Web service recommendation based on event ontology

Hang Lv1*, Junjun Pan2, Jinliang Wu3 and Haiyang Ren3

1College of computer engineering and Science, Shanghai University, Shanghai 200444, China
2Pudong New Area Big Data center, Shanghai, 200135, China
3The 54th research Institute of China Electronics Technology Group Corporation, Beijing 100036, China
*Corresponding author’s e-mail: lvhang96@shu.edu.cn

Abstract. Web service recommendation is an important means to help users quickly discover and select services. Service recommendation combined with situational information is one of the development trends of Web services in the future. This paper proposes a Web service recommendation method based on event ontology, which abstracts Web service requirements into service event classes, according to the historical service records of the target users, Web service recommendations combined with service event ontology and content similarity calculation are recommended, and event class elements and hierarchical models in service event ontology are used as reference, calculate the service similarity and generate the fusion recommendation result according to the weight; Then, based on the current situation status of the user, the user's current situation is located to a node on the event ontology situation model, the predictive recommendation of services is based on the semantic relationship between event classes in the event scenario model. Finally, the service with the highest similarity among the two parts of service recommendation is recommended to users. The experimental results show that this method can better improve the accuracy and diversity of service recommendation.

1.Introduction
With the wide application of service-oriented computing mode, the number of Web services in the Internet is increasing exponentially. The emergence of more and more Web service directories, registries and market demands has led to information redundancy, which is a waste of internet resources [1] [2]. However, there are two important ways to solve the problem of information overload-search and recommendation. Web service search refers to searching based on services actively entered by users. However, Web service recommendation is different. It predicts the service required by the user by analyzing the user's history or current situation when the user does not propose a demand. Moreover, due to the user's lack of a field knowledge, as well as searching for the same content at different times and in different places, the results of actively searching may not be satisfied by users. Recommendation can make up for the shortcomings of searching.

Early Web service recommendation methods are recommended through service descriptions. For example, Li S and other [3] select useful fields in the wsdls file that describe service functions, which can represent the service field reuse statistics. The service characteristics cannot be better displayed only by description. In order to better express the service characteristics, QoS (Quality of Service) value is introduced, for example, Bagga P et al. [4] proposed a method based on QoS parameter definition and multi-criteria decision-making to help users choose appropriate services. However, in the actual
situation, QoS data is small and unstable, which makes it difficult for users to collect real QoS data to affect the recommendation results [5]. Therefore, in order to solve the problem of less QoS data, Yin Y et al [6] proposed a new matrix decomposition model based on deep feature learning, the model also integrates convolution neural network to judge the characteristics of users and services, and obtains higher QoS prediction results. In the later research, it became more popular to regard the current user information and user situation as important factors affecting service recommendation. Ling-Lan G U and others [7] proposed a comprehensive user evaluation, the similarity of user features and context information are clustered to filter the results and generate Web service recommendation results. Kang G et al. [8] proposed a method based on personalized clustering and trust perception, which uses clustering to identify similar users and displays text information, the context information of scoring information and implicit information is integrated to calculate task similarity and make reliable service recommendations.

2. Service model based on event ontology

2.1. Service level model
The hierarchical model in the event ontology mainly shows the classification relationship of events. Mapping to service description is the hierarchical model of services, which is constructed according to the classification standard of services by the World Trade Organization. In the hierarchical model, the sub-services of travel service include hotel service, play service, transportation service and catering service, and these sub-services are taken as the second layer of the model. As shown in figure 1.

2.2. Service scenario model
The scenario model in the event ontology usually describes the logical relationship between event classes. Accordingly, the service scenario model describes the user to achieve a certain goal, the logical relationship model between Web services formed in the process of searching and using Web services. Taking the situation in the travel service as an example, starting from the trip booking, when we travel booking to a place, we usually check the hotel at that place and take a taxi to go there, at the same time, you will search for local food and scenic spots and book them in advance. As shown in figure 2.

3. Web service recommendation based on event ontology

3.1. Web service recommendation combining service event ontology and content similarity calculation
Firstly, collect the historical records of users using services and vectorize the services used by users. Then, we calculate the similarity of elements and hierarchical models between other services and services used by users based on the elements and hierarchical models in the event ontology. Finally, because each element and hierarchical model has different influence on similarity calculation between services, different weights should be assigned for fusion weight calculation.
3.1.1. Element similarity. Similarity calculation based on event elements is carried out from two aspects. All service event classes are classified according to the time when the event occurs. The time when the event occurs is from 2:00 to 5:59, 6:00 to 9:59, 10:00 to 13:59, 14:00 to 17:59, 18:00 to 21:59, 22:00 to 1:59 (the next day) is divided into Early Morning, Morning, Noon, afternoon, evening and late night, where the location event set T={t₁, t₂, t₃, t₄, t₅, t₆}; or classify the location elements in the event into: home, company (school), outdoor and public places, corresponding location elements set P={p₁, p₂, p₃, p₄}. Calculating the similarity of time (location) elements of a service is to calculate the distance between the current service vector and other service vectors after vectorizing the service in vector space. Where vectorization represents the three-dimensional matrix space of an object set, a service set, and a time set (or a place set). The service used is represented in the three-dimensional matrix space, that is, the discrete text features are mapped to the vector space, and the values in the matrix are the user's rating of the service.

We use cosine formula to calculate the distance between service vectors. Assume that the similarity of time elements and location elements of service sᵢ and sⱼ are calculated under n time elements and m location elements as Simₜ(sᵢ, sⱼ), Simₚ(sᵢ, sⱼ) respectively, element similarity Simₑ(sᵢ, sⱼ) is calculated as (1):

\[
Simₑ(sᵢ, sⱼ) = \frac{\sum^n \text{sim}_t(sᵢ, sⱼ)}{n} + \frac{\sum^m \text{sim}_p(sᵢ, sⱼ)}{m}
\]

(1)

Where Simₜ(sᵢ, sⱼ), Simₚ(sᵢ, sⱼ) similarity is calculated by cosine similarity.

3.1.2 hierarchical model similarity. Taking the hierarchical model of travel service as reference, the four sub-event classes of the second layer of the travel service hierarchical model are taken as classification criteria, and the four sub-event classes correspond to the set H={h₁, h₂, h₃, h₄}, and then classify all services that occur under the second layer sub-event class according to the hierarchical model. Assuming that services sᵢ and sⱼ belong to hₙ, the user's services are vectorized as Vᵢ and Vⱼ in a four-dimensional space of service, user, time, and location, the formula for calculating similarity Simₕ(sᵢ, sⱼ) is (2):

\[
Simₕ(sᵢ, sⱼ) = \begin{cases} 
\frac{Vᵢ \cdot Vⱼ}{||Vᵢ|| ||Vⱼ||} & \text{If services } sᵢ \text{ and } sⱼ \text{ belong to a parent event class} \\
0 & \text{If services } sᵢ \text{ and } sⱼ \text{ do not belong to a parent event class}
\end{cases}
\]

(2)

Due to the different influence of elements and hierarchical models in event ontology on similarity calculation between services, weighted distribution is adopted for weight fusion calculation. Combine the similarity between Simₑ(sᵢ, sⱼ) and Simₕ(sᵢ, sⱼ) into the total similarity Simₑᴴ(sᵢ, sⱼ) formula (3):

\[
Simₑᴴ(sᵢ, sⱼ) = αSimₑ(sᵢ, sⱼ) + βSimₕ(sᵢ, sⱼ)
\]

(3)

Among them, α and β are the weights of two similarities, which satisfy α + β = 1. The larger Simₑᴴ(sᵢ, sⱼ), the more similar the service sᵢ and sⱼ are.

3.2. Web service recommendation based on event context

Assuming that the non-hierarchical relationship between specific service event classes in the travel service is of the same importance, we assume that if two service event classes occur at the same time and at the same place multiple times, the two event classes are considered to have a non-hierarchical relationship. Set service Sₒ and service set S={s₁, s₂, s₃, ..., sₙ}, centered on S, the number of occurrences of s₀ and a service sᵢ in the collection as radius Pᵢ, and set the minimum threshold of the radius of the relationship to R, if R ≤ Pᵢ, the service s is deemed to meet the conditions, records the qualified services and constructs a scenario model according to the description in section 3.2.

Locate the location node of the user in the scenario model based on the time and location of the user and the service currently in use. The currently located service event class is s₀, which is traversed to the directly adjacent service event class through breadth first, and recorded to the service collection.
Class $S=\{s_1, s_2, s_3, ..., s_n\}$, calculate the situational similarity between the current service event class $s_0$ and other event classes $s_i$ in the collection Class $S$. The $\text{Sim}_N$ formula is (4):

$$\text{Sim}_N(s_0, s_i) = \sum^n_i \text{Sim}_{\text{element}}(s_0, s_i)$$

(4)

Where $\text{Sim}_{\text{element}}(s_0, s_i)$ indicates the similarity between service event class $s_0$ and $s_i$ based on context under a certain element. In this experiment, context-based service recommendation only considers time and place elements to calculate, so $n$ is recorded as 2, while element represents time (T) and place (P). You only need to calculate the similarity of context-based elements under time and place elements. Under a certain element, the formula for calculating the element similarity $\text{Sim}_{\text{element}}(s_0, s_i)$ based on the situation is (5):  

$$\text{Sim}_{\text{element}}(s_0, s_i) = \frac{|s_0 \cap s_i|}{|s_0|} = \frac{|\text{count}(s_0 \cap s_i)|}{|\text{count}(s_0)| + |\text{count}(s_i)| - |\text{count}(s_0 \cap s_i)|}$$

(5)

Where $|\text{count}(s_0 \cap s_i)|$ indicates the number of times that service event classes $s_0$ and $s_i$ occur simultaneously under a certain element, while $|\text{count}(s_0)|$ and $|\text{count}(s_i)|$ indicates the number of times that service event classes $s_0$ and $s_i$ occur separately under a certain element.

4. Experimental Analysis

4.1. Dataset and Experimental Environment

The dataset in this article uses Python to crawl the real service information on ProgrammableWeb websites as a reference for building the model, including 1734 service information, mainly including the types of service information: Transportation, Food, Hotels, travel and Play. At the same time, 200 users were interviewed in the form of online questionnaires, and their occupations were teachers, students, workers, etc. They obtained the use of "Ctrip", "Meituan", the service usage records of users on related travel apps such as Zhixing can be recorded to obtain no less than 30 pieces of service usage information for each user.

4.2. Evaluation Criteria

In this paper, Precision, Recall, F-measure and Coverage rate are selected as the criteria for evaluating service recommendation quality.

Precision is to calculate the proportion of services selected by actual users in algorithm recommendation in all results of algorithm recommendation to evaluate the quality of recommendation. Accuracy is defined as formula (6):

$$P = \frac{\sum_{u \in U} |\text{TopK}(u) \cap \text{Test}(u)|}{\sum_{u \in U} |\text{TopK}(u)|}$$

(6)

$\text{TopK}(u)$ is the recommendation list provided by the recommendation algorithm for users; $\text{Test}(u)$ is the list of favorite services selected by users on the Test set.

Recall is to evaluate the quality of recommendations by comparing the results recommended by algorithms with the services selected by actual users. Recall rate is defined as formula (7):

$$R = \frac{\sum_{u \in U} |\text{TopK}(u) \cap \text{Test}(u)|}{\sum_{u \in U} |\text{Test}(u)|}$$

(7)

F-measure as a comprehensive index, balances the influence of Recall and Precision, and evaluates a recommendation result comprehensively. F-measure value is defined as formula (8):

$$F - \text{measure} = \frac{2PR}{P+R}$$

(8)

Where, $P$ and $R$ indicate the accuracy and recall rate.

Coverage is an important indicator to evaluate the diversity of recommendation results of recommendation systems, which is positioned as the proportion of recommended services and services used by recommendation systems. The greater the coverage rate, the better the diversity of recommendation algorithms. The coverage rate is defined as the calculation formula (9):

$$\text{Coverage} = \frac{|U \cup \text{Test}(u)|}{|S|}$$

(9)

Where $U$ is a set of all users of the system, $S$ is a set of all services, and $R_u$ is a set of services recommended by the recommendation algorithm to users $U_i$. 
4.3. Experimental results. The lowest threshold R is used to determine whether there is a relationship between service event classes in the travel service scenario model. There are three main settings for R value: average value, median value, and sum of maximum value and minimum value divide by 2. In the experiment, the average value is R = 3, the median value is R = 2, and the sum of the maximum value and the minimum value divided by 2 is R = 4. The individual experimental results of the scenario model are shown in Table 1 below, and the experimental results show that the accuracy, recall rate and F value are better when the lowest threshold R is taken as the average value.

| R/ N     | precision | recall | F-measure |
|----------|-----------|--------|-----------|
| R = 3, N = 2 | 0.724    | 0.301  | 0.425     |
| R = 3, N = 3 | 0.645    | 0.363  | 0.465     |
| R = 3, N = 4 | 0.520    | 0.465  | 0.492     |
| R = 3, N = 5 | 0.468    | 0.482  | 0.475     |
| R = 2, N = 2 | 0.680    | 0.281  | 0.397     |
| R = 2, N = 3 | 0.591    | 0.316  | 0.411     |
| R = 2, N = 4 | 0.512    | 0.380  | 0.436     |
| R = 2, N = 5 | 0.453    | 0.402  | 0.426     |
| R = 4, N = 2 | 0.764    | 0.275  | 0.406     |
| R = 4, N = 3 | 0.712    | 0.306  | 0.428     |
| R = 4, N = 4 | 0.633    | 0.342  | 0.444     |
| R = 4, N = 5 | 0.560    | 0.358  | 0.437     |

However, the weights $\alpha$ and $\beta$ in Similarity Calculation satisfy $\alpha + \beta = 1$, so we need to determine the optimal $\alpha$ and $\beta$ values through experiments to obtain the best experimental results. With $\{\alpha = 0.7, \beta = 0.3; \alpha = 0.8, \beta = 0.2; \alpha = 0.9, \beta = 0.1\}$ three groups of different value combinations, get the performance of F-measure values based on the recommendation of event ontology under different values.

In addition, this paper uses the traditional content-based recommendation algorithm, user-based collaborative filtering recommendation algorithm and project-based collaborative filtering recommendation algorithm as comparative experiments, the nearest neighbors of the latter two benchmark methods are also set to 5, 8, 10, 12, and 15. Compared with traditional recommendation methods, the final comparison results are shown in figure 3.

The final experimental results show that when $\alpha = 0.8$ and $\beta = 0.2$ and the number of recommendations is 12, the service recommendation based on event ontology has better
recommendation effect than the other three methods, including accuracy, both recall rate and F value have better performance.

In order to test the diversity of service recommendation based on event ontology proposed in this chapter, we introduce coverage rate for experimental verification. As shown in figure 4, the coverage rate of traditional content-based recommendations is very low. This is because content-based recommendation mainly depends on the similarity between services. Poor diversity results in low coverage. It can be seen from the experiment that the coverage rate of Web service recommendation based on event ontology proposed in this chapter is higher than that of the other three recommendation methods, thus improving the diversity of recommendation.

5. Conclusion
This paper proposes a Web service recommendation method based on event ontology, the recommendation method consists of Web service recommendation combined with service event ontology and content similarity calculation and Web service recommendation based on event context. Compared with other service recommendation methods, it can better solve the shortcomings of less QoS data and instability, and reduce the problem of semantic mismatch. The experimental part verifies that this paper can better improve the accuracy and diversity of service recommendation.

Acknowledgments
Supported by "construction and application of event relationship analysis model" (SKX192010019), university research project of 54th Research Institute of China Electronics Technology Group

References
[1] Gao M , Liu K , Wu Z . Personalisation in web computing and informatics: Theories, techniques, applications, and future research[J]. Information Systems Frontiers, 2010, 12(5):607-629.
[2] Lehrer C , Wieneke A , Vom Brocke J , et al. How Big Data Analytics Enables Service Innovation: Materiality, Affordance, and the Individualization of Service[J]. Journal of Management Information Systems, 2018, 35(2):424-460.
[3] Li S, Wen J, Luo F, et al. A New QoS-Aware Web Service Recommendation System based on Contextual Feature Recognition at Server-Side[J]. IEEE Transactions on Network & Service Management, 2017, PP(99):1-1.
[4] Bagga P , Joshi A , Hans R . QoS based Web Service Selection and Multi-Criteria Decision Making Methods[J]. International Journal of Interactive Multimedia and Artificial Intelligence, 2017, InPress(InPress).
[5] Blanquer J , Bruno J , Gabber E , et al. Resource Management for QoS in Eclipse/BSD[M]// Wireless Communications and Networking for Unmanned Aerial Vehicles. 2020.
[6] Yin Y , Chen L , Xu Y , et al. QoS Prediction for Service Recommendation with Deep Feature Learning in Edge Computing Environment[J]. Mobile networks & applications, 2020, 25(2):391-401.
[7] Ling-Lan G U , Department C E . Web service recommendation method based on context[J]. Computer Engineering and Design, 2014.
[8] Kang G, Tang M, Liu J, et al. Diversifying Web Service Recommendation Results via Exploring Service Usage History[J]. IEEE Transactions on Services Computing, 2016, 9(4):566-579.