Utilizing Web2.0 in Web Service Ranking

Ioan Toma
University of Innsbruck,
Technikerstrasse 21a, A-6020,
Innsbruck, Austria
ioan.toma@sti2.at

Ying Ding
Indiana University
Bloomington, IN 47450, USA
dingying@indiana.edu

Krissada Chalermsook
Asian Institute of Technology,
P.O. Box 4, Klong Luang,
Pathum Thani, 12120 Thailand
krissada.chalermsook@ait.ac.th

Elena Simperl, Dieter Fensel
University of Innsbruck,
Technikerstrasse 21a, A-6020,
Innsbruck, Austria
firstname.lastname@sti2.at

Abstract

The current shift in realizing distributed applications by applying service-oriented principles translates into more and more web services being published on the Web each day. This introduces a set of new challenges such as how to organize, search, rank and select services. For ranking especially the use of any external information to the service, such as social information coming from Web2.0 sources has the potential to improve the accuracy of this task. In this paper we explore the idea of using social annotations technology for ranking web services. We show how such approach can be implemented using information provided by one of the largest social networks - del.icio.us.

1 Introduction

Two recent trends are shaping the current Internet bringing on one hand a *service-oriented perspective*, allowing existing functionalities to be published, discovered and invoked over the net in a standardized manner as services, and on the other hand a *social-oriented perspective*, allowing users to contribute and share information within communities.

The *service-oriented perspective* promoted by Service Oriented Architectures (SOA) pushes the notion of service as the central notion, abstracting from the underlying implementation and hardware. This abstraction provides only the common philosophy on how to design distributed application but the paradigm shift introduces a set of new challenges such as how to organize, search, rank and select services. Among them, ranking web services is a core challenge that any SOA-based system needs to address. Existing solutions for service ranking are tightly integrated with service discovery and selection solutions. They use service data such as Non-functional properties or Quality of Service (QoS) to compute rank values for services. In most cases multiple Non-functional properties (e.g. price, availability, etc.) as well as dynamic values are considered ([13], [9], [14], [8]). Using this kind of data makes the above mentioned approaches to be categorized as content-based ranking solutions. An analogy with the Web reveals that except content-based ranking, solutions that exploit 'visibility' of the objects to be ranked as well as user annotations and feedback have a clear potential.

With the new Internet wave created by the Web 2.0, the current Web is changing from static publication platform to interactive information sharing platform, bringing a social-oriented perspective in the current Internet. Web2.0 pioneers the way for online publishing which makes it so transparent and easy that everyone who can type can join online information communication and sharing. Normal users can easily add their contents and opinions to the Web. These information accumulated in large size social network starts to provide certain value for interesting applications.

As mentioned above ranking web services has been a hot topic and various approaches have been tried in order to improve ranking algorithms and ranking results. But the social information coming from Web2.0 is not significantly considered in this area. This paper aims to provide a social ranking approach based on one of the largest social networks - del.icio.us - for web service selection and ranking. del.icio.us is one of the most well-known social networks for sharing bookmarks. It was founded by Joshua Schachter in 2003 and had immediately reached one million users in 2006 [1]. In 2005, it was bought by Yahoo. del.icio.us is one of the most successful social networks attracting millions of users because it provides convenient services for tagging and sharing bookmarks. Each user can have his own bookmark store and access it anywhere around the web as long as the Internet is provided. Furthermore he can share his bookmark store within the del.icio.us community which can generate many interesting and useful community services, such as what are the highly tagged bookmarks, who are most active users, what are the tag clouds in the community, etc. del.icio.us also publishes its functionality as a web service by providing common API which allow easy access and query of data (mainly tags and bookmarks) remotely.

The rest of the paper is organized as follows. Section 2 contains the system description, including a high level overview of the system, the system implementation, architecture and components description. Section 3 presents the experimental results and Section 4 summarizes the related work. Finally Section 5 discusses several issues with respect to implementation and extension of the approach and Section 6 concludes the paper and presents our future work.

2 System Description

2.1 Overview

Social web service ranking (SWSR) features itself by utilizing Web2.0 into web service ranking. In this paper, del.icio.us has been selected as the social network for ranking web services. But similar approach can be easily extended to other social networks.

The system is quite simple and straightforward. Given a set of web services the system checks if there are web pages in the web services domains that are bookmarked in del.icio.us. If this is the case, the services are ranked based on the numbers of people in del.icio.us that have tagged the associated web pages. For example, one user would like to know which of the shipping services have a high social visibility. He should identify some keywords and run his query against the service repository and gets a set of .wsdl shipping services. For each web service the system checks if there are web pages from the same domain already bookmarked in del.icio.us. These web pages are retrieved from the Web and stored locally. The resulting set of web services is finally ranked based on how many users have tagged the corresponding bookmarks.

2.2 Architecture

The system we propose consists of a set of loosely coupled components: an *annotations search engine* that retrieves relevant del.icio.us bookmarks given a user query, a web service finder responsible to crawl the domain of the bookmarks and identify relevant .wsdl files if any and finally a *web service ranker* to perform web service ranking. The system is implemented in Java and runs as a Web application hosted in the Apache Tomcat container. The rest of this section details each component of the system and described how these components work together.

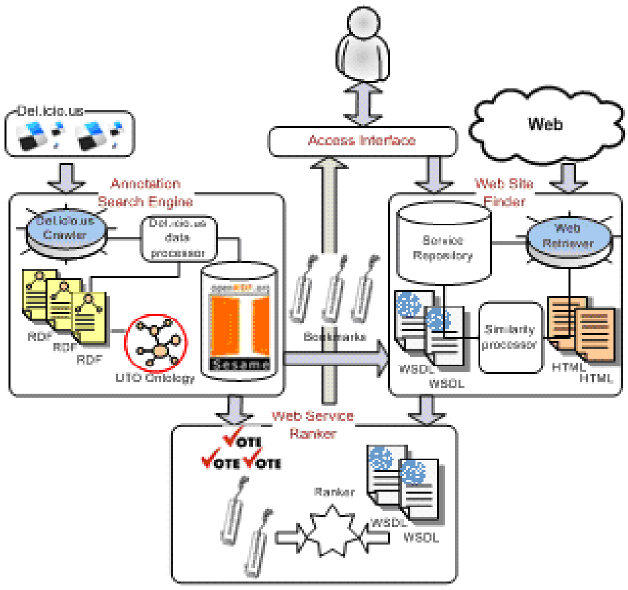

Figure 1. System Architecture

2.2.1 Access Interface

The *Access Interface* component is the visual gateway to our system. It allows users to formulate queries as set of keywords and to submit them to the system. Once a request has been processed by the system, the resulting set of ranked services is returned to the *Access Interface* which displays them to the user. The Access Interface is implemented using JSPs technology.

2.2.2 Annotation Search Engine

The functionality of the annotation search engine can be shortly summarized as follows: given a user keyword query the *Annotation Search Engine* returns a set del.icio.us bookmarks relevant for that query. As illustrated in Figure 2 the Annotation Search Engine includes a set of subcomponents, namely *del.icio.us Crawler*, *del.icio.us Data Processor* and *del.icio.us Semantic Repository*.

del.icio.us Crawler subcomponent is responsible for crawling social information from del.icio.us social network. We chose an internally developed multi-crawler designed for crawling major social networks including del.icio.us, flickr and youtube [6]. This crawler is based on the "Smart and Simple Webcrawler". Since flickr and youtube are the platforms for sharing photos and videos and not web services, they are not considered in our research. Figure 2 shows the detailed class diagrams of the crawler. Another option we were initially considering to retrieve the relevant bookmarks given a query was the del.icio.us API. This seams a nicer approach but we face serious limitations. More precisely due to authentication issue, user can only access his own data (such as tags, bookmarks, updates and bundles) thus one can not have access to all bookmarks relevant to a query.

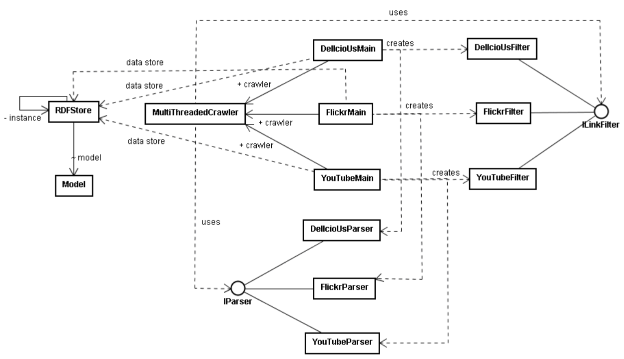

Figure 2. Class diagram overview of the crawler

del.icio.us Data Processor subcomponent is responsible for processing the .html pages crawled from del.icio.us, parsing them and generating structured, light weight semantic annotated data represented in RDF [11]. RDF is a standard language and method for making statements about Web resources in the form of subject-predicate-object. Since RDF is schema independent, we use the Upper Tag Ontology [5] as a model to structure the RDF social data. The Upper Tag Ontology, shortly UTO, is a light ontology which includes concepts such as tag, tagger, object, vote and relations such as hasTag, hasVote, hasObject. These ontological elements are used to build the RDF representation of Del.icio.us crawled data.

del.icio.us Semantic Repository subcomponent stores the RDF triple generated by the Data Processor according to the UTO Ontology. We chose Sesame [4] as a semantic repository to store del.icio.us data. Sesame is an efficient storage and for RDF [11] and RDF Scheama [3] which can be deployed on top of RDBMS storages. Please notice that a RDBMS repository could have been use as well. The choice for a semantic repository was motivated by flexible data structures that can be stored without the necessity to extend or change the data model.

2.2.3 Web Site Finder

The Web Site Finder component is a top level component responsible for mining the correspondences between web services and possible web pages bookmarked on del.icio.us in the domain of web services. Given a web service description this component checks if web pages having the same domain have been bookmarked with del.icio.us. If such pages exist they are retrieved from the Web and their similarity with the web service description is computed. A correspondence matrix between web services and most similar web pages is finally built.

As illustrated in Figure 1 the Web Site Finder includes a set of subcomponents, namely *Web Retriever*, *Service Repository* and *Similarity Processor*.

Web Retriver subcomponent is responsible for retrieving .html files from the domain of the web service. More precisely the component makes first a query to del.icio.us *Semantic Repository* and receives the URLs of the web pages from the same domain as the web service. It retrieves afterwards from the Web the content of these web pages and stores them locally.

Service Repository subcomponent is used to store .wsdl files identified by the WSDL Processor subcomponent together with the set of keywords in the query. It is simply a persistent layer implemented using a classical RDBMS. Please note that the *Access Interface* component interacts with the *Service Repository* component allowing user to search for services based on the keywords they provide as input.

Similarity Processor subcomponent determines the similarity between a web service wsdl description and each page in its domain which was bookmarked in del.icio.us. To determine the similarity between a web service and a web site we use two techniques. First we check the co-occurrence of same keywords in the set of keywords used to name web service operations and keywords in the web page. Second we check the co-occurrence of same keywords in the set of keywords used to name web service operations and annotation keywords associated with the page available in del.icio.us.

We regard services, web pages and set of tags used to annotate a web pages as being sets of keywords. To build the vector model for services we consider only the names of the operations exposed by the services. For web pages the initial preprocessing step consists in removing all html tags. For all keywords sets representing services, web pages and tags a further preprocessing is performed using stopwords and stemming. We use the cosine similarity metric [10] to measure the similarity between two sets of keywords representing either services, web pages or tags.

2.2.4 Web Service Ranker

The Web Service Ranker component is another core component of the system being responsible with the actual ranking of web services. This component performs first a ranking of del.icio.us bookmarks which translates in the end in ranking of associated web services. The straight forward approach we adopt is to rank each bookmark based on how many people have bookmarked the web page, in other words how many votes a bookmark has. This information is already available in del.icio.us Semantic Repository being crawled from the del.icio.us web site. The ordered list of bookmarks based on the number of votes is used to generate the ordered list of web services which are in the end returned to user.

2.3 Algorithm

The current implementation of the system follows a simple and straightforward approach with respect to how the rank of services is determined. Basically the number of votes received by the associated bookmark in delicious becomes the rank value of the service. In the rest of this section a general algorithm to compute the rank of a web service is proposed. Our algorithm is based on the algorithm proposed in [2] that was extended to address the web service dimension. The SPR algorithm evaluates the popularity of a web page based on the mutual enhancement among three distinct sets of objects: (a) popular web pages, (b) web users and (c) hot social annotations. We introduce a fourth dimension/set of objects: (d) web services. We call the new algorithm *SocWSRank*. The algorithm is provided in the following listing.

If P is the set of web pages having N_P elements, U is the set of users having N_U elements, A is the set of annotations having N_A elements and S is the set of web service having N_S elements, one can define association matrixes between annotation and pages M_{AP}, pages and users M_{PU}, users and annotations M_{UA} and finally between web pages and web services M_{PS}. The matrix elements are assigned with values capturing the associations between each pair of dimensions. For example the $M_{AP}(p_i, u_j)$ element is assigned with the number of annotations user u_j uses to annotate page p_i. Elements of matrixes M_{AP} and M_{UA} are computed similarly as described [2]. Elements of matrix M_{PS} are assign value 1 if there is a correspondence between the web page and the service, value 0 otherwise.

Algorithm 1 Social Web Service Rank (SocWSRank)

Require: Set of association matrixes M_{AP}, M_{PU}, M_{UA}, M_{PS}.
Ensure: S^* the converged Social Web Service Rank

while S_i not converged **do**
  $P_i = M_{PS} * S_i$
  $U_i = M_{PU}^T * P_i$
  $A_i = M_{UA}^T * U_i$
  $P_i' = M_{AP}^T * A_i$
  $A_i' = M_{AP} * P_i'$
  $U_i' = M_{UA} * A_i'$
  $P_{i+1} = M_{PU} * A_i'$
  $S_{i+1} = M_{PS}^T * P_{i+1}$
end while

3 Experiments

We performed a set of experiments using three datasets, a social dataset build using del.icio.us data and 2 web service datasets. The social dataset was created by crawling del.icio.us data. Once crawled, the data was annotated using the UTO ontology and stored as RDF triples in a Sesame repository. The del.icio.us dataset has in total 1.64 GB.

The second and third datasets are .wsdl datasets that were kindly provided to us by seekda OG. The first one is a 80 files dataset containing services providing shipping functionality having valid wsdl descriptions and being accessible online. The second one is a larger dataset containing 5000 services from different domains, as well valid .wsdl files and available online. For the 80 shipping services dataset we found that only 3 of their domains have pages annotated in del.icio.us.

For each of these domains we query Sesame and determine the total number of web pages annotated to be 14. The distribution of web pages per web service domains is between 0 and 7 web pages.

For the 5000 services dataset 86 of them have web pages from their domain annotated with del.icio.us.

The number of web pages per domain varies from 0 to 131. In total we found 547 web pages annotated in del.icio.us that are connected to the initial dataset.

One can notice that in both cases the number of pages annotated in del.icio.us which have a correspondence with the web services set is rather limited. However for a small number of services we were able to find considerable number of pages some of them having a strong correlation/similarity with the web services. We can conclude that even though the link between the web service data and del.icio.us data is in general not so strong, when such a link is visible, it can be used to determine more information about the services and also to better rank them.

4 Related Work

Ranking various types of entities has been a challenging research problem over the years. Various approaches have been developed in order to improve ranking algorithms and ranking results. Ranking gained popularity especially in the context of improving web search experience of user. Classical approaches in this area are using statistical information such as term frequency, document length, etc. to compute the similarity degree of a document and a query.

An interesting approach which uses social annotations to improve web search has been proposed in [2]. The work was motivated by two basic observations: (1) annotations are usually good summaries of the corresponding pages and (2) the number of annotations indicates the popularity of the web pages. Two algorithms that use the previous observations and extend page ranking have been proposed defining how the similarity between queries and annotations can be computed, respectively what is the popularity of a web page based on its relation with annotations and users.

A classification of various types of web service ranking approaches is proposed in [7]. The authors distinguish two distinct categorization axes, one measuring the localness of the ranking approach depending on whether local or global knowledge is used in computing ranking values, the second measuring the absoluteness of the ranking approach depending on whether the measurement is of absolute scope or refers to a particular request. Some of the most prominent categories for web service ranking are *hub-authorities based ranks* which examine the relation between the number of services that link to a specific service (in-degree) and the number of services that service links to (out-degree) and *non-functional ranks* which use the non-functional aspects to compute the rank of services. The first category of methods exploits the given in-/out relations between entities computing rank values using global knowledge. The most prominent representative is PageRank [12]. In the second category the approaches such as [13], [9], [14], [8] can be included. In [13] for example a multi-criteria QoS model is used to determine the importance or rank of a service. Quality vectors are built for each service and a correspondence matrix between services and QoS. A simple additive weighting method is afterwards applied to select the optimal Web service.

5 Discussions

Using Web 2.0 data to rank web services can be addressed in a similar manner as described in this paper but adopting a different starting point. Instead of starting from web service data, a social web service ranking approach can start from Web2.0 social data. User's query will be first directed to the del.icio.us Semantic Repository. The domain of each relevant bookmark found in del.cio.us repository is crawled and checked for .wsld files. Those web pages which have .wsdl files in their domain are ranked based on how many people have bookmarked them on del.icio.us. The same correspondences between bookmarked web pages and web services could be used but in a reverse manner as proposed by our approach.

Nowadays there are many web service search engines such as seekda, stikeiron, etc. which have already collected large web services collections. Our approach could be easily integrated with such search engine extending their ranking mechanisms by using social information collected from Web2.0 sources.

As pointed out in [7], and reconfirmed in our experiments many web pages, and in consequence related web services, may not have annotations. These pages are not considered by the system and thus possible associated web services can not be ranked using our approach. However annotation systems tend to be sensitive to newly created web pages or pages which are not already in the system thus partially closing the existing gap.

6 Conclusions

In this paper we investigate the use of social annotations from Web 2.0 sources in web service ranking. More precisely we use the annotation data from one of the biggest social network del.icio.us and we discover and rank web services connected to del.icio.us bookmarks. Following this straightforward idea we have design and implemented a running prototype. Furthermore a global algorithm to compute the social rank of web services is proposed.

As future work we plan to compare and integrate our system with other approaches for ranking services. We are mainly interested to know *if* and *in which* situations social based service rankers perform better than rankers using properties/descriptions of services such as Non-functional properties. The focus will not be on the comparison itself but rather on defining an integrated framework for ranking web services which is able to select and use in relevant situations the results of various ranking approaches including Non-functional properties/description based and Social based rankers.

Also left as future work is the full implementation and validation of *SocialWSRank* algorithm described in Section 2.3 which globally computes rank values for available services. More exhaustive experiments will be performed once a reference ranking order for the web services datasets used in our experiments is defined. We are in the process of building with the help of students and volunteers the reference rankings for the two sets.

7 Acknowledgements

The authors would like to thank Michael Fried for his support with the implementation. Further, we especially thank seekda OG for providing the web service data set used in our experiments.

References

[1] Del.icio.us official blog: http://blog.del.icio.us/blog/2006/09/million.html.

[2] Shenghua Bao, Guirong Xue, Xiaoyuan Wu, Yong Yu, Ben Fei, and Zhong Su. Optimizing web search using social annotations. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 501–510, New York, NY, USA, 2007. ACM.

[3] Dan Brickley and R.V. Guha. Resource Description Framework (RDF) Schema Specification 1.0. Recommendation, World Wide Web Consortium, February 2004. Available from http://www.w3.org/TR/2004/REC-rdf-schema-20040210/.

[4] Jeen Broekstra, Arjohn Kampman, and Frank van Harmelen. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In *In Proceedings of the 2nd International Semantic Web Conference, Sardinia, pages 54-68. Springer, 2002.*, 2002.

[5] Ying Ding, Sin-Jae Kang, Ioan Toma, Michael Fried, Omair Shafiq, and Zhixian Yan. Adding Semantics to Social Tagging: Upper Tag Ontology (UTO). In *Proceedings of the 70th Annual Meeting of the American Society for Information Science & Technology (ASIS&T), Oct 24-29, 2008, Columbus, Ohio, USA*, 2008.

[6] Ying Ding, Ioan Toma, Sin-Jae Kang, Michael Fried, and Zhixian Yan. Data Mediation and Interoperation in Social Web: Modeling, Crawling and Integrating Social Tagging Data. In *WWW2008 Workshop on Social Web Search and Mining (SWSM2008), April 21-25, 2008, Beijing, China.*, 2008.

[7] John Gekas. Web Service Ranking in Service Networks. In *Poster paper at 3rd European Semantic Web Conference ESWC '06*, pages 501–510, 2006.

[8] C L Hwang and K Yoon. Multiple attribute decision making: Methods and applications. Springer-Verlag, 1981.

[9] Yutu Liu, Anne H. Ngu, and Liang Z. Zeng. QoS Computation and Policing in Dynamic Web Service Selection. In *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 66–73, New York, NY, USA, 2004. ACM.

[10] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, July 2008.

[11] Frank Manola and Eric Miller. RDF Primer. Recommendation, World Wide Web Consortium, February 2004. Available from http://www.w3.org/TR/REC-rdf-syntax/.

[12] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[13] Liangzhao Zeng, Boualem Benatallah, Anne H.H. Ngu, Marlon Dumas, Jayant Kalagnanam, and Henry Chang. QoS-Aware Middleware for Web Services Composition. *IEEE Trans. Softw. Eng.*, 30(5):311–327, 2004.

[14] Chen Zhou, Liang-Tien Chia, and Bu-Sung Lee. Semantics in Service Discovery and QoS Measurement. *IT Professional*, 7(2):29–34, 2005.