Web Service Recommendation Technology Based on Knowledge Graph Representation Learning

Xinghao Qiao\textsuperscript{1,a} Zhiying Cao\textsuperscript{2,b,*} Xiuguo Zhang\textsuperscript{3,c}

\textsuperscript{1}Dalian Maritime University
Dalian, China
\textsuperscript{2}Dalian Maritime University
Dalian, China
\textsuperscript{3}Dalian Maritime University
Dalian, China

\textsuperscript{a}844194829@qq.com \textsuperscript{b}czysothy@dlmu.edu.cn \textsuperscript{c}zhangxg@dlmu.edu.cn

ABSTRACT: This paper proposed a recommendation algorithm based on knowledge graph representation learning (RABKGR). The algorithm embeds the entities and relationships of the knowledge graph into the low-dimensional vector space. The relationship information of services is incorporated into the recommendation algorithm by calculating the distance between the service entities. The association between services that is not considered when using the collaborative filtering algorithm can be supplemented, and the recommendation effect is enhanced. The experimental results show that this algorithm can not only effectively improve the accuracy rate, recall rate and coverage rate of recommendation, but also solve the cold start problem to some extent.

CCS Concepts
\begin{itemize}
\item Information systems--Recommender systems
\item Computing methodologies--Knowledge representation and reasoning
\item Information systems--Service discovery and interfaces
\end{itemize}

1. INTRODUCTION

As an important branch of personalized service research field, the recommendation system helps users find out the items they might be interested in from a large amount of data and generate personalized recommendations to meet individual needs.

Web services are the foundation of networked application software development. Networked software composed of them has been gradually an important form and development trend of software. However, with the rapid development of information technology in the era of information overload, how to find the service of interest from a large number of services has become an important issue in the field of service computing, which is to be solved by service recommendation [1].

Currently, most mainstream service recommendation algorithms are generally based on collaborative filtering (CF) recommendations.

Due to the perceptibility of quality of service (QoS), it has also become the basis for mainstream service recommendations. Many service recommendation systems use collaborative filtering algorithms to mine the QoS attribute information observed by users, find users who are similar to the current requester, and predict the unused service QoS attribute values for the current requester.
The Web service recommendation methods above all have the cold start problem existed in collaborative filtering itself, and the problem of service semantic information is not considered as well. They cannot meet users' needs for follow-up services and related services after using services.

In order to solve above problems, this paper proposes the RABKGRl, which fully considers the semantic relationship between Web services. The cold start problem is solved in a certain extent. By calculating the distance between the services used by the user through knowledge graph representation learning, we can get user classification. Finally, the recommended services are sorted according to the user's QoS preference, and a better recommendation set is obtained.

2. RELATED WORK

2.1 Research Status Of Knowledge Graph Representation Learning
The goal of representation learning is to represent the semantic information of the research object as a dense low-dimensional real-value vector through machine learning.

The main methods of knowledge representation learning include, single layer neural network[3], matrix decomposition model[4] and translation model[5]. Due to the huge performance improvement of the translation model, the translation model becomes the representative model of knowledge representation learning. The TransE algorithm proposed by Bordes A et al. [6] is the most representative algorithm in the translation model.

2.2 Research Status Of Web Service Recommendation
Zheng Zibin et al [7,8] proposes a recommendation system based on user feedback collaborative filtering algorithm (WSRec algorithm), which used collaborative filtering algorithm to predict QoS values, and proposed linear combination of user-based collaborative filtering and item-based collaborative filtering to improve the accuracy of QoS prediction. However, there are limitations in the recommendation system caused by the use of collaborative filtering algorithms, that is, sparse data seriously affects the accuracy of QoS prediction.

Sreenath [9], Karta [10] and others proposes using user-based CF to recommend services, but this scheme only considers user similarity, lacking the comprehensiveness of recommendation.

Ma et al. [11] predicts QoS values by tensor decomposition, adding user QoS preferences to implement service recommendation through service score.

Herlocker et al. [12] proposes a K-nearest neighbor (KNN) model based on collaborative filtering, namely UPCC algorithm, which is the most popular collaborative recommendation algorithm at present. However, it has serious problems of data sparseness and cold start. For new users, new services cannot be given reasonable recommendation.

3. SERVICE RECOMMENDATION ALGORITHM BASED ON KNOWLEDGE GRAPH REPRESENTATION
This paper proposes the RABKGRl. The basic idea is to firstly analyze the entity relationship of Web service knowledge graph. The entities in the graph are Web services and users. The relationship between services is that the output of the triplet header service satisfies the input of the tail service. User and service are linked by scoring. The score has five levels, which are recorded as 1-5 points corresponding to five relationships. The entities in the knowledge graph are embedded into a low-dimensional space by TransE algorithm and the specific optimal dimension is obtained experimentally. Then, the service relevance and the user similarity are calculated by the similarity formula. The nearest \( K \) users are neighbor users of the user to be recommended. After the neighbor users and the associated services are obtained, the semantic similar services of high user scoring service form the recommended set \( RS \) while the high scoring services of the neighbor users and its semantic similar service form the recommended set \( US \). Through the user QoS preference calculated, the services in \( US \) and \( RS \) are sorted to form an ordered sequence \( L \) and \( K \). Combine \( L \) and \( K \) in proportion of \( p:q \). The proportion is adjusted by classifying the user behavior, that is, whether the user to be recommended is
an active user or a negative user. Finally, the final ranking is performed according to the user’s QoS preference to form a final recommendation set C.

The RABKGRL proposed in this paper can solve the cold start problem well by utilizing the semantic information of knowledge graph. The algorithm fully considers user behavior and its preference factors in recommendation, which can improve the recommendation effect.

3.1 Web Service Knowledge Graph Representation Learning

The knowledge graph is a network of knowledge sets consisting of triples and links between triples, which carry the semantic information of the entity itself. Each entity acts as a node and the relationship between the entities acts as an edge. For the Web service knowledge graph, the entities in the graph are Web services and users. The relationship between services is a dependency relationship, that is, the output of the triplet header service satisfies the input of the tail service. Users and services are linked by scoring. There are five levels of scoring that 1-5 points correspond to the following relationship: Extremely dissatisfied, dissatisfied, Medium satisfaction, satisfaction and very satisfaction.

Based on the knowledge graph, the TransE algorithm is used to embed the entities and relationships in the knowledge graph into a low-dimensional vector space.

For the triples \((h, r, t)\) in the knowledge graph \(G\), \(l_h\), \(l_r\) and \(l_t\) represent vectors corresponding to \(h\), \(r\) and \(t\). TransE expects to get the formula (1).

\[
\|l_h + l_r - l_t\| = 1 \tag{1}
\]

Train using the loss function shown in formula (2).

\[
L = \sum_{(h, r, t) \in K} \sum_{(h', r, t') \in K'} \max(0, x + d(h + r, t) - d(h', r, t')) \tag{2}
\]

\(E\) is a set of entities in knowledge graph, and \((h, r, t)\) is the correct triple from the training set. \(K'\) is a set of error triples. \(x\) is the separation distance between the score of the legal triple and the score of the wrong triple, generally set to 1. The error triple is not randomly generated. In order to select a representative error triple, TransE randomly replaces one of the header and tail entities of each triple in \(K\) with other entities to get \(K'\).

\([x]_+\) represents the hinge loss function. Formal description as formula (3).

\[
[x]_+ = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases} \tag{3}
\]

TransE uses the idea of maximum spacing to train by pulling the distance between the right and wrong samples. The loss function is optimized by a stochastic gradient descent algorithm. The resulting vector space has the following characteristics, the closer the semantic distance of the knowledge graph is, the closer the entity distance is [13].

Since TransE calculates the loss function by Euclidean distance, the Euclidean distance is used to measure the similarity of the entities, and the entity similarity is defined as formula (4).

\[
\text{sim}(A, B) = \frac{1}{\|A - B\| + 1} \tag{4}
\]

\(A\) and \(B\) are two entity vectors in the knowledge graph. The higher the correlation between the entity vectors \(A\) and \(B\) is the closer the two entities are in the knowledge graph.

Find the set \(U\) of the user's neighbors by the similarity calculation. For every neighbor user find out his highly rated services to get the recommended service set \(RS\).
3.2 User QoS Preference Calculation and User Behavior Classification

3.2.1 QoS preference calculation
QoS is a key factor affecting users' evaluation of a service. Different users are sensitive to different QoS factors. Therefore, calculating the user's preference for different QoS attributes can make recommendation service more in line with the user's requirements.

Suppose that \( u_m \) has awarded ratings to \( N \) Web services and there are a total of \( K \) QoS properties for each Web service.

Let \( R_n = (r_1, r_2, ..., r_N)^T \) denote the rating vector of \( u_m \), where \( r_n \) denotes the rating awarded by \( u_m \) for service \( n \). \(^T\) denotes the transposed vector.

Let \( Q_n = (q_{nk})_{N \times K} \) denote the QoS matrix of \( u_m \), \( q_{nk} \) denotes the value of the k-th QoS property of service \( n \) which is observed from \( u_m \).

Let \( P_m = (p_1, p_2, ..., p_K)^T \) denote the preference vector of \( u_m \), \( p_k \) denotes \( u_m \)'s preference for k-th QoS attributes \([14]\).

For \( R_n \), \( Q_n \), \( P_m \) has the relationship expressed by formula (5).

\[
R_n = Q_n \cdot P_m \tag{5}
\]

\( R_n \) and \( Q_n \) are known, \( P_m \) is unknown and is required to be computed. According to the principles of linear algebra, we can solve \( P_m \) easily. Because there are generally not too many QoS properties. However, \( P_m \) should not be solved using linear equations. The noise in the QoS data and the noise in the rating data still present a problem and will become overfitting. So define the loss function as shown in formula (6).

\[
F = \| R_n - Q_n \cdot P_m \|^2 + \alpha \| P_m \|^2 \tag{6}
\]

\( \alpha \| P_m \|^2 \) is the regularization term to avoid overfitting. Specify training accuracy by \( F \leq \xi \). \( P_m \) can be solved by gradient descent algorithm. The specific process is as follows:

1. Randomly initialize the values of the parameters in \( P_m \).
2. Compute the partial derivative of \( F \) for \( p_k \), and update parameters. Formula (7) for partial derivatives and formula (8) for parameter updates.

\[
\frac{\partial F}{\partial P_k} = \sum_{n=1}^{N} (2q_{nk} - 2r_nq_{nk} + 2q_{nk} \sum_{j=1}^{k} q_{nj}P_j) \tag{7}
\]

\[
p_k = p_k - \tau \frac{\partial F}{\partial P_k} \tag{8}
\]

Where \( \tau \) is the iteration step size for solving \( P_m \).
3. If \( F > \xi \) back to 2; else if \( F \leq \xi \), then terminate the computation and \( P_m \) is determined.

3.2.2 User behavior classification
Different users have different acceptance of services. One type of users prefer to use the good services that they have experienced and related services, called negative users; The other type of users are open to new services and are willing to try new services, called positive users. In the knowledge graph, the distance between entities represents the semantic similarity of the entities. By calculating the distance of the Web service entities in the graph, the degree of association between the two services can be judged. Therefore, the user category can be effectively determined by the average distance between all the services used by the user.

Let \( S_m = (s_1, s_2, ..., s_n) \) denote the set of services that \( u_m \) has used. According to above user definition and the distance of the entity vector in the knowledge graph, the user behavior is judged by the formula (9).
\[
\overline{d} = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} || s_i - s_j ||
\]

(9)

\( \overline{d} \) represents the average distance of the Web services used by the user in the knowledge graph, which is used to determine whether it is an active user or a negative user.

3.3 Recommendation Set Fusion

By formula (4) the neighbor user high evaluation service set US and the high-scoring service association service set RS of the user to be recommended are obtained. Predictive scores are obtained by multiplying the calculated user preferences \( p_m \) with the predicted QoS attribute values of services. The predicted value \( Q \) of QoS is the average of the QoS evaluation of the service used by the neighbor users. Sort Web services in US and RS by predictive scoring to get \( L, K \). The services in the US are far away from the services used by the user to be recommended, the active user prefers to be recommended more services in the US. The services in RS are less distant from the services used by the user to be recommended, the negative users prefer to be recommended more services in RS. Assume that the fusion ratio is \( p:q \), the length of \( L \) is \( l \), and the length of \( K \) is \( k \), the fusion length \( n \) is formula (10).

\[
n = \begin{cases} 
\frac{lp}{p+q} & \text{a is negative user} \\
\frac{lp}{p+q} & \text{a is positive user} 
\end{cases}
\]

(10)

4. EXPERIMENTS

4.1 Experimental Data Set

In order to obtain real Web services, 850 travel-related and map-related Web services are captured on multiple open platforms through web crawlers. Travel-related services may be used during the tour, including hotel services, ticketing services, attraction services, raiders services, etc. Analyze the WSDL of the service, the dependencies between services are obtained by input and output keyword matching. At the same time, the rating information of 300 users and the evaluation of the two QoS attributes of response time and reliability are obtained. The user's QoS evaluation is stored in the user entity of the knowledge graph by means of entity attributes. Finally, through the above service relationship and user rating correspondence, the Web service knowledge graph required by the experiment is obtained.

4.2 Evaluation Standard

For the recommendation results, the Precision Rate(P), Recall Rate(R) and Coverage Rate(C) indicators are used for analysis. Define services used by less than ten users as new services. Coverage rate refers to the coverage of new services in recommendation results. Coverage is used to judge the degree of solving the cold start problem. The indicators are defined as:

\[
P = \frac{|L_r \cap L_u|}{L_u}
\]

(11)

\[
R = \frac{|L_r \cap L_u|}{L_r}
\]

(12)

\[
C = \frac{|C_u|}{|S_u|}
\]

(13)

\( L_r \) represents the list of services actually used by the user, \( L_u \) represents the recommended list of services. \( C_u \) represents new services in recommendation list, \( S_u \) represents new services in knowledge graph.
4.3 Experimental Comparison and Verification

Vector representation of Web knowledge graph entities can be got by representation learning. After calculating the average distance of services used by each user, determine when the parameter in formula (11) $d \leq 0.2$ the user is negative user, when $d > 0.2$ the user is positive user.

4.3.1 Fusion ratio determination

The proportional fusion experiment was performed with the learning embedding dimension of 200 and the number of recommended set $C$ is 30. Table 2 shows the Precision(P) and Recall(R) for each ratio when the user classification was not performed.

Table 1. P, R, F values for each proportion when no user classification

| Fusion Ratio (p:q) | P   | R   |
|------------------|-----|-----|
| 0:10             | 0.462 | 0.345 |
| 1:9              | 0.497 | 0.373 |
| 2:8              | 0.515 | 0.386 |
| 3:7              | 0.535 | 0.401 |
| 4:6              | 0.561 | 0.420 |
| 5:5              | 0.540 | 0.405 |
| 6:4              | 0.521 | 0.391 |
| 7:3              | 0.487 | 0.365 |
| 8:2              | 0.456 | 0.342 |
| 9:1              | 0.421 | 0.316 |
| 10:0             | 0.401 | 0.301 |

After user classification, $p>q$ for active users, $p<q$ for negative users. The results of user classification are shown in Table 2.

From Table 1 and Table 2, it can be seen that the performance of P and R values after user classification is better than that before classification. The optimal fusion ratio is 6:4 for active users and 4:6 for passive users.

Table 2. P, R, F values for each proportion when user classification

| NU Fusion Ratio (p:q) | PU Fusion Ratio (p:q) | P   | R   |
|---------------------|-----------------------|-----|-----|
| 0:10                | 10:0                  | 0.477 | 0.356 |
| 1:9                 | 9:1                   | 0.505 | 0.379 |
| 2:8                 | 8:2                   | 0.527 | 0.395 |
| 3:7                 | 7:3                   | 0.562 | 0.421 |
| 4:6                 | 6:4                   | 0.587 | 0.441 |
| 5:5                 | 5:5                   | 0.540 | 0.405 |

4.3.2 Embedded dimension determination

Knowledge graph representation learning is to embed entities into low-dimensional vector space, so it will have different effects for different embedding dimensions. The experimental results of 100-500 dimensions are shown in Table 4.

Table 3. P, R values corresponding to different embedding dimensions

| dimension | P   | R   |
|-----------|-----|-----|
| 100       | 0.580 | 0.435 |
| 200       | 0.587 | 0.441 |
| 300       | 0.582 | 0.437 |
| 400       | 0.578 | 0.434 |
It can be seen from Table 3 that the recommended effect is best when the dimension is 200 dimensions.

4.3.3 Algorithm comparison
Experiments were performed with an optimal dimension of 200 and an optimal ratio of 6:4, 4:6. The following algorithms are selected to be compared with the RABKGRL:
- UPCC: Research on the neighbor model, KNN model.
- IPCC: Research on product-based KNN model.
- WSRec: Improved collaborative filtering prediction algorithm and integrated information on similar users and similar services.
- Pearson-CF: Collaborative filtering service recommendation algorithm based on Pearson coefficient.

The recommended results for the comparison algorithms are mainly affected by the QoS matrix data density (D). Therefore the experiment will set three different densities 10%, 20%, 30%. Figure 1 shows the accuracy rate comparison compared with RABKGRL.

Figure 2 shows the comparison of recall rates.

In order to compare the cold start problem, inject different numbers of new services into the experimental data and compare algorithm coverage rate. Figure 3 shows the comparison of algorithm new service coverage rate.

It can be seen from Figure 1, Figure 2, Figure 3 that the proposed algorithm improves the accuracy rate, recall rate and coverage rate. And the influence of data density on the algorithm is small. This algorithm solves the problem of cold start better.

![Figure 1. Comparison of algorithm accuracy rate.](image1)

![Figure 2. Comparison of algorithm recall rates.](image2)
5. CONCLUSION
This paper proposes a Web service recommendation algorithm based on knowledge graph representation learning. Both the dependency information of the service and the user’s QoS evaluation for the service are considered. And the user’s behavior classification is considered as well. The algorithm embeds the entity into the low-dimensional space through knowledge graph representation learning. Calculate similarities between entities to find neighbor users and associated services and classify users by the distance between services used, which improves the accuracy of traditional Web service recommendations. At the same time, using the associated information of the services in the knowledge graph, the cold start problem is solved to some extent.

Experiments show that the RABKGRML can improve the effect of recommendation algorithm and have certain effect on the solution of cold start problem.

ACKNOWLEDGMENTS
This work is supported by the National Natural Science Foundation of China (Grant No. 51479021, 51679025) and the Fundamental Liaoning Provincial Natural Science Foundation of China(Grant No.20170520196).

REFERENCES
[1] Zhang X, He K, Wang J, et al.2013. A Survey of Web Services Personalized Recommendations[J]. Computer Engineering and Science.35(9):132-140.
[2] Mikolov T , Sutskever I , Chen K , et al. 2013.Distributed Representations of Words and Phrases and their Compositionality[J]. Advances in Neural Information Processing Systems.
[3] Bordes A,Weston J, Collobert, R. Bengio Y .2011. Learning Structured Embeddings of Knowledge Bases. Aaai Conference on Artificial Intelligence. DBLP.
[4] Socher R , Chen D , Manning C D , et al. 2013.Reasoning With Neural Tensor Networks for Knowledge Base Completion[C]// International Conference on Neural Information Processing Systems. Curran Associates Inc.
[5] Nickel M, Tresp V, Kriegel H P.2011. A three-way model for collective learning on multi-relational data[C]// International Conference on International Conference on Machine Learning.
[6] Bordes A., Usunier N., Garcia-Duran A., et al.2013.Translating embeddings for modeling multi-relational data[C]//Proc of NIPS.Cambridge,MA:MIT Press.2787-2795.
[7] Zheng Z, Ma H, Lyu M R, et al.2009. Wsrec: A collaborative filtering based web service recommendersystem C]. Proceeding of the 7th IEEE International Conference onon Web Services. 437-444.
[8] Zhen W, Zhang J, Feng J, et al. 2014.Knowledge graph embedding by translating on hyperplanes[C]// Twenty-eighth Aaai Conference on Artificial Intelligence.
[9] Sreenath R M,Singh M P .2004.Agent-based service selection[J].Web Semantics:Science,Services and Agents on the World Wide Web.1(3):261-279.
[10] Karta K .2005.An investigation on personalized collabororative filtering for web service selection[J].Honours Programme thesis,University ofWestern Australia,Brisbane.
[11] You M A, Xin X, Wang S, et al. 2015. QoS Evaluation for Web Service Recommendation[J]. China Communications. 12(4):151-160.

[12] Herlocker J L, Konstan J A, Borchers A, et al. 1999. An algorithmic framework for performing collaborative filtering[C]// Sigir 99: International Acm Sigir Conference on Research & Development in Information Retrieval. DBLP,

[13] Salakhutdinov R, Mnih A. 2007. Probabilistic Matrix Factorization[C]// International Conference on Neural Information Processing Systems.

[14] Salakhutdinov R, Mnih A. 2007. Probabilistic Matrix Factorization[C]// International Conference on Neural Information Processing Systems.