CHAPTER – 1

INTRODUCTION

1.1 Context

The evolution of communication networks and technologies involves the explosion of services over the Internet. Service providers always compete to rapidly provide the best services to users. This circumstance requires the development of service oriented applications. Web services (WS) look as if as a smart paradigm for publishing and consuming services. The goal of Web service development is to assist service providers to flexibly create new services and dynamically exchanging data with their partners for collaborative business. WS are developed as loosely-coupled applications capable of being run separately to provide a simple function or composed to create new value added services. For instance, simple WS can be services for city codes, local temperatures, quotes, up-to-date news, and composite WS can be flight booking processes that compose functionality offered by other services such as customer authentication, online check-in, car rental, and payment to accomplish a flight booking transaction.

To consume a service, the user sends request and obtains a response from the using service. Basically, services can be consumed in two different ways. They can be used as simple services which provide an interface to receive inputs and return outputs or they can be used as components which can be integrated into business processes. The first type of usage is called as individual use and the second type of usage is called as process use. This research work deals with recommending services with respect to the individual case.

To find a service for an individual use, a user can use a well-known search engine such as Google, Yahoo or Baidu. However, in most cases, the specific service search engines that can not only provide ‘good’ services but also can assist to discover other interesting services are preferred by the user. Also many service portals such as XMethods [1], BindingPoint[2], WebServiceX.NET [3], WebServiceList [4], StrikeIron [5], RemoteMethods [6] and Woogle [7] and service crawlers such as Seekda [8] and Embrace Registry [9] were developed as explicit tools for assisting users in searching and invoking WS for individual use.
In order to support users to consume services for individual use, earlier approaches proposed by various authors take into account data such as Web service descriptions, Quality of Service (QoS) and semantic concepts of services making recommendations not considering data that reveal user concern such as usage data. In addition, they can meet text-based synonym and polysemy problems. Some of them are time-consuming and some others require efforts from users such as rating WS.

1.2 Motivation
Though Web Service technologies and Service Oriented Computing (SOC) promises about loose coupling among components, agility to respond to changes in requirements, clear distributed computing and lesser ongoing investments, WS are not shared and reused as expected [10]. One of the reasons that impede the usage of such technologies and SOC is that efficient WS discovery presents many challenges [11]. Recommender systems (RS) are one tool to help bridge this gap. There are various mechanisms being employed to create RS and the common systems include two main classes such as content-based and collaborative filtering systems. Content-based RS does the matching between textual information of a particular product with the textual information representing the interests of a customer. Collaborative filtering systems make use of patterns in consumer ratings to recommend. Both types of RS require significant data resources in the form of a customer’s ratings and product features; hence they are not able to generate high quality recommendations.

Content-based Web service recommender systems recommend a target user with services that are similar to those previously preferred by the user. Such systems are centered on the analysis of the similarities of the content of the WS. There have been two main approaches in content-based Web service recommendation such as syntactic-based approaches and semantic-based approaches. Syntactic-based approaches have obvious limitations in suggesting high quality recommendations. One direction proposes to enhance service descriptions instead of exploiting them as they are [12]. The Semantic Web offers annotating WS descriptions using unambiguous concept definitions and common ontologies [13]. Supposing that services are accurately described, it is anticipated that finding them will be simplified at the expense of increasing development effort [14]. Unfortunately, semantic-based approaches for service discovery suffer from the typical problems associated with ontologies such as the high
complexity involved in building them [15], the lack of standard ontologies, the absence of public semantically annotated WS, and the effort needed for adopting semantics-aware registries.

WS discoverers often get frustrated with the huge number of search results due to the increased space of published services. There is a lack of search space reduction techniques, which allow discoverers to promptly obtain proper WS, in spite of the huge number of alternatives. Therefore, the scalability of discovery systems should be enriched without ignoring their efficiency of retrieval.

When several WS exist providing the similar service, the QoS factors are considered for selection. Since all the similar services are not invoked by service user, the assessment for such services QoS has to get help from other similar users’ or service users’ invocation records on other WS. Collaborative filtering (CF) forecasts assessment scores of a user for WS by considering other users’ rating on the services. CF is generally decomposed into two techniques such as memory and model-based. Memory-based CF consists of user-based approaches that forecast the assessments of active users based on the assessments of similar users found, while item-based CF methods forecast the assessments of active users based on the computed information of items similar to those chosen by the active user. Pearson Correlation Coefficient is widely used to calculate similarities between users and predicts QoS values based on similar users in memory-based CF [16]. Based on the predicted QoS values, the Web service with the best score or the Top-N WS are selected for the recommendation. In contrast, the forecasting of WS QoS is considerably tougher than the recommending product.

It is important to note that most of the methods for forecasting proposed by various researchers use Pearson-based similarity method. Although this similarity method produces better forecasting result, it takes more computation time and loses performance when the dataset matrices are highly sparse. Compared to similarity-based collaborative filtering, Slope One [17] has been certified to have forecasting method because of its simple nature and high performance.

Furthermore, it is evident that the retrieved result contains more than one matched WS that meet the functional and non-functional (QoS) criteria. Therefore, it is essential to devise an efficient technique to measure the ranking relation order between the retrieved services based on users’ requirements on different QoS attributes. The process of ranking WS is a dominant part of a WS selection system, as it helps users select their desired service easily.
Calculating the similarity degree between the user's request and a service is the fundamental step in the ranking process. There have been different approaches proposed for the problem of ranking WS. These models mostly used different methods to compare all quality parameters of similar WS with the optimal values for each QoS attribute.

According to these approaches, the service with the maximum similarity degree to the optimal values will be returned as the result. There are some open issues that are not well supported by the current approaches. Firstly in some works they either used complicated data indexing methods in their query structure of the ranking process or compared all WS in a pair-wise fashion which will result in a larger computation time. With the growing number of WS with the similar functionality in a registry, the number of pair-wise comparisons will increase, resulting in making the algorithms much slower.

Moreover, most of the ranking approaches ignored the role of consumers in developing their models. To implement the discovery system, they focused only on the services with the optimal values, not the real constraints appeared in the query. In the final result list, users can only have access to those WS with minimum distance to the data items with optimal values. Regardless of user's actual requirements, the result of the algorithms is always the same. It is not rational to ignore consumer’s needs, while the main goal of the model is to respond to user's request. Clients have different anticipations and definitions of an appropriate service, so without considering their demands, there is no guarantee to satisfy them.

Furthermore, most of the existing frameworks concentrated on a small and limited number of QoS attributes. In the majority of works, they considered variables of only numeric type and conducted their experiments on small-sized WS repositories. In reality, there are various types of variables for QoS attributes, which need to be considered to fulfill a desired task. Consumers prefer efficient methods capable of dealing with different types of constraints and large-sized WS repositories.

A CF system uses user feedback, such as ratings, to reflect users’ opinions or experiences on performance or quality, which would be a major factor to be considered in a recommender system. The explicit user feedback systems, such as reputation-based or community feedback-based systems, usually involve human efforts to provide feedback or ratings [18, 19].

However, feedback or ratings after each usage are not provided by all the users [20], and
the provided user ratings might not be accurate [15]. Also explicit user feedbacks are often impractical or unable to completely reflect users’ true opinions or experiences.

To sum up, the cornerstone of service-oriented development methodologies consists of publishing, and discovering services through service registries. This vision requires registries capable of facing an increment on the number of published services. Therefore, it is understood that there is a need of an approach to service discovery that addresses the hitches discussed to make a better recommendation to the service user.

1.3 Research Objectives

This research attempts to overcome the limitations related to service discovery obstructing the growth of service-oriented applications. The focus of this research is that it is feasible to allow discoverers to efficiently discover third-party services, without demanding all the full semantic-based techniques specifications and not charging them with the task of defining highly descriptive and precise queries by providing an approach for efficient ranking of services.

This work proposes methods to service discovery that are lighter than those based on semantics can be a feasible way towards the realization of service-oriented applications. Also attempts to settle the difficulties of forecasting QoS values by combining Pearson similarity and Slope One methods and a simple enhanced algorithm for ranking services considering users’ requirements is better than the existing complicated algorithm. Hence, the main objectives of this research are:

- To propose a new approach to build semantic kernel consisting of semantically similar Web services using clustering and merging method.
- To improve the quality of the predicted QoS values of Web services.
- To enhance a vector-based ranking method by considering users’ requirements.
- To propose a new hybrid-based recommender system using collaborative filtering and content filtering approaches for ranking Web services by means of implicit feedbacks such as the invocation and query histories and also to tackle the so-called cold start problem with respect to collaborative filtering systems.
1.4 Problem Statement

To find a service for individual use, users often spend much time to find, compare and decide the services that are best fitted to their needs. They may easily get confused by the number of Web services returned by search engines or service crawlers.

Naturally, users need support to understand their interests and suggest the users with appropriate services. In this case, RS can be a good solution as they are developed to recommend users the most suitable services to their needs. So, the aim of the proposed research work is to develop a RS-based on hybrid approach using both content-based filtering and collaborative filtering methods to recommend best suitable services to the service user based on his query which has both functional and non-functional requirements using implicit feedback.

The collaborative filtering techniques in the service selection process consist of two steps. The first step is identifying functionally similar services and the second step is QoS-based service ranking and selection.

With respect to the first step which uses the content-based filtering technique, given a user query, the RS have to find the semantically similar services as syntactic or keyword-based search mode suffers from low recall, where results containing synonyms or concepts at a higher (or lower) level of abstraction describing the same service are not returned. Among these returned services, the services that have the most similar features as the profile based on certain criteria will be recommended. Some of the problems with the content-based system are that it cannot distinguish the quality of the items if they possess the same or very similar features and it is difficult to recommend an item that is not similar to any item the user has ever selected.

In the presence of multiple WS with identical or similar functionalities, Quality of Service (QoS) provides non-functional Web service characteristics for the optimal Web service selection. Since the service user has not even invoked the service in the past, the estimation for such service’s QoS has to get help from other similar users or self’s invocation records on other Web services. The QoS performances of a Web service are to be predicted using CF methods for an active user by employing historical QoS information from other similar service users, who have similar historical QoS experience on the same set of commonly-invoked Web services. Then using the predicted values, the WS are to be ranked and the best Top-K services satisfying the user’s requirements are returned as recommended services to the user.
In a nutshell, the proposed research work aims at recommending services that are close to user interest using implicit knowledge hidden in usage data. To address this research problem, the questions how to select semantically similar services, how to predict and improve the prediction quality of QoS, how to rank similar services considering user’s requirements and how to overcome the cold-start problem are to be answered.

1.5 Research Contributions

The contributions of this research work are fourfold:

- An innovative approach is proposed to find semantically similar WS using the information from the description in WSDL document for a service user request by means of the support-based semantic kernel built from a low ordered matrix. Most of the algorithms which involve singular-vector-decomposition based methods for kernel creation involve computationally expensive matrix factorization introducing scalability bounds. This work addresses the issue of scalability by proposing a new idea of dimensionality reduction by clustering and merging method.

- In order to resolve the difficulties in forecasting the QoS values of WS, a new CF-based RS approach by combining Pearson similarity and a Slope One method is proposed. In the proposed method, the Pearson similarity is adopted among any two services as their deviation weight. Also, statistical process control-based smoothing and weight adjustment approaches are also used to minimize forecasting error.

- An enhanced vector-based ranking method is proposed by considering user's requirements. The vector-based model is chosen because of its simplicity and high efficiency.

- Finally, where a new hybrid approach using content-based and CF methods is proposed and implemented for recommending Web services to the user where the invocation and query histories are utilized to deduce the users’ preferences and to calculate the user similarities. To overcome the cold start problem with the CF systems, the final ranking score is calculated using QoS-based matching and CF-based scores.
1.6 Thesis Organization

This thesis contains a survey of the different Web service discovery approaches, a description of our work, and an overview of the results and contributions. An outline of the structure of this thesis is shown as follows:

Chapter 2 includes details about the background of Web service technology along with their related standards. Also it includes the basics of recommender system and of its kind with limitations.

Chapter 3 describes and analyses the most relevant related work. It includes the various approaches for Web service selection, different approaches to maximize the prediction accuracy of QoS values of WS, various rank aggregations/ranking methods and the usage of log content for providing recommendations.

Chapter 4 introduces the proposed Web service discovery method created on semantic analysis to find semantically similar WS using a new space reduction technique and discusses how the dimensionality reduction of the term-document matrix by clustering and merging of the training documents creates the semantic kernel which is one of the innovations of this research.

Chapter 5 attempts to settle the difficulty as to how to forecast the missing QoS value of Web service by using the standing information by combining Pearson similarity and Slope One methods for QoS forecast. It discusses the Pearson similarity which is used between two services as their weight deviation and how the statistical process control is employed for reducing prediction error.

Chapter 6 presents the baseline of three current well approved ranking algorithms such as Borda Fuse, original Distance-based and Skyline operation methods and how they are applied to ranking web services, and then includes the common issues and problems that arise in ranking WS in a selection system. Then the chapter includes the proposed optimized distance-based algorithm with consideration of users’ requirements. It also presents an enhanced Positional-based model by considering users’ requirements.

Chapter 7 explains the proposed architecture for Web service recommendations using a hybrid approach, the history data used for selection and ranking, the algorithms used in finding similar users, as well as the ranking algorithm in details. It also includes details about the
assessment of the efficiency of our algorithms using the time complexity analysis. Finally, it includes qualities of Top-K returned results.

Chapter 8 concludes the dissertation with summary of the contributions, some perspectives for future research, and those publications that this research originated.