

Article

Collaborative Web Service Discovery and Recommendation Based on Social Link

Lijun Duan ¹  and Hao Tian ^{2,*} 

¹ School of Computer, Hubei University of Education, No. 129, Gaoxin Second Road, Wuhan 430205, China; duanlijun@hue.edu.cn

² School of Information Engineering, Hubei University of Economics, No. 8, Yangqiaohu Ave., Jiangxia Dist., Wuhan 430205, China

* Correspondence: haotian@whu.edu.cn; Tel.: +86-027-8197-2302

Received: 13 September 2017; Accepted: 9 October 2017; Published: 13 October 2017

Abstract: With the increasing application of web services in varying fields, the demand of effective Web service discovery approaches is becoming unprecedentedly strong. To improve the performance of service discovery, this paper proposes a collaborative Web service discovery and recommendation mechanism based on social link by extracting the latent relationships behind users and services. The presented approach can generate a set of candidate services through a complementary manner, in which service discovery and service recommendation could collaborate according to the formalized social link. The experimental results reveal that the proposed mechanism can effectively improve the efficiency and precision of Web service discovery.

Keywords: web service; discovery; recommendation; social link

1. Introduction

Web service can realize interoperable interactions between different machines via standard interfaces and communication protocols without the aid of additional third-party software or hardware. As an important innovation in service computing, more and more Web services are developed and published to the Internet. Therefore, service discovery is becoming a critical problem in service application. According to a survey [1], although there are a huge number of services advertised on the Web, about 75% of them have not been used, and only 16% of these services have been discovered or invoked. The main reasons are: most existing discovery methods only consider services as isolated functional islands with no links to related services, and they usually deal with each user and each request in isolation instead of mining and utilizing the latent social relationship among them [2].

To address these problems, in this paper, we propose a new method that differs from existing approaches in that: (1) we pay more attention to the potential social relations between users and services in the whole process of Web service discovery; (2) we redefine and formalize the social link through mining and defining typical link factors; and (3) we present a collaborative mechanism with high flexibility and satisfied performance. In this way, our proposal tackles the opportunity of exploiting social links to improve the performance of service discovery. Experimental results show that compared with similar methods, the proposed method has higher efficiency and precision.

The rest of this paper is organized as follows. Section 2 discusses the application and significance of social information in service discovery. Section 3 details our definitions about link factors and the formalization of social link. A collaborative discovery and recommendation mechanism based on social link is described in Section 4 before conducting the experimental evaluations in Section 5. The conclusion is given in Section 6.

2. Related Work

From recent research, leveraging latent user social information in the process of Web service discovery to improve the performance of discovery result is a feasible and effective method. In [3], the concept of community and the metaphor of social networking were discussed to build a framework to managing Web services, in which a community was proposed to gather Web services that offer similar functionalities together, and a social networking was designed to capture all interactions that occur between services located in the same or separate communities. Z. Maamar et al. discussed the intertwining of social networks of users and social networks in [4]. They used the former to help users select the necessary services, and adopted the latter to permit extending these services with new services or maintaining the operation continuity in case of failure. To maximize the utilization of limited network resources, a hybrid service-demand discovery architecture for mobile social networks was introduced in [5], in which users can discover their relationships with others based on the historical data of movement. In order to address the problems of the isolation of services and the lack of social relationships in Web service discovery, Chen et al. presented a global social service network connecting the isolated service islands to enhance the services' sociability on a global scale in [6]. The network can be constructed following linked social service-specific principles based on complex network theories. In [7], Kalai et al. proposed a web service discovery process based on users' queries and their social profiles that can be extracted from their egocentric network. The process employs the best social friendships of the current user and the past invocation histories with satisfactory web services of his friends to generate satisfactory results. To limit the disadvantages of distributed approaches, a novel distributed reputation model was presented in [8], which treats multi-agent systems as social networks and leverages social relations among these agents.

Some typical social factors such as trust and clustering are also widely discussed in service discovery. A novel evaluation model for Web Service discovery by leveraging trust as an approach was introduced in [9]. The authors incorporated a trust management module into the standard Service Oriented Architecture and transformed a Web Service network to a small-world network based on the trust relationships of service entities. Then they proposed a trust evaluation model with an amendatory subjective logic. In [10], a trusted service discovery method based on trust and recommendation relationships was presented. In this method, two types of trust relationships and an open service network were both defined, and a dynamic service discovery algorithm was also proposed. Based on the intuition that a user's web service QoS usage experiences can be predicted by both the user's own characteristics and the past usage experiences of other similar users, Z. Zheng et al. proposed a neighborhood-integrated matrix factorization approach for making personalized prediction in service discovery and service selecting [11], which could systematically fuse the neighborhood-based and model based collaborative filtering approaches to achieve higher prediction accuracy. A new service discovery method based on referral network and ant-colony algorithm was proposed in [12]. The approach can make use of recommendation and ant colony algorithm to improve the success and recall rate of service discovery. Based on friend relationships and interaction behaviors, a new recommender approach was introduced in [13]. The approach adopted a trust computing method and depended on an authentic social network to obtain more authentic neighbors to improve the accuracy of recommendations. In order to acquire satisfactory performance-based service discovery/selection, Q. Duan suggested that it is important for both service providers and service consumers to obtain thorough understanding of performance of service provisioning [14]. To tackle the challenge, a new model for characterizing the service provisioning capabilities of converged systems was presented, and the analysis techniques for evaluating performance were also developed.

All the above studies show that it is a non-trivial task to leverage social information to improve the performance of service discovery. However, as mentioned above, most of the tasks are still troubled by low precision and low efficiency. Therefore, in this paper, we are motivated to investigate how social link could benefit Web services discovery by reducing the retrieval time and improving the accuracy of results.

3. Social Link

Different from the general definition in social network, in this paper, we define social link in another way and exploit it to capture the interaction relationships that influence the performance of Web service discovery, which are similar to those found in people's daily life such as reputation, clustering, preference and trust. More specifically, we decompose the social link in Web service discovery into four factors: reputation degree, clustering link, preference link and trust link, which can be defined as follows.

3.1. Link Factors

3.1.1. Reputation Degree

Ontology is essentially a conceptual framework that gives a set of terms to identify a set of concepts. In addition, domain ontology is a professional ontology that describes the concepts and the relationship between concepts in a particular field. In our previous study [2], we have proposed a user personalization model based on ontology technology. Therefore, for a specific user, by analyzing and refining his/her historical operation records, we can mine their interesting points which performance as local nodes of domain ontology. In other words, each personalization model is a sub-tree of an ontology tree.

For instance, if the numbers of ontology concepts called by the user u_a can be gained based on his/her operation history and the distribution of the numbers can be shown as Figure 1, in which the number above each box represents the number of calls to the concept in the box, it can be concluded that the total number of ontology concepts called by u_a is 75, then, u_a 's interestingness in each concept can be calculated, for example, the interestingness in concept "CompanyBond" is $12/75$, which is 0.16. It is worth noting that although the number of concept "InvestmentFund" called is zero, since it is the parent concept of concept "OpenEndFund", concept "ExchangeTradeFund" and concept "IndexFund", the interestingness in it is actually 0.34.

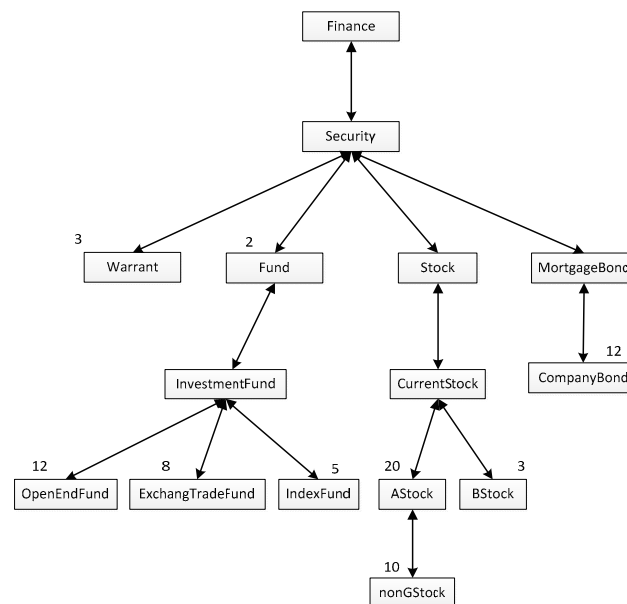


Figure 1. The distribution of the numbers of ontology concepts called by u_a .

Replacing the numbers of concept called in Figure 1 with the calculated interestingness, the user personalization model of u_a can be obtained, which is shown as Figure 2. Compared with Figure 1, it has the same tree structure, and the main difference is that there is a value above each concept box.

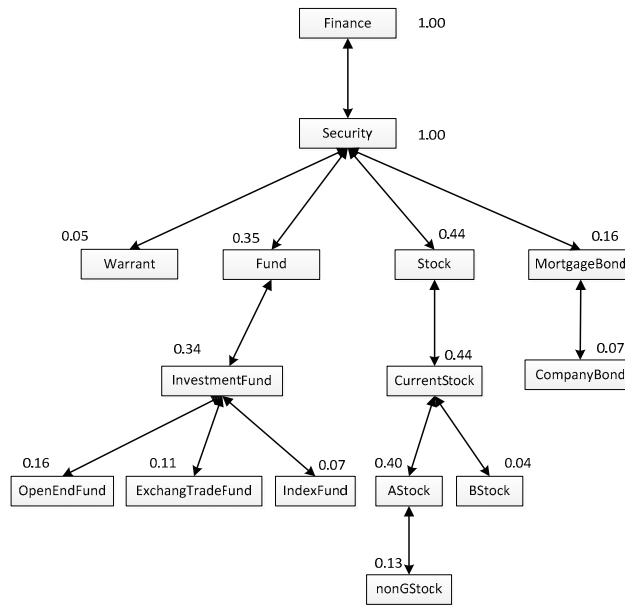


Figure 2. The user personalization model of u_a .

Based on the proposed user personalization model, we define a user's Reputation Degree (RD) is the proportion of the quantity of the concepts in his/her personalization model to the quantity of the whole concepts in the corresponding ontology.

$$RD_m = \frac{N_m}{N_o} \quad (1)$$

where, RD_m is the Reputation Degree of user u_m ; N_m means the quantity of the concepts in the personalization model of u_m ; N_o is the quantity of the whole concepts in the corresponding ontology. For example, if the personalization model of user u_a is shown as Figure 1, and the quantity of the whole concepts in the Finance Ontology is 100, that is, $N_a = 15$, and $N_o = 100$, then $RD_a = 0.15$.

3.1.2. Clustering Link

Clustering Link (CL) contains two sub-classes: Service Clustering Link (SCL) and User Clustering Link (UCL), the former is the set of services and their similarities, which is described as Equation (2).

$$SCL_i = \{(ws_j, SSim_{ij}) | SSim_{ij} > SSim_0\} \quad (2)$$

where, SCL_i denotes the Service Clustering Link of Web service ws_i ; $SSim_{ij}$ is the service similarity between ws_i and ws_j , and $SSim_0$ is the threshold of it.

A representation of a service specification was introduced in [15], which can be shown as following:

- A service profile $P = \langle sn; \odot \rangle$ consists of a service name sn and a set \odot of operations.
- An operation $O \in \odot$ has a name $O:n$ and a set $O:IN$ and $O:OUT$ of inputs and outputs: $O = \langle n; IN; OUT \rangle$. For example, $P:sn$ is the service name, and $P:\odot_i:IN_j$ is the j -th input of the i -th operation.

Based on the representation, we can calculate the similarity $SSim_{ij}$ by Equation (3).

$$SSim_{ij} = \omega_1 \times NSim(ws_i.sn, ws_j.sn) + \omega_2 \times FSim(ws_i.o, ws_j.o) \quad (3)$$

where, $ws_i.sn$ and $ws_i.o$ are respectively the name and the set of function properties of ws_i ; $NSim(\cdot)$ and $FSim(\cdot)$ are the similarity functions proposed in [2]; additionally, $0 < \omega_1, \omega_2 < 1$ and $\omega_1 + \omega_2 = 1$.

Similarly, User Clustering Link can be defined as the set of users and their similarities, which is described as Equation (4).

$$UCL_m = \{(u_n, USim_{mn}) | USim_{mn} > USim_0\} \quad (4)$$

In this equation, UCL_m denotes the User Clustering Link of user u_m ; $USim_{mn}$ is the user similarity between u_m and u_n , and $USim_0$ is the threshold of it.

In our view, we can get the value of a user similarity between two users by comparing their personalization models. For example, if there is another personalization model of u_b , which is shown as Figure 2, then, based on the pruning strategy proposed in [2], we can construct a matching tree between Figures 2 and 3, which is shown as Figure 4. Different from the personalization model, each box in the match tree has a concept pair instead of a single concept, and near to the box is a description of the concept pair. The description is always a triple such as “((AStock, HStock), 0.736, 0.08)”, specifically, the first element describes the two concrete concepts in the concept pair; the second one is the similarity between the two concepts; and the last one is the minimum interestingness in them.

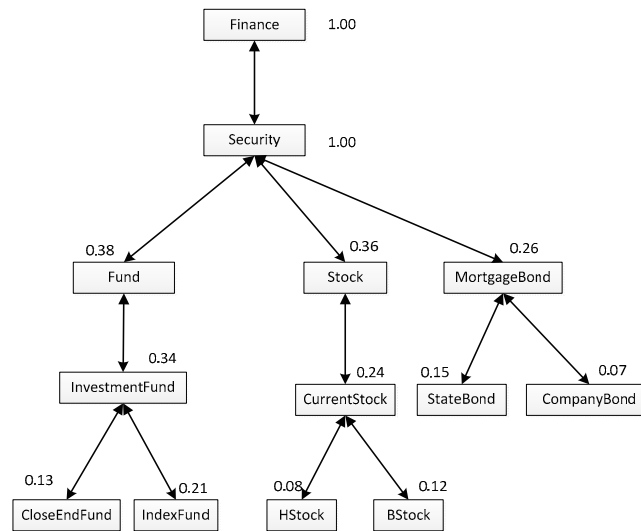


Figure 3. The user personalization model of u_b .

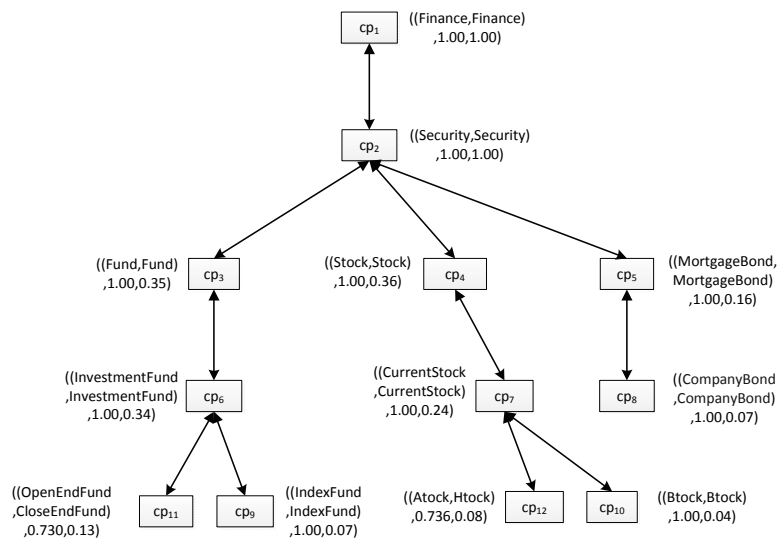


Figure 4. The matching tree between the user preference models of u_a and u_b .

Based on the match tree, we can calculate the value of the user similarity between users according Equations (5) and (6), for u_a and u_b , the user similarity $USim_{ab} = 0.68$.

$$USim_{ab} = \frac{2 \times FDO_{ab} \times \sum_{cp_k \in MT} (Intr(cp_k) \times Sim(cp_k) \times \frac{D(cp_k)}{DO_{MT}})}{RDO_{ab} \times (\sum_{c_i \in P_a} (Intr(c_i) \times \frac{D(c_i)}{DO_a}) + \sum_{c_j \in P_b} (Intr(c_j) \times \frac{D(c_j)}{DO_b}))} \quad (5)$$

$$Sim(cp_k) = \frac{D(par(cp_k)) + \sqrt{\frac{N(par(c_i)) \times N(par(c_j))}{N(par(c_i)) + N(par(c_j))}}}{S(cp_k) - 1 + D(par(cp_k)) + \sqrt{\frac{N(par(c_i)) \times N(par(c_j))}{N(par(c_i)) + N(par(c_j))}}}, cp_k = (c_i, c_j) \quad (6)$$

where, MT represents the matching tree between user preference models; $S(cp_k)$ is the distance between concept c_i and concept c_j ; $par(cp_k)$ is the parent node of cp_k ; $Intr(c_i)$ means the interestingness in concept c_i ; $D(cp_k)$ is the depth of cp_k ; DO_a signifies the depth of user personalization model P_a ; $N(c_k)$ denotes the density of concept c_k ; FDO_{ab} is the floor depth between user personalization model P_a and user personalization model P_b , and RDO_{ab} is the roof depth between them. Their definitions and calculations are shown in [2].

3.1.3. Preference Link

A user's Preference Link (PL) is described as Equation (7), it is defined as the set of services invoked by him/her and corresponding experience.

$$PL_m = \{(s_i, e_i^m) | e_i^m > e_0\} \quad (7)$$

where, PL_m means the Preference Link of user u_m ; e_i^m is the experience of u_m on service ws_i , it is essentially the rating value of ws_i by u_m , specifically, to formalize the experience, we normalize e_i^m with integers in the range from 1 to 5, $e_i^m = 5$ means the best experience and $e_i^m = 1$ denotes the poorest experience, and e_0 is the threshold of it.

3.1.4. Trust Link

A user's Trust Link (TL) is designed to formalize the relation between him/her and his/her trusted friends, it is the set of users that ever recommended service to him/her and corresponding trust degree, which is described as Equation (8).

$$TL_m = \{(u_n, TDgr_{mn}) | TDgr_{mn} > TDgr_0\} \quad (8)$$

For the user u_m , TL_m is his/her Trust Link; $TDgr_{mn}$ is the trust degree to user u_n , and $TDgr_0$ is the threshold of it. $TDgr_{mn}$ can be calculated by Equation (9).

$$TDgr_{mn} = \frac{1 + RD_n}{|S_R|} \sum_{ws_i \in S_R} \frac{e_i^m - e_i^n}{e_{\max}} \quad (9)$$

where, S_R is the set of services that u_n has recommended to u_m , and $|S_R|$ is the number of services in it; e_{\max} is the maximum value in the normalization range. Normally, $|S_R| \neq 0$, otherwise, the link between the two uses will be released.

3.2. Formalization of Social Link

Based on the above factors, we formalize the Social Link (SL) in Web service discovery as a tetrad shown as Equation (10).

$$SL = \langle RD, CL, PL, TL \rangle \quad (10)$$

Figure 5 shows an instance of Social Link. In this instance, there are two users: u_2 and u_3 , in the User Clustering Link of u_1 , and his/her Preference Link contains four services and, moreover, they belong to three different Service Clustering Links.

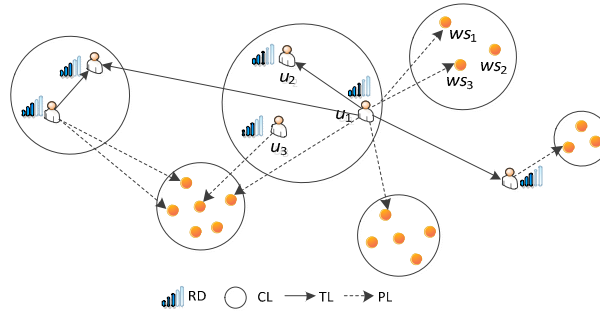


Figure 5. Instance of a formalized social link.

4. Discovery and Recommendation Mechanism Based on Social Link

In this section, we propose a Collaborative Web Service Discovery and Recommendation Mechanism (CSDRM) to further select or recommend a subset of services over SL. First we introduce the basic algorithm of service discovery based on Clustering Link and Preference Link (DCLPL). Then we present an auxiliary algorithm of service recommendation based on Reputation Degree and Trust Link (RRDTL).

4.1. DCLPL

Given a user request, DCLPL performs search starting from the user's Preference Link to his/her Clustering Link, to narrow the search scope gradually, while matching the latent candidates preferentially.

Algorithm DCLPL.

Input: user u_m 's request r

Output: candidate service list L_1

1. initialize u_m 's Social Link SL_m ;
 2. set $L_0 = \phi$ and $L_1 = \phi$;
 3. **for** each $ws_i \in PL_m$ **do** // query the service collection in the preference link
 4. calculate the similarity $RSim_i$ between ws_i and r ; // calculate and compare the matching degree
 5. **if** $RSim_i > RSim_0$ **do**
 6. add ws_i to L_1 // add the service into the candidate list
 7. **else**
 8. add ws_i and each $ws_l \in SCL_i$ to L_0 // add the service and its service collection in the clustering link into the unfit list
 9. **end if**
 10. **for** each $ws_j \in SCL_i \wedge ws_j \notin L_0$ **do** // query other services in the clustering link of current service and not into the unfit list
 11. repeat from step 4 to step 9
 12. **end for**
 13. **end for**
 14. **for** each $u_n \in UCL_m$ **do** // query the user collection in the current user's clustering link
 15. **for** each $ws_k \in PL_n \wedge ws_k \notin L_0$ **do**
 16. repeat from step 4 to step 12
 17. **end for**
 18. **end for**
 19. **for** each remained $ws_p \notin L_0$ **do** // query remained services
 20. repeat from step 4 to step 6
 21. **end for**
 22. **return** L_1
-

In the above algorithm, we calculate $RSim_i$ by Equation (11):

$$RSim_i = \omega_1 \times NSim(ws_i.sn, r.sn) + \omega_2 \times FSim(ws_i.o, r.o) \quad (11)$$

Similar to Equation (3), $r.sn$ is the user's demand for service name, and $r.o$ is the user's demand for service functions. Additionally, the value setting of ω_1 and ω_2 is same with Equation (3).

4.2. RRDTL

If the user's request conditions are too restrictive or the request statements are vague, L_1 will be empty, then, CSDRM will recommend a candidate service set by performing RRDTL algorithm.

Algorithm RRDTL

Input: u_m 's Social Link SL_m , user u_m 's request r

Output: candidate service list L_1

1. **if** $L_1 = \phi$ **do**
 2. **for each** $u_n \in TL_m$ **do** // query the user collection in the current user's trust link
 3. **for each** $ws_i \in PL_n$ **do** // query the service collection in each user's preference link
 4. calculate $R_i = RSim_i * (1 + RD_i)$; // calculate the reliability of each service
 5. **end for**
 6. add ws_i which has the maximum R_i to L_1 ; // select the most reliable service
 7. **end for**
 8. rank services in L_1 by R_i and remain Top- k services
 9. **return** L_1
 10. **end if**
-

5. Experiment and Analysis

To evaluate the performances of CSDRM, we conducted following experiments, in which we considered three comparison partners: Ant-Algorithm-Based Service Discovery Algorithm (ABSDA) [12], Trustworthy Service Discovery based on Trust and Recommendation relationships (TRTSD) [10] and Random Ergodic Matching (REM) [16]. ABSDA adopts recommendation mechanism in service discovery, TRTSD can reduce the searching path and search trusted services by exploiting trust and recommendation relationships, and they are partly similar to CSDRM on the principles of algorithms. REM is actually a flooding algorithm, and we select it to highlight the comparison effect.

5.1. Experimental Setup

We constructed the same experiment environment with [16], namely, we created 10 different requests, and performed the experiments with service number being [300, 400, 500, 600, 700, 800, 900, 1000] respectively. Eventually, we adopted the average efficiency (ACT) and the average precision (AP) that were defined in [16] as the metrics to compare the four methods. The average completion time refers to the time from matching the service according to the parsed request to returning the list of results, and the average precision can be described as Equation (12):

$$P_{avg} = \frac{n_T}{n_T + n_F} \quad (12)$$

where, n_T is the number of services that meet user's requirements in the returned results, and n_F is the number of services that cannot fully meet the user's requirements.

5.2. Parameters Evaluation

From the above analysis, it can be found that the specific values of some parameters will affect the performance of our proposed algorithm. Furthermore, since ω_1 is the weight of service name and

ω_2 is the weight of service operation, in our previous study [2], we have drawn a conclusion that $\omega_1 = 0.3$ and $\omega_2 = 0.7$ is reasonable. Thus, in order to evaluate the influence and decide the suitable value of these parameters, we compared the average efficiency and the average precision of CSDRM with different values of the key parameters in it. Here, we set $RSim_0 = 0.4$ and $k = 5$.

Table 1 provides the results of the average efficiency and the average precision of CSDRM with different values of the key parameters. We can observe that the greater the values of parameter $SSim_0$ and parameter $USim_0$, the lower the efficiency and the higher the precision of the algorithm, and in the case of invariant parameter $SSim_0$ and parameter $USim_0$, the higher the value of parameter $TDgr_0$, the higher the efficiency and the less change in precision. On the whole, when $SSim_0 = 0.5$, $USim_0 = 0.5$ and $TDgr_0 = 0.2$, the algorithm can achieve a better balance between efficiency and precision.

Table 1. Comparison with different values of the key parameters in CSDRM.

Key Parameters			ACT (ms)	AP
$SSim_0$	$USim_0$	$TDgr_0$		
0.3	0.3	0.2	11.2653	0.62
0.4	0.4	0.2	11.6689	0.64
0.5	0.5	0.2	12.1875	0.68
0.6	0.6	0.2	12.5635	0.69
0.7	0.7	0.2	12.7656	0.69
0.3	0.3	0.3	11.1254	0.62
0.4	0.4	0.3	11.2356	0.65
0.5	0.5	0.3	12.0145	0.67
0.6	0.6	0.3	12.2153	0.68
0.7	0.7	0.3	12.5332	0.68

5.3. Top-k Analysis

To analyze the effect of parameter k on the performance of the algorithms, we define a new precision metric:

$$P@k = \frac{n_k}{k} \quad (13)$$

where, k is the concrete value of Top- k , and n_k is the number of services that meet the user's requirements in the Top- k services.

In this experiment, we set $SSim_0 = 0.5$, $USim_0 = 0.5$, $TDgr_0 = 0.2$ and $RSim_0 = 0.4$, use the same requests and candidate services, and compare the average precision of the four approaches in the conditions of $k = 5, 8$ and 10 .

Table 2 shows the results of the average precision of the four approaches with different values of k . It can be seen that our proposed method achieves the best precision results among these four approaches from beginning to end. For the synthetic precision, CSDRM achieves 5.05~71.39% improvements. It is also clear that the methods work best when $k = 5$.

Table 2. Precision comparison of Top- k effect.

P@k	CSDRM	ABSDA	TRTSD	REM
P@5	0.6800	0.5200	0.6600	0.4400
P@8	0.6625	0.4875	0.6125	0.3625
P@10	0.5300	0.3900	0.5100	0.2900
Synthetic	0.6242	0.4658	0.5942	0.3642

5.4. Efficiency and Precision Comparison

We conducted this experiment to compare the efficiency and precision of these four methods, and based on the above analysis, we set $\omega_1 = 0.3$, $\omega_2 = 0.7$, $SSim_0 = USim_0 = 0.5$, $e_0 = 3$, $TDgr_0 = 0.2$, $RSim_0 = 0.4$ and $k = 5$.

Figure 6 shows the average efficiency. It is obvious that the completion time is getting longer as the number of services increases. Among these four methods, CSDRM costs the shortest completion time from beginning to end. Specifically, compared with the flooding algorithm REM, the completion time of CSDRM is improved by 50.53% on average; compared with the Ant-Algorithm-Based algorithm ABSDA, the completion time is increased by 33.58% on average; and compared with the trust and recommendation based algorithm TRTSD, and the completion time is increased by 29.91% on average. This is because CSDRM can rapidly narrow the scope of matching based on *SL*, discard irrelevant services. Comparing with the others, it has the minimum number of services required to be matched. The results indicate that CSDRM can improve the efficiency of Web service discovery effectively.

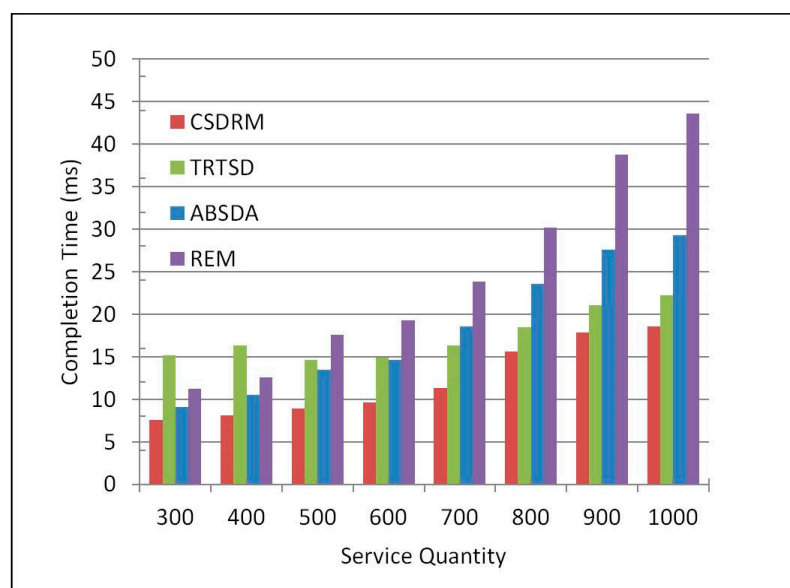


Figure 6. Comparison of efficiency.

Figure 7 describes the result of comparing the average precision of the four approaches. In the best case, CSDRM can achieve the precision of 72.23%. Meanwhile, the best Success Rates of TRTSD and ABSDA are 70.32% and 56.52% respectively, and the highest precision in REM is only 46.52%. Furthermore, when the service quantity is over 500, CSDRM is always the best one in the precision performing. On average, CSDRM improves the percentage of Success Rate by nearly 22% compared to REM. The results reveal that *SL* can also benefit the precision in Web service discovery.

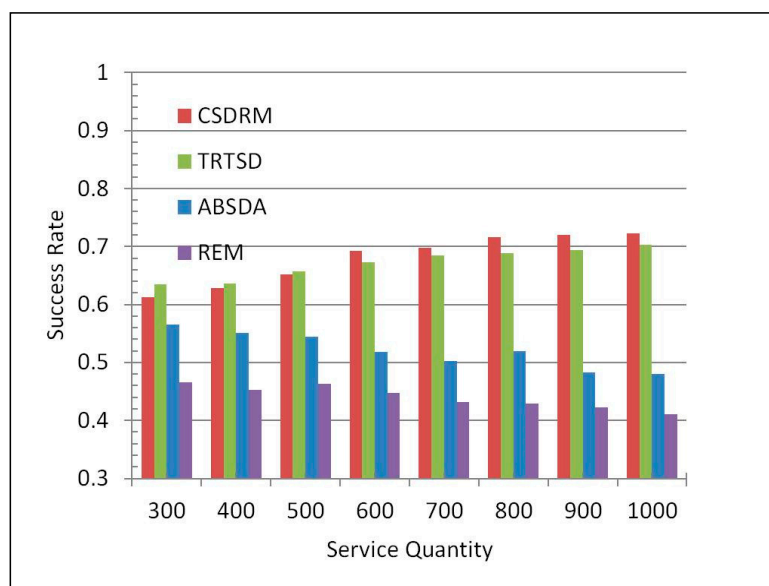


Figure 7. Comparison of precision.

6. Conclusions

In this paper, we present a collaborative web service discovery and recommendation mechanism grounded on the latent social relationships behind users and services. By defining four link factors, we formalize the social link. Based on it, we further propose two algorithms, which can collaborate to select or recommend a subset of services to meet the requirements of users. The experiment results have demonstrated that our proposed approach is capable of improving the efficiency and precision of Web service discovery. In the future, we plan to extend SL by taking into consideration more specific link factors, so as to make the mechanism more flexible and effective.

Acknowledgments: This work is supported by the Science Research Key Project of the Department of Education in Hubei Province, China under Grant No. D20162202.

Author Contributions: Lijun Duan and Hao Tian conceived and designed the experiments; Lijun Duan performed the experiments; Hao Tian analyzed the data; Lijun Duan contributed analysis tools; Hao Tian wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jiang, W.; Lee, D.; Hu, S. Large-scale longitudinal analysis of soap-based and restful web services. In Proceedings of the 2012 IEEE 19th International Conference On Web Services (ICWS), Honolulu, HI, USA, 24–29 June 2012; pp. 218–225.
2. Tian, H. *Research on User-Centered Web Service Discovery and Its Application in Financial Services*; Wuhan University: Wuhan, China, 2014.
3. Yahyaoui, H.; Maamar, Z.; Lim, E.; Thiran, P. Towards a community-based, social network-driven framework for Web services management. *Future Gener. Comput. Syst.* **2013**, *29*, 1363–1377. [[CrossRef](#)]
4. Maamar, Z.; Faci, N.; Sheng, Q.; Yao, L. Towards a user-centric social approach to Web services composition, execution, and monitoring. In Proceedings of the 13th International Conference on Web Information Systems Engineering, Paphos, Cyprus, 28–30 November 2012; pp. 72–86.
5. Wu, D.; Yan, J.; Wang, H.; Wang, R.Y. Service demand discovery mechanism for mobile social networks. *Sensors* **2016**, *16*, 1982. [[CrossRef](#)] [[PubMed](#)]
6. Chen, W.; Paik, I.; Hung, P.C. Constructing a global social service network for better quality of Web service discovery. *IEEE Trans. Serv. Comput.* **2015**, *8*, 284–298. [[CrossRef](#)]

7. Kalai, A.; Zayani, C.A.; Amous, I. User's social profile-based web services discovery. In Proceedings of the IEEE 8th International Conference on Service-Oriented Computing and Applications, Rome, Italy, 19–21 October 2015; pp. 2–9.
8. Kravari, K.; Bassiliades, N. DISARM: A social distributed agent reputation model based on defeasible logic. *J. Syst. Softw.* **2016**, *117*, 130–152. [[CrossRef](#)]
9. Liu, F.; Wang, L.; Gao, L.; Li, H.; Zhao, H.; Men, S. A Web service trust evaluation model based on small-world networks. *Knowl.-Based Syst.* **2014**, *57*, 161–167. [[CrossRef](#)]
10. Liu, Y.; Zheng, X.; Chen, D. Trustworthy services discovery based on trust and recommendation relationships. *Syst. Eng. Theory Pract.* **2012**, *32*, 2789–2795.
11. Zheng, Z.; Ma, H.; Lyu, M.; King, I. Collaborative Web service QoS prediction via neighborhood integrated matrix factorization. *IEEE Trans. Serv. Comput.* **2013**, *6*, 289–299. [[CrossRef](#)]
12. Xie, X.; Song, C.; Zhang, Z. A service discovery method based on referral network and ant-colony algorithm. *Chin. J. Comput.* **2010**, *33*, 2093–2103. [[CrossRef](#)]
13. Zhu, Q.; Sun, Y.Q. Collaborative filtering recommender method based on trust. *J. Tsinghua Univ. (Sci. Technol.)* **2014**, *54*, 360–365.
14. Duan, Q. Modeling and delay analysis for converged network-cloud service provisioning systems. In Proceedings of the 2013 International Conference on Computing, Networking, and Communications, San Diego, CA, USA, 28–31 January 2013; pp. 66–70.
15. Pirro, G.; Talia, D.; Trunfio, P. A DHT-based semantic overlay network for service discovery. *Future Gener. Comput. Syst.* **2012**, *28*, 689–707. [[CrossRef](#)]
16. Tian, H.; Fan, H.; Du, W. Research on Web service discovery based on user community relations. *J. Commun.* **2015**, *36*, 28–36.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).