figure 2: Matching using semantic similarity.

an example showing the keywords obtained from five OWL-S-based TIS names.

(b) Service Classification. Using the proposed method, we could find the desired TIS without searching the entire service library by using the following steps to measure the similarity between the keywords and to identify the center by clustering. First, we applied the WordNet tool to extend the keywords obtained. This action provided a better understanding of the meaning of each of the words and allowed further vectorization, as discussed in Section 4.1. However, this approach resulted in high-dimensional data that were so sparse that they reduced the precision and recall of TIS matching. Consequently, we introduced a latent semantic index (LSI) [30] to reduce the dimensions of the keywords. Then, we used K-means clustering as shown in Figure 3, where the number of service categories was determined experimentally, and the services were then labeled. Each cluster center was considered an atomic service for each category. A query represented as vectors from a traveler would be matched in terms of the similarity with each atomic service. Then, the query would be assigned to the most similar service category and treated as the one requested by the user. Thus, this method facilitated the reduction of the search space and maximized the number of TIS matches by using clustering.

(c) Functional Matching. There were four types of information associated with the TIS input/output involved in this matching. Because these functions were adequately described by simple terms, it was difficult to extract matches by using similarity measures when the functions were represented in a natural language. Therefore, we generated an ontology tree to allow for semantic matching and to improve the precision. Figure 4 illustrates the functional matching where the similarity between the nodes in the tree, and between the sets of nodes, was sequentially identified offline. This procedure also decreased the waiting time in the case of a large number of concurrent requests.

6. Performance Analysis

The database of OWLS-TC [31] provides 1083 semantic services from a wide variety of fields, such as education, communication, and geography. We adopted OWLS-TC to verify the model proposed in this paper.

6.1. Verification of Clustering. Three groups were selected from the categories of tourism. The total number of TISs in each group was 50, 100, and 150. To better reduce the dimensions of the feature data used for further clustering, we applied a 12-fold cross-validation test to the acquired
feature set and determined the top ten features with the most relatively high values using LSI. In addition, considering that clustering varied each time the K-means algorithm was used because the initial cluster center could be identified randomly, we averaged the accuracy of clustering from three experiments for each group. We measured the accuracy of the classification results, i.e., the similarity of the classification results to the ground truth results, and found that the three groups achieved an accuracy of 87.85%, 90.67%, and 93.80%.

The experiment showed that the error rate of the TIS classification decreased rapidly with an increase in the number of services. This could be attributed to the fact that it was possible to obtain more features to distinguish one category from another for clustering by increasing the number of textual descriptions. Furthermore, the features and the number identified using a cross-validation test contributed positively to the LSI-based process and further improved the classification.

### Table 1: Some of the descriptions including the name and the keywords in the service library.

<table>
<thead>
<tr>
<th>Name of TIS</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>Car Price Service</td>
<td>purchase, price, wheel, model, etc</td>
</tr>
<tr>
<td>Adventure Urban Area Service</td>
<td>adventure, urban area, return, etc</td>
</tr>
<tr>
<td>Activity Town Service</td>
<td>activity town, name, provide, etc</td>
</tr>
<tr>
<td>Auto Bicycle Price service</td>
<td>auto bicycle, price, rent price, etc</td>
</tr>
</tbody>
</table>

### Table 2: Comparison results of response times.

<table>
<thead>
<tr>
<th>Number of services</th>
<th>Clustering</th>
<th>Non-clustering</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>162 ms</td>
<td>356 ms</td>
</tr>
<tr>
<td>100</td>
<td>216 ms</td>
<td>425 ms</td>
</tr>
<tr>
<td>150</td>
<td>239 ms</td>
<td>495 ms</td>
</tr>
<tr>
<td>1000</td>
<td>651 ms</td>
<td>2361 ms</td>
</tr>
</tbody>
</table>

#### 6.2. Verification of Response Time.

The effect of service clustering on the model’s response time was measured using four groups of data from five categories when the number of services was 50, 100, 150, and 1000. Table 2 shows a comparison of the response times needed in the matching model, including clustering and nonclustering. The results indicated that there would be an explosion in the response times without offline clustering; therefore, it would be difficult to accelerate the matching.

The experiment also revealed that calculating the similarity between travel services represented as ontology-based
concepts was very time-consuming. Therefore, in this study, to improve the speed and accuracy of service matching, the similarity could be measured offline and stored in a file. This was useful when similar parts of concepts were found during past matching, because we could create an index to find the desired TIS without the same concepts being counted twice while determining the similarities.

6.3. Recall and Precision Rate. We evaluated the performance of three models: a model based on information theory, another based on taxonomic distance, and the proposed mixed matching model. In terms of the parameter settings for formula (4), $\delta$ was set to 2 and $\beta$ was set to 1, 6, and 10, indicating the relationships of the nodes in the generated ontology tree, i.e., direct inheritance, indirect inheritance, and noninheritance, respectively. The similarity threshold denoted by $\theta$ in the range from 0 to 1 was experimentally determined to be 0.7 and 0.8.

Figures 5 and 6 show the variation trends of the recall and the precision rates of the three models when $\theta$ had two different values. Clearly, the three models worked well in terms of precision when $\theta$ was 0.8. We found that the model based on information theory performed better in terms of precision, but its recall rate was relatively low because a service was likely to be matched incorrectly to other sibling nodes. Furthermore, the model had good precision if and only if there were very few sibling nodes in the ontology tree, and the similarity between concepts was relatively low. We might lose other matches from the information theory-based model because the similarities calculated could fall below the threshold $\theta$. The precision and the recall rate of the model based on taxonomic distance ranged between the performance of the other two models. When there was a wide similarity threshold, this model's performance was in accordance with the results obtained by using the proposed mixed matching model, whose precision decreased.

On the basis of the performance analysis, we concluded that the recall rate of the mixed matching model was higher than that of the other two models because the similarities between the concepts increased as a result of the parameter adjustment. The precision rate decreased mainly as a result of the low accuracy of the matching performed using a bipartite graph for multiple inputs or outputs. We also found that the performance of the mixed matching model was better than that of the other two models as a consequence of increasing the similarity threshold i.e., when the threshold was 0.8 in the experiments.

7. Conclusions

In this paper, we proposed a multilevel semantic matching model for traffic information services. First, we used K-means to classify the services automatically. Then, we applied ontology knowledge and a bipartite graph to calculate the semantic similarity between functions. We demonstrated how the proposed model could be used to implement semantic matching for TIS solutions. Our experiments revealed the effectiveness and feasibility of the model for handling a large search space, heterogeneous dynamic data, and limited resources. We suggested that distributed TIS matching might be the acceptable solution for user queries in parallel. However, there was still room for improvement in terms of the response time. Further, our model was tested on the basis of prior knowledge, i.e., by using a predetermined ontology tree. While this approach allowed for offline clustering, similarity measure, and making matching decisions quickly, this form of knowledge limited the model's ability to understand real-world travel purposes. Going forward, we believe that it would be useful to examine QoS-based matching as a basis for understanding the needs of people who are making local or longer journeys. In addition, we will focus on optimizing the process of service matching considering the specific attributes of traffic information, e.g., disconnection, geo-space issues, and temporal aspects. It is therefore necessary to deal with more than only domain heterogeneity.

Data Availability

Previously reported OWLS-TC data were used to support this study and are available at [http://projects.semwebcentral.org/projects/owls-tc/]. These prior studies (and datasets) are cited at relevant places within the text as [31].

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.
Acknowledgments

This work was funded by the National Natural Science Fund of China under Grant no. 61303041, Shanxi Province Industrial Research Projects under Grants nos. 2015GY-002 and 2016GY-078, and Funds for Key Scientific and Technological Innovation Team of the Shaanxi Province, China, under Grant no. 2017KCT-29.

References


12 Journal of Advanced Transportation


Submit your manuscripts at www.hindawi.com