WSRS: A WEB SERVICE RECOMMENDER SYSTEM

Esma Aïmeur, David Daboue, Flavien Serge Mani Onana
University of Montreal, Department of Computer Science, Montreal, Canada

Djamal Benslimane
Claude Bernard University, Lyon 1, Villeurbanne, France

Zakaria Maamar
Zayed University, Dubai, United Arab Emirates

Keywords: Profile, Recommender System, Similarity, Web service.

Abstract: Despite Web services widespread adoption, users still struggle with the problem of locating the Web services that best satisfy their needs and meet their requirements. Unfortunately, current Web services repositories suffer from various limitations, such as providing Web services to users regardless of these users' past experiences and these Web services' intrinsic characteristics like popularity and credibility. In this paper, we introduce a Web Service Recommender System that uses collaborative filtering, demographic filtering, and content-based filtering techniques to help facilitate the search of Web services. The WSRS system uses both users' and Web services' profile for the sake or recommending Web services.

1 INTRODUCTION

The W3C introduced Web services to provide a standard way of communication between software applications. Web services are technologies designed to support interoperable computer-to-computer interaction over networks.

Like a company wishing to advertise and make its activities more visible to prospective users, the need for publishing Web services for discovery needs draws much attention from several research bodies. To this end, UDDI enables applications to publish and find Web services. Indeed, today, it is a challenge for application designers and simple users to find the Web services that best satisfy their needs and meet their requirements. Usually, techniques to search for Web services in UDDI repositories are keyword-based. Tsalgatidou *et al.* (2002) argue that even if the WSDL/UDDI contained semantic information in connection with the providers and syntactic information in connection with the Web services, it remains difficult to decide how to find "relevant" Web services, starting from simple keywords. Therefore, there is a need to design systems that assist users locate Web services. In particular, such systems may take into account the user profile, which summarizes what a user likes and dislikes. This profile includes the user's demographic data, interests, preferences, usage records, purchase records, browsing behavior, etc. Usually, the profile is used for Web personalization, which is an Internet technique for adapting websites to individuals (Mobasher and Anand, 2005). Nowadays, there exist several ways to compile user profiles (Turban *et al.,* 2006). However, the different approaches should consider the "size and the heterogeneous nature of the data itself, as well as the dynamic nature of user interactions with the Web".

In this paper, we compile the profile of users and Web services in order to match Web services' capabilities to users' needs. Thus, we concentrate on the approach based on *recommender systems,* which make inferences from data provided by users on other issues or by analyzing similar users. Recommender systems allow entities providing items to guide the choices made by users. The recommendation can be issued by the service provider who possesses the items or by brokers. The

Aïmeur E., Daboue D., Serge Mani Onana F., Benslimane D. and Maamar Z. (2007).
WSRS: A WEB SERVICE RECOMMENDER SYSTEM.
In *Proceedings of the Third International Conference on Web Information Systems and Technologies - Internet Technology*, pages 284-289
DOI: 10.5220/0001288102840289

recommender system we introduce, here, is a search tool that can be anchored to any UDDI.

2 PROFILING APPROACH

2.1 How to Profile Users?

Usually, a Web user could be defined with three types of information: demographic, usage behavior (what the user bought in the past, for how much, etc.), and browsing behavior (*click stream*). The demographic information is provided by the user when he opens an account in the system being used. The user can then update this information during his subsequent visits. For the last type of information, as the user progressively browses the system's website, information about the items he is testing or using, is collected and then stored for further use. This kind of information is known as implicit feedback that the system gathers from the user's browsing behavior. Moreover, the user is usually invited to provide a rating on a scale for any items that he tests or uses. This rating is an explicit feedback that the system also stores for further use. Browsing behavior information and ratings are data that the system gathers on each user to enrich his profile and use later during recommendation. The usage behavior information is an implicit feedback gathered from the user's activities in the system. Each user profile contains a list of *Web services he used in the past*. Moreover, the user profile includes the *usage* percentage, which expresses the number of times that a given Web service has been accessed by the user, session after session. It represents the *fidelity* of the user in using a given Web service.

2.2 How to Profile Web services?

Besides its operations, input parameters and output, what are other characteristics that may distinguish a Web service from other peers? Moreover, this question leads to the notion of Web service profile. We suggest profiling a Web service with a set of characteristics, which could to a certain extent overlap with what is usually known as non-functional aspects of Web services: *Popularity (*number of times the Web service has been used with success), Response delay, Intelligibility (how comprehensible is a given Web service), Credibility, Lifetime, Updatability (does the Web service take into account users' feedback?).

This paper only focuses on *popularity* and an *average* rating that mixes the aforementioned characteristics (response delay, intelligibility, etc.).

3 RECOMMENDER SYSTEMS

The main interest in recommender systems is backed by the plethora of applications dealing with information overload so personalized content and services are provided to users A recommender system relies on an item's features and/or previous user ratings to provide an opinion or a list of selected items that assists the user in evaluating items that are not yet rated by him. Five main types of filtering techniques are identified in (Burke, 2002). We define below the three most used techniques, as we use them in our recommender system: Collaborative Filtering, Content-based Filtering, and Demographic Filtering.

3.1 Collaborative Filtering

Collaborative Filtering (CF) was first introduced by Tapestry's developers (Goldberg *et al.,* 1992). It accumulates user item ratings, identifies users with common ratings and offers recommendations based on inter-user comparison. CF techniques are becoming increasingly important in the context of e-commerce, with the unprecedented growth in the number of users.

In this paper, we use a well-known memory-based (as opposed to a model-based) (Breese *et al.*, 1998) CF technique, based on the *Nearest Neighbor* classification algorithm. However, our general approach to Web service recommendation is not tied to this choice. Here, data is represented by a matrix where entry $v_{u,i}$ represents the rating user u gave to item (Web service) i. This entry is set to null in case user u has not rated yet item i in which case this entry is not used in the computation. Suppose that the Web service provider's database, T, contains t items, $p_1, p_2, \ldots, p_t$, and that m users, $u_1, u_2, \ldots, u_m$, have rated some items from T. Rating predictions for a given user are produced in two stages, as we now review.

In order to estimate similarity between users, various metrics have been proposed. One of them, for example, is the *Pearson correlation* (Resnick *et al.*, 1994). The results obtained range from -1 for negative correlation to +1 for perfect positive correlation. Specifically, let equation (1) stand for the correlation between users c and u, where J is the set of items rated by both users c and u, $v_{l,j}$ is the rating user l gives to item j and $\bar{v}_l$ is the average rating of user l for the items that belong to J, for

$l \in \{c, u\}$.

$$corr(c,u) = \frac{\sum_{j \in J}(v_{c,j} - \bar{v}_c)(v_{u,j} - \bar{v}_u)}{\sqrt{\sum_{j \in J}(v_{c,j} - \bar{v}_c)^2 \sum_{j \in J}(v_{u,j} - \bar{v}_u)^2}} \quad (1)$$

The weighted sum equation below can then be used to predict the rating of user *c* on item *i*. The resulting predictions are sorted and those with highest values are selected for recommendation.

$$P_{c,j} = \overline{v_c} + \frac{\sum_{u \in U} corr(c,u)(v_{u,j} - \overline{v_u})}{\sum_{u \in U} |corr(c,u)|} \quad (2)$$

In equations (1) and (2), *U* is the set of all users who rate item *i*. (In particular, user *c* for whom predictions are being computed, does not belong to *U*). These equations are used in the recommendation process (Section 4.2).

3.2 Content-based Filtering

When Content-based Filtering (CN) techniques are used, items are compared based on their content, which can be described using explicit features. The description can also consist of textual documents with their titles, illustrations, tables of contents, etc. In this paper, items are Web services consisting of one or several operations, which we consider as features. For its part, an operation consists of name, input parameters, output, and description. Thus, comparing two Web services is like finding the similarity between their operations. A recommender system using CN learns a user's interests from the description of the items the user rates. This enables the system to profile users (Pazzani, 1997). As was the case for collaborative filtering, such profiles are long-term user models. These models can be updated as long as users rate items and implicitly change their preferences. In order to estimate similarity between items, various metrics have been proposed (e.g. (Schafer *et al.*, 1999)). In our case, we use ***similar_text (string** first, **string** second),* a well-known PHP function, originally proposed by Oliver (Oliver, 1993), to compute the similarity between Web services (Section 4.2).

3.3 Demographic Filtering

Recommender systems based on Demographic Filtering (DF) aim at categorizing users based on their demographic information and recommend items accordingly. More precisely, demographic information is used to identify the types of users that like similar items. The key element of DF is that it creates categories of users having similar demographic characteristics, and tracks the aggregate usage behavior or preferences of users within these categories. Recommendations for a new user are issued by finding to which category he belongs in order to apply the aggregate usage preferences of previous users in that category. Even though several categorization techniques have been proposed, we shall concentrate on *Data Clustering* to illustrate DF in our Web service recommender system. However our approach is independent of the specific categorization technique used in a DF-based recommender system.

In DF, clustering is used to create the user categories mentioned above by considering the set of all previous users. The objects are users, and each dimension of the space represents one of their relevant demographic characteristics. For a given cluster *C*, its density represents the number of users in it and its radius is a measure of how demographically dissimilar they are. Then, the historical data on usage behavior or preferences of each user in *C* is used to associate with the cluster *C* an aggregate buying behavior. In its simplest form, this aggregate consists of the list of items (Web services in our case) p_1, p_2, ..., p_c that were used/purchased or for which positive feedback was given by users in *C*. When a new user requires a recommendation, the recommender system computes the cluster *C* to which he is closest, and then produces a recommendation of the corresponding list of items.

4 THE WSRS SYSTEM

4.1 Architecture

The architecture of WSRS as depicted in Figure 1 consists of the following components:

Web services providers. The WSRS system connects to these servers, which we call Web service providers, to find Web services required by users.

Web services administrator. Its role consists of organizing the information on each Web service and its related operations, as well as the input parameters needed for the execution of the operations. This information is kept in the Web service database.

Web services database. Web services' information retrieved from Web service providers and managed by the Web service administrator are stored in the WSRS's Web service database. In addition, the information on Web services includes their profiles. Therefore, the Web service database records are almost of the following format: (name, description, associated operations, corresponding execution paths, popularity, average rating).

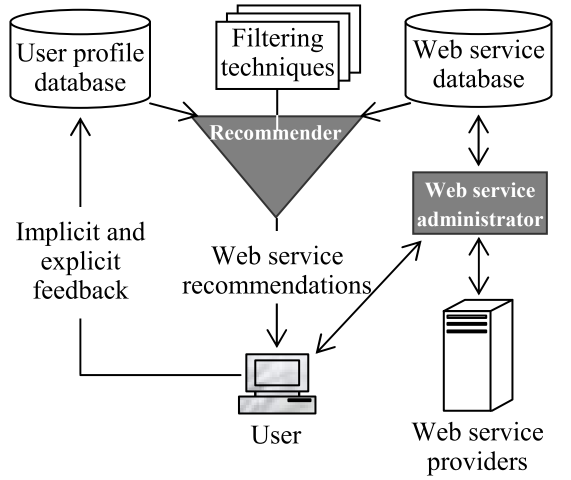

Figure 1: Architecture of the WSRS system.

User profile database. This database stores all users' profiles information like demographic, preferences, usage behavior, fidelity, etc.

Recommender. The recommender is the key-component of the WSRS system. It uses the user profile database and the Web service database to recommend the user suitable Web services, according to his demographic data and previous use of the system, the similarity of his ratings with other users' as well as the similarity between the Web services he used in the past and available ones.

User. The user is any person looking for Web services. He initially submits her username so her profile gets loaded from the user profile database.

4.2 Recommendation Process

The recommendation process involves three components that are described as follows.

The collaborative recommendation (*CF-Rec*) of Web services takes the rating matrix, *R* (this is the contents of Table 1 without the *fidelity* column), and a target user (to whom the recommendations are made), *U*, as inputs. It also uses Equation (1) to compute the similarity between users and Equation (2) to predict ratings for the current user. Recommendations are generated as follow.

CF-Rec(Matrix R, User U){
For each user A from the user profile database{
  If A has rated Web services in common with U{
    Use Eq. 1 to compute similarity between U and A}
  For each Web service j that the user has not yet rated{
    Use Eq. 2 to compute the predicted rating, $P_{U,j}$, of U on j}}
Return Web services j with the highest predictions $P_{U,j}$.}

The content-based recommendation (*CN-Rec*) of Web services uses the similarity computation between Web services. The similarity computation is performed for the operations that compose Web services. More precisely, two operations are similar if their names, inputs, outputs and the keywords that describe them are respectively similar. This requires a lexical analysis over all the parameters of a given operation. Each operation parameter is associated with a *similarity index*, which is a constant value indicating the importance of the parameter in the similarity computation. Let q_{name}, q_{in}, q_{out}, and q_{desc} be respectively the similarity indexes for the name, inputs, outputs, and keywords for a given operation. It is supposed that $q_{name} + q_{in} + q_{out} + q_{desc} = 1$. The similarity computation, *Compare*, between two strings uses *similar_text*() function (Oliver, 1993) and implemented in PHP: Compare(String S_1, String S_2){return similar_text (S_1, S_2);}

The general algorithm to compute the similarity between Web services follows:

Sim (Operation O_1, Operation O_2) {
  name ← Compare (O_1.name, O_2.name);
  in ← Compare (O_1.in, O_2.in);
  out ← Compare (O_1.out, O_2.out);
  desc ← Compare (O_1.desc, O_2.desc);
  Return q_{name}·name + q_{in}·in + q_{out}·out + q_{desc}·desc;}

Now, let *c* be the current operation, v_j the rating prediction for operation *j*, and v_c the rating for the current operation. The algorithm for content-based is as follows:

CN-Rec (Operation c){
  For j in the set of operations{
    v_j ← v_c·Sim(c,j);}
  H ← Operations with highest rating prediction v_j
  Return Web services with associated operations in H}

The demographic recommendation. Using the user's demographic profile, π, WSRS computes the distance between the user's demographic profile and the centroid of each cluster, to find the nearest cluster, C_j, for a certain $j \in \{1, \ldots, k\}$, where *k* is the number of clusters in the demographic clustering table. We use Equation 3 to compute the average fidelity, $f_w(j)$, of a Web service, *w*, that users in C_j have chosen in the past:

$$\overline{f}_w(j) = \frac{\sum_{u \in C_j} n_{u,w} \cdot f_{u,w}}{\sum_{u \in U} n_{u,w}} \qquad (3)$$

where $f_{u,w}$ is the fidelity of user u for the Web service w, and $n_{u,w}$ is the number of times user u used the Web service w. Starting with users in cluster C_j, the WSRS system uses Table 1 to select the fidelity values of these users, and then computes the average fidelity values (Equation 3).

Table 1: The clustering table: each cluster C_j is associated with Web services that present highest average fidelity values with users in C_j.

Clusters	Associated Web services
C_1	ConvertRate, ValidateEmail, getQuote
C_2	ValidateEmail, getQuote
...	…
C_k	IRomanservice

Finally, Web services with highest average fidelity values are selected to be part of the clustering table (Table 1), updated in the same manner, as long as Web services are used.

The demographic recommendation (*DF-Rec*) of Web services takes Table 1 as input to recommend Web services that users in C_j have chosen and used in the past and that have the highest average fidelity values. The DF-Rec process follows:

DF-Rec (Clustering table K, U's demographic profile π){
For each cluster C_j from K{
$d_j \leftarrow$ distance between π and C_j}
C $\leftarrow$ cluster corresponding to minimum distance d_j
Return Web services j with the highest usage percentage that users in C have used in the past}

4.3 Implementation

Currently, we only implemented the CF-Rec and CN-Rec algorithms. However, an example to illustrate the execution of these algorithms is hereafter provided.

Let us illustrate the WSRS system, using the following scenario: Suppose that John, a first year student in the department of History is interested in the conversion between Roman symbols and Integer representations. For example, converting Integer "1200" into Roman symbols results in "MCC". The WSRS system contains *IRomanservice* Web service, which consists of *IntToRoman* and *RomanToInt* operations. However, John does not know that information. Nonetheless, he decides using WSRS.

John needs to log in before using the system and receiving recommendations. This allows WSRS to gather his demographic data, track his activities (Web services he accesses, how many time, the URL links he follows, etc.), and update his fidelity.

After login in, John is automatically presented with a list of Web services recommended by the DF-Rec part of WSRS. This list is likely to include *IRomanservice*, since John is an historian and most of historian may have used the WSRS system before. Moreover, John can also input keywords, such as "convert integer", to describe his need, and be presented another list of Web services to use. Let us suppose that at least one of these two lists include *IRomanservice*, John thus uses this Web service now. More precisely, he clicks on *IRomanservice* link and gets access to *IntToRoman* and *RomanToInt* Operations. He then opts, for instance, to use the former operation. If John tries the operation *IntToRoman* with "1200" as the input parameter, he gets the corresponding roman value, "MCC". Meanwhile, the WSRS system update John usage behavior of *IRomanservice* by incrementing the number of times he used this Web service by "1", and by re-computing his usage percentage (fidelity) versus all the other Web services he used in the past within WSRS (if any). Moreover, John is invited to provide a rating for the operation *InToRoman*, and equivalently for the Web service *IRomanservice*.

Following the rating step, the CF-Rec and CN-Rec recommendation processes now take place. According to John's previous use of the system (he may have used and evaluated other Web services in the past) and his actual feedback (the rating provided to *IntToRoman* operation), he receives Collaborative and Content-based Web service recommendations. Collaborative recommendations result from the CF-Rec process, while Content-based recommendations come from the CN-Rec process. Plus, John is provided with URL links to other Web services related to the *IntToRoma*n operation, which may help him find more information about Web services of his interest. John then may decide to use other Web services from the recommendations, or to follow other URL links.

5 RELATED WORK

Zhang *et al.* (2002) mentioned several problems related to search mechanisms in the WSDL/UDDI systems; the most important one is the lack of accuracy of the search results, which, for its part, is particularly due to the lack of Web services categorization. Therefore, they introduced a new platform, AUSE (Advanced UDDI Search Engine). AUSE uses BE4WS (Business to Explore for Web Services) to facilitate cross research and USML (UDDI Search Markup Language) to support more complex requests. In WSRS, the demographic recommendation process is based on the categorization of users and/or Web services. We are still conducting the implementation of this process, but we hope that to reach a better accuracy than AUSE/USML.

Limthanmaphon and Zhang (2003) used Case-Based Reasoning to search for Web services. Wang *et al.* (2003) proposed the "query by example" process. In this case, partial description of the Web service is provided as an input to the system, which extracts keywords to compare with textual information of other Web services. The system returns Web services having similarity values higher than a certain threshold. The resulting set of Web services is then refined with the *structure-matching techniques* on WSDL documents.

Contrary to the approaches presented above, the WSRS system takes into account the user's profile to tailor Web service recommendations accordingly. WSRS provides user with Web services that better satisfy their needs and requirement because the recommendation process is based on the implicit and explicit feedback gathered from the user during his activities on the WSRS system.

6 CONCLUSION

In this paper, we introduced WSRS, which uses collaborative, content-based, and demographic filtering techniques to provide users with recommended Web services. In a more general context, the WSRS system can be integrated in any UDDI to extend its registry with additional functionalities. This integration is left for future work. Since creating profiles allow the WSRS system to track people and get access to which Web services they are interested in, there is a real need to introduce privacy-preserving mechanism in the Web service recommendation process. Moreover, it could happen that malicious users decide to cheat the WSRS system with false ratings, with many motives behind this kind of behavior, such as *fun and profit* (Lam, 2004). This practice brings out a dangerous aspect for the WSRS system (affecting its reputation for instance). Therefore, we also continue investigating ways to adequately address this issue.

REFERENCES

Breese J., Heckerman D., and Kadie C. Empirical analysis of predictive algorithms for collaborative filtering. *Proc. of 14th Conf. on Uncertainty in Artificial Intelligence (UAI-98)*. Pages 43–52, Madison, WI, 1998.

Burke, R. Hybrid recommender systems: Survey and experiments. *Customer Modeling and Customer-Adapted Interaction* 4(12):331–370, 2002.

Goldberg D., Nichols D., Oki B.M., and Terry D. Using collaborative filtering to weave an information tapestry. *Communications of the ACM* 35(12):61–70, 1992.

Lam S.K., and Riedl J. Shilling recommender systems for fun and profit. *Proc. of the 13th Intern. Conf. on World Wide Web (WWW '04)*. Pages 393–402, New York, NY, 2004.

Limthanmaphon B., and Zhang Y. Web service composition with case-based reasoning. *Proc. of the 4th Australasian Database Conf. on Database Technologies*. Pages 201–208, Adelaide, Australia, 2003.

Mobasher B. and Anand S.S., Eds. *Proc. of the 3rd Intern. Workshop on Intelligent Techniques for Web Personalization (ITWP 2005), 19th Intern. Joint Conf. on Artificial Intelligence (IJCAI 2005)*. Edinburgh, Scotland, 2005.

Oliver J. Decision Graphs – An extension of Decision Trees. *Department of Computer Science, Monash University, Technical Report No. 92/173*. Clayton, Victoria, 1993.

Pazzani M. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13(5-6):393–408, 1999.

Resnick P., Iacovou N., Sushak M., Bergstrom P., and Riedl J. Grouplens: An open architecture for collaborative filtering of netnews. *Proc. of the 1994 Computer Supported Collaborative Work Conf.* Pages 175–186, North Carolina, 1994.

Schafer J.B., Konstan J.A., and Riedl J. Recommender systems in e-commerce. *EC'99: Proc. of 1st ACM Conf. on Electronic Commerce*. Pages 158–166, Denver, CO, 1999.

Tsalgatidou A., and Pilioura T. An overview of standards and related technology in Web Services. *Distributed and Parallel Databases*, 12(3): 135–162, 2002.

Turban E., King D., Viehland D., and Lee J. *Electronic Commerce: A Managerial Perspective*. Prentice Hall, 2006.

Wang Y., and Stroulia E. Semantic Structure Matching for Assessing Web-Service Similarity. *Proc. of the 1rst Intern Conference on Service Oriented Computing*. Trento, Italy, 2003.

Zhang L., Li H., Chang H., and Chao T. XML-based advanced UDDI search mechanism for B2B integration. *Proc. of the 4th IEEE Int'l Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWIS 2002)*. Pages 1530–1534, Newport Beach, CA, 2002.