Result Refinement in Web Services Retrieval Based on Multiple Instances Learning

Yanzhen Zou, Peking University, China
Lu Zhang, Peking University, China
Yan Li, Peking University, China
Bing Xie, Peking University, China
Hong Mei, Peking University, China

ABSTRACT

Web services retrieval is a critical step for reusing existing services in the SOA paradigm. In the UDDI registry, traditional category-based approaches have been used to locate candidate services. However, these approaches usually achieve relatively low precision because some candidate Web Services in the result set cannot provide actually suitable operations for users. In this article, we present a new approach to improve this kind of category-based Web Services retrieval process that can refine the coarse matching results step by step. The refinement is based on the idea that operation specification is very important to service reuse. Therefore, a Web Service is investigated via multiple instances view in our approach, which indicates that a service is labeled as positive if and only if at least one operation provided by this service is usable to the user. Otherwise, it is labeled as negative. Experimental results demonstrate that our approach can increase the retrieval precision to a certain extent after one or two rounds of refinement.

Keywords: data management; data retrieval; multiple instances learning; service registry; software selection; query processing

INTRODUCTION

Faced with limited time-to-market and volatile requirements, software development currently relies on reusing of existing software components more and more. As the foundation of Services-Oriented Architecture (SOA), Web Services technologies aim at facilitating and improving the quality of component-based applications on the Web (Wang & Stroulia, 2003). Web Services has gained a lot of support in both the academic and industrial fields. Many corresponding tools and specifications have been developed. A Web Service is described...
through the Web-Services Definition Language (WSDL), and it can be advertised, discovered, and integrated via the Universal Description, Discovery, and Integration (UDDI), or be invoked at run time by the Simple Object Access Protocol (SOAP). As a result, application developers usually wrap existing software components as Web Services and build a new system by leveraging existing Web Services.

To facilitate discovering and reusing existing Web Services, UDDI provides directory services for Web Services offered by businesses (Colgrave, Akkiraju & Goodwin, 2004). In UDDI registry, Web Services are organized according to some categories of business activities. Service providers could advertise their services by classifying a service to the appropriate UDDI directory category. Thus, to discover the potentially relevant services, service requestors (e.g., software developers) can submit a query leveraging some category information to UDDI registry for discovering an appropriate service. Here we treat this kind of approach as category-based retrieval (Prieto-Diaz & Freeman, 1987; Ostertag, Hendler, Prieto-Diaz & Braun, 1992; Mili, Mili & Mittermeier, 1997), because they are actually performed by matching the user’s requirement against each Web Service’s category information.

Currently, UDDI is limited in its retrieval mechanism by its inability to extend the simple keyword-based matches and category-based matches. It is pointed out that the precision of the retrieval mechanism in UDDI registry is not satisfying (Wang et al., 2003; Li, Pan, Zhang, Xie & Sun, 2006) In a typical retrieval process, a service registry will acquire a query generated by the service requestor and match the query against each Web Service’s category information. Thus, they return all functionally relevant Web Services in a category as candidates, which can achieve relatively high recall. However, the precision of these approaches is quite low, partially because some Web Services with relevant semantic descriptions in the candidate set may not provide actually suitable operations for users. Subsequently, the service requestor has to browse and comprehend those candidate services in order to identify the appropriate ones. Thus, the process of identifying appropriate services will become much more tedious and time-consuming when the candidate services set is quite large.

In this article, we present a novel approach to refine the coarse-matching results in the category-based Web Services retrieval mechanism. The basic idea of our approach is to exploit category matching as the first step whose target is to gain coarse candidate services with the required functionality description. Secondly, the retrieval results are refined based on a multiple instances learning technique (Dietterich & Lathrop, 1997; Zhou, Jiang & Li, 2005). In this refinement step, a retrieval mechanism collects some browsed services in the retrieval process as training examples to refine the coarse retrieval results. The browsed services are fed back as user interested (positive) or user uninterested (negative) and stored in a log for learning the interface requirements of the user-required service in order to achieve better retrieval performances. The refinement iterates until the matching result is refined enough for the service requestor to select. Based on our approach, we have built a prototype service retrieval system, which is composed of three parts: a category matching based retriever, a log collector, and a refinement tool based on multiple instances learning. Experimental results demonstrate that our approach can increase the retrieval precision to a certain extent after one or two rounds of refinement.

The rest of this article is organized as follows: Section 2 presents the motivation for refining coarse services set. Section 3 explains the technology background of multiple instances learning (MIL). Section 4 describes an improved Web Services retrieval process integrated into our refinement mechanism. Section 5 proposes our refinement tool for Web Services retrieval using MIL. Section 6 shows a case study, and Section 7 presents our experiments for evaluating our refinement. Section 8 discusses related works. Finally, we conclude this article and present our future work in Section 9.