Web Services Enabled Text Categorization System: Service Infrastructure Designing

Xiaobin Zhang, Jian Mei, Suge Wang and Wu Zhang

Department of Computer Science and Technology, Shanghai University,
Shanghai, 200072, China

Summary

Web services over the Internet have become a broadly used concept in a variety of applications for business, science, engineering and entertainment. To this end, this paper also proposes to create a new web service application, PSE-TC — a Problem Solving Environment (PSE) for text categorization that integrates the traditional text categorization system and the novel Grid technologies based on PSE infrastructure. In our previous paper (E-Science 2006), we have proposed the framework of PSE-TC which is able to support the activities that concern the building of the text classifier service, the classifying of the texts, the defining of the workflow, the selecting of service's nodes and the reflection of the execution status through the web portal. In this paper, more web service modules are introduced in detail. Meanwhile, we have launched a novel middleware named Agent for (1) Executing those defined workflows by the users and locating and invoking the services on the Grid nodes according to those workflows, (2) saving the data produced by every service of the workflows, and (3) being a module as the liaison system uniforms the output data from preceding modules and/or external modules as the request of the next modules and transfer the data to the input data for the next module. By embedding this Agent service into our framework, a web service can be plugged into easily, and therefore the services modules connectivity and compatibility are improved to a great extent. In the end, some convincing experiment data are obtained, and prove that PSE-TC can provide a seamless and powerful text categorization research platform.

1. Introduction

Services such as automatic purchasing, automatic updating of prices, or getting latest information etc, can be provided on the Internet using Web services technology. A client can access these services using the Internet. Web services infrastructure includes some standards, such as simple object access protocol (SOAP), Web services description language (WSDL) and universal description, discovery and integration (UDDI). The problem solving environment (PSE) is a system that provides all the computational facilities necessary to solve a target class of problems. It uses the language of the target class and users need not have specialized knowledge of the underlying hardware or software [1]. Examples of such systems include AVS [2], Cactus [3], etc. Owing to the rapid advancement of the web services technologies, employing these technologies into building the PSE framework has become more and more significant. And as a result, we can add, delete or modify the PSE's components dynamically without changing the PSE framework.

In this paper we propose a novel Grid PSE framework for text categorization. Within this PSE platform, we can construct the text classifier and estimate the category of the new text. Our key contributions of this project include the following aspects: (1) constructing the PSE framework with the web service technologies, partition the process of text categorization and design the text categorization service; (2)Showing and explaining the concrete workflow of this PSE-TC; (3) Launching a novel middleware named Agent for executing those defined workflows by the users and locating and invoking the services on the Grid nodes according to those workflows, saving the data produced by every service of the workflows, and being a module as the liaison system uniforms the output data from preceding modules and/or external modules as the request of the next modules and transfer the data to the input data for the next module.

The rest of the paper is organized as follows. In Section 2, we simply describe the PSE-TC architecture. We illustrate some workflow of this PSE-TC and the Agent designing mechanism in detail in Section 3. In Section 4, we show an instance of this application

system and Section 5 concludes the paper and discusses the future work.

2. PSE-TC Architecture

This PSE for text categorization contains mainly four components: Web Portal Service, Construct Text Classifier Service, Classify Text Proceeding Service and Workflow Management Service (see Fig.1).

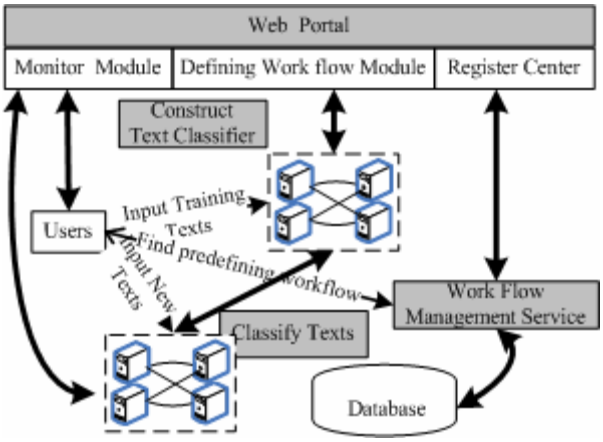

Fig. 1. The framework of the Grid enabled PSE for text categorization.

In our Grid enabled PSE, the text categorization process is focus on the two components, "Construct Text Classifier" and "Classify Text". A whole process from the beginning of inputting new texts to the end of getting the classify results is divided into several independent modules, which are assigned to specific services offered by Grid nodes.

In this PSE-TC, we partition the whole process of text categorization into two main stages. One is to construct the text classifier, and the other is to classify the texts. Firstly, the users log on the Web Portal, define a workflow and input the training texts into "Construct Text Classifier". After the text classifier is constructed, the user input new texts into "Classify Text". Finally, the "Classify Text" proceeds to classify the texts based on the text classifier built on the first stage and get the final classify results.

The Workflow Management Service is employed to record the information of the workflows, such as the service names, the related Grid nodes and the execution efficiency, etc. When the users define the workflow, they consider the Workflow Management Service as a reference.

3. Agent and Workflow Designing

The web services technology uses SOAP (Simple Object Access Protocol) [4-5] for the XML payload and uses a transport such as HTTP to carry the SOAP messages back and forth. SOAP messages are XML documents that are sent between a web service and the calling application. We use Apache AXIS [6] and Tomcat web service server [7] to implement PSE-TC. Apache AXIS is an implementation of the SOAP submission to W3C.

AXIS services are of different types, including the RPC (Remote Procedure Calls) –type and the Message-type. At present, there are existing default RPC services, Message services with byte stream serialization and RPC with SOAP attachment services.

Based On our previous research work [8], it is observed that message services with byte stream serialization and RPC with SOAP attachment services have the advantages of smaller message size and better timing performance over the default RPC approach in web service applications, which is particularly important as the mesh size getting bigger and bigger. In addition, RPC with SOAP attachment services are widely supported by many popular Web Services providers. On the other hand, unlike for a default RPC service, the corresponding WSDL document generated for a message service with byte stream serialization and a RPC with SOAP attachment service does not provide enough information for clients to develop application programs.

To overcome this disadvantage, we introduce a middleware named Agent. Besides this, the agent can execute those defined workflows by the users, invoke the services on the Grid nodes according to those workflows and be a module as the liaison system uniforms the output data from preceding modules and/or external modules.

Once the user defines a workflow, the portal will retransmit the workflow to the agent. The agent will produce an instance for managing this workflow. In the PSE, each step is visual for the user. So the agent is also response for recording the results of each steps or storing the results to the appropriate node. For the agent is communicated with the Web Portal, all these message can be obtained through the web portal for the user. Through this agent, the user can monitor the specific workflow with ease and have unnecessarily to understand the mechanism of monitoring. Another important function of this agent is to find some running error timely. For instance, a user defines a workflow which contains a kind of Feature Select Method Service, and this workflow is retransmitted to the agent. When the workflow reaches the Feature Select Method Service, however, the host providing this server is broken. In this case, if the system has not the monitoring module, the only way to finish this task is only to wait for the host's restarting. In a general

way this task ended in failure. But if the system has the monitoring module, the user can find the executing status, find the failure of the host and select another service node to finish this task. At our present PSE-TC system, we only find the running status of the workflow with this agent. Our next work will focus on restore mechanism and efficiency, such as finding the substitutable service, reflecting the useful status message with a smaller cost, etc.

The third primary effect is serving as a liaison system, which uniforms the output data from preceding modules and/or external modules as the request of the next modules and transfer the data to the input data for the next module. The communication between the agent and the web service is show as Fig.2.

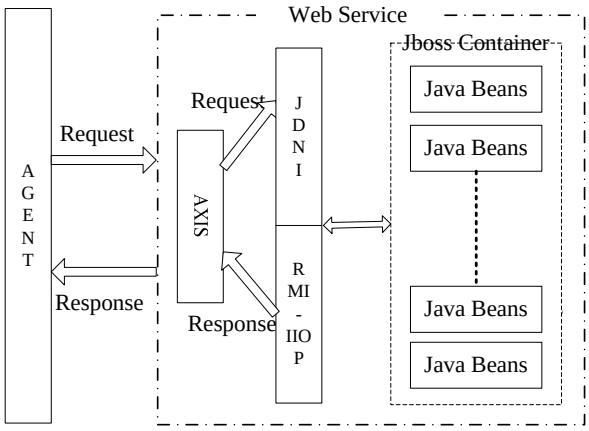

Fig. 2. The communication between the Agent and the concrete web service

We introduce the Jboss as the Java beans container, extend the primary Java Naming and Directory Interface (JNDI), a standard extension to the Java platform providing Java technology-enabled applications with a unified interface to multiple naming and directory services and Java Remote Method Invocation over Internet Inter-ORB Protocol technology ("RMI-IIOP") function as the service interface, and adopt the AXIS as the service portal.

The agent is used to communicate with the different concrete web service. To establish the communication among the distributed modules, output data from preceding modules must be fitted to input data for the next module. Therefore each PSE module may require a data converter to fit the input/output data for each module. For example, the D-NCAS system (9) proposes and constructs a liaison system to connect modules. The computer-assisted module liaison system generates an adapter module among the distributed PSE modules. As a result, the module liaison system extends the potential capability of PSEs extraordinarily.

For in our PSE-TC, all input/output data are described by XML. We develop this agent to construct a XML tree to express the input/output data. From this tree, we can see the output data structure clearly. And we can compare this structure with the required data structure of the next module. Through making some modifications, we can use the output data as the next module's input data smoothly.

As we have mentioned that we partition the whole process of text categorization into two main stages, constructing the text classifier and classifying the texts. At the text training service stage, the users input some training texts and define training parameters, select the training method, choose the service provider. Preprocessing Service, Remove Stop Word Service, Character to Feature Service, Feature to Feature Vector Service, Feature Select Method Service, Construct Classifier Model Service are included in this stage. And the other stage is Text Classifying. At this stage the users input the testing text, and select the classifier model, execute the classifying process, and make an evaluation on the chosen classifier according to the standard of the text categorization. Recall, Precision and F1.

In the Fig.3, the whole workflow of the service publishing and invoking is depicted.

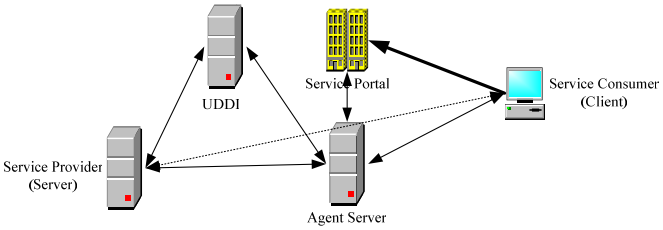

Fig. 3. The communication between the Agent and the concrete web service

The Service Provider is to transfer the handling result between the Service Consumer and the Service Provider, monitor the existing workflow and reflect the UDDI.

1. When you handle a task, it must be existed the three services (The Service Provider, the Agent service and the Portal service).
2. The Service Providers register their service to the UDDI, and send the available resource status to the UDDI in order to reflect the real status periodically.
3. A client initiates a request to the Service Portal, and finds the wanted service through the UDDI module of the Portal.
4. According to the load of the current server, the

UDDI select the server based on some selected strategy.

5. The Agent assigns the task launched from the client to an Agent instance.
6. The client has a session with the Service Portal, define the workflow, order the service and define the input parameters.
7. The Agent formats and transfers the input parameters to the Service Provider. The Service Provider handle this process, and send the results to the agent.(HTTP, SOAP)
8. The Agent transmits back the result to the client using the SOAP.

4. A simple Implementation on this PSE

We test our Grid enabled PSE on a repository with 2930 Chinese texts. We select the Information Gain method as the "Feature Extraction Service" and KNN method as the "Training Service". In the Fig.4, we describe the process of constructing text classifier and classifying text category.

In this text repository, all the texts are classified into 38 categories by the experts. In our experiment, we only make use of six categories' texts which are about 500 texts. The key checkpoint of this experiment is to verify the consistencies between the results are drawn through our PSE and the ones are drawn from the existing text categorization system (TxtCat [10]). Here, we only list the results generated by our PSE as Table.1:

We do the same experiment based on the same training set and testing set on the TxtCat system. And the conclusion is the same as the above table. Through this experiment, we can verify the usability of this Grid enabled PSE framework, ensure the validity of the workflow of this PSE application and propose the future work.

Table 1 : The values of three index of text categorization

Category	Precision	Recall	F1
Literature and Art	84.34%	84.56%	84.44%
Commerce and Economy	85.83%	86.67%	86.24%
Entertainment	83.45%	84.34%	83.89%
Government and Politics	87.08%	87.08%	87.08%
Society and Culture	82.39%	83.78%	83.08%
Education	89.11%	91.42%	90.25%

5. Conclusion and future work

This Grid enabled PSE facilitates the usability of the distributed computing resources, the various Feature Extraction method service and the diverse Training method service on the Grid. According to the characteristic of the text categorization, we divide the text categorization procedure into two stages: Construct Text Classifier and Classify Text. Also, this PSE framework integrates Test Representation services, Feature Extraction services, Training services, Calculation Weigh services, Execution Status service, Evaluation service and Feedback Control service. In this paper, we have described the PSE framework shown and explained the concrete workflow of this PSE-TC, launched a novel middleware named Agent for executing those defined workflows by the users and locating and invoking the services on the Grid nodes according to those workflows, saving the data produced by every service of the workflows, and being a module as the liaison system uniforms the output data from preceding modules and/or external modules as the request of the next modules and transfer the data to the input data for the next module.

Additional future work of this project will be to integrate the existed and novel classifying method and training method. Further, since on defining the workflow the optimal path selection is not considered in our PSE, we plan to add the path selection algorithm into consideration in the future research.

References

[1] I.Foster and C.Kesselman (Eds): The Grid: Blueprint for a New Computing Infrastructure. http://www.mkp.com/books_catalog/1-55860-475-8.asp, Morgan Kanfmann, Los Altos.
[2] Avs/advanced visual systems. http://www.avs.com.
[3] The cactus code server. http://www.cactuscode.org.
[4] Ian Foster, Carl Kesselman, Jeffery M.Nick, Steven,Tuecke. (2002) Grid services for distributed system integration. IEEE computer,35 (6):37-46
[5] SOAP and Web Services http://www-136.ibm.com/developerworks/webservices
[6] Martin Gudgin, Marc Hadley, Jean-Jacques Moreau, Henrik Frystyk Nielson. (2001) SOAP Version 1.2,W3C Working Draft 9. http://www.w3c.org/TR/SOAP12/.
[7] The Apache AXIS project http://ws.apache.org/axis/
[8] Xiaobin Zhang ,Mo Mu,Guoyong Mao and Wu Zhang, Performance of Grid-Based PDE.Mart, E-Science 2006
[9] Shigeo KAWATA, Masumi INABA, Hideaki FUJU, Hideaki SUGIURA, Yuichi SAITOH and Takashi KIKUCHI, Computer-Assisted Liaison among Modules in a Distributed Problem Solving Environment (PSE) for Partial Differential Equation Based Problems Transaction of JSCES, Paper No.20050029
[10] Mario Cannatara, Camela Comito, Antonio Conguista. (2003) Grid-based PSE Toolkits

for Multidisciplinary Applications. FIRB "Grid.it" WP8 Working Paper

[11]Ian Foster, Carl Kesselman, Jeffery M.Nick, Steven,Tuecke. (2002) Grid services for distributed system integration. IEEE computer,35 (6):37-46

[12] SOAP and Web Services http://www-136.ibm.com/developerworks/webservices

[13] Martin Gudgin, Marc Hadley, Jean-Jacques Moreau, Henrik Frystyk Nielson. (2001) SOAP Version 1.2,W3C Working Draft 9. http://www.w3c.org/TR/SOAP12/.

[14] The Apache AXIS project http://ws.apache.org/axis/, http://jakarta.apache.org/

[15]Erik Christensen, Francisco Curbera, Greg Meredith, Sanjiva Weerawarana. (2001) Web Services Description Language (WSDL) 1.1. W3C Note 15. http://www.w3.org/TR/wsdl

[16] Java Serialization API， http://java.sun.com/developer/technicalArticles Programming/serialization/

[17] Base64 API， http://jakarta.apache.org/commons/codec/apidocs/org/apache/commons/codec/binary/Base64.html

[18] MIME Specification， http://www.mhonarc.org/~ehood/MIME/

[19] Richard Monson-Haefel, J2EE Web Services, Addison-Wesley Professional, 2004