Routledge
Taylor & Francis Group

Library Collections, Acquisitions, and Technical Services

ISSN: 1464-9055 (Print) 1873-1821 (Online) Journal homepage: https://www.tandfonline.com/loi/ulca20

Advanced information retrieval Web services for digital libraries

Weimao Ke, Yueyu Fu & Javed Mostafa

To cite this article: Weimao Ke, Yueyu Fu & Javed Mostafa (2005) Advanced information retrieval Web services for digital libraries, Library Collections, Acquisitions, and Technical Services, 29:2, 220-224, DOI: 10.1080/14649055.2005.10766053

To link to this article: https://doi.org/10.1080/14649055.2005.10766053

Available online at www.sciencedirect.com
SCIENCE DIRECT®
Library Collections, Acquisitions, & Technical Services 29 (2005) 220–224
ELSEVIER
Library Collections, Acquisitions, & Technical Services

Advanced information retrieval Web services for digital libraries

Weimao Ke*, Yueyu Fu, Javed Mostafa

Laboratory of Applied Informatics Research, Indiana University, Bloomington, IN 47405-3907, USA

Available online 11 July 2005

Abstract

Web service as a standardized XML-based protocol has been useful for inter-system communication and integration. However, Web services in the IR domain have not been widely used. In a previous paper [Fu, Y., & Mostafa, J. (2004). Toward information retrieval Web services for digital libraries. *IEEE/ACM Joint Conference on Digital Libraries 2004.* Tucson, Arizona], we discussed a system supporting several information retrieval (IR) functions. This system called LUCAS is a Web service for extracting, weighing, and ranking terms. In this paper, we are going to discuss a more advanced version of the system called Lucas II. This updated implementation includes functions of term generation, clustering, and document classification that can be applied to different knowledge domains.

Keywords: Web services; Information retrieval; Online information services; Web-based services

1. Introduction

In this paper, we concentrate on three important IR functions (term generation, cluster generation, and document classification) and demonstrate how these can be offered as

* Corresponding author.
E-mail addresses: wke@indiana.edu (W. Ke), yufu@indiana.edu (Y. Fu), jm@indiana.edu (J. Mostafa).

1464-9055/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.lcats.2005.04.015

Web services for supporting basic digital library functions. The paper will briefly discuss the importance of the implemented IR functions and outline the architecture of our system. It will describe primary parameters of the Web services and give examples of accessing them in multiple ways. The paper will conclude with a discussion of future work.

2. Importance of the functions

The Web services offered by Lucas are fundamental functions of IR. They can be applied to a variety of digital library practices [1,4]. For instance, term generation can be used for indexing, information extraction, and summarization while clustering is useful in searching [2]. In addition, classification is important as related to indexing and filtering. Web services encapsulating these functions are widely applicable to projects in this area.

3. Architecture

The system mainly consists of three components: the Lucas II Web services, which are deployed on Tomcat and Apache Axis server; the client, which passes the user-selected parameters to the Web services and gets the results back based on the SOAP protocol; and data access modules to access domain terms and document collections. We refer to this system as Library of User-Oriented Concepts for Access Services II (i.e., LUCAS II) (Fig. 1).

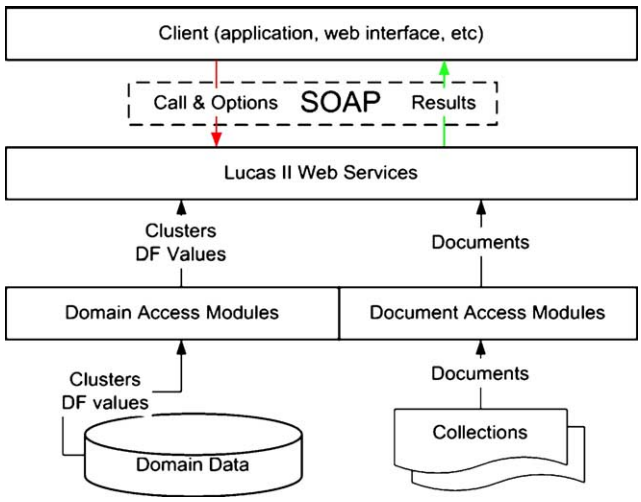

Fig. 1. System architecture.

4. Web services

In Lucas II, we developed and deployed three Web service methods (operations). These services are described below.

4.1. Term generation

The term generation function retrieves terms from a given online document (URL), computes the term weights, and sorts the result. The algorithm follows:

(1) Extract all the terms except the stop words from a given document and count term frequency.
(2) Retrieve the DF values from the domain DB for all the extracted terms and compute TF*IDF weights.
(3) Sort the terms based on the TF*IDF values and select a number of the top terms specified by the user.

The method for term generation is *generateTerms* with the following parameters: (1) *domain*: specifies the knowledge domain for DF values; (2) *URL*: the URL of a Web document; (3) *numberTerms*: the number of top terms requested; (4) *showWeights*: to show the weights of terms or not; and (5) *format*: format of the result [0 Text | 1 HTML]. The service returns a list of the extracted terms.

4.2. Cluster generation

This function has two steps: term extraction and term clustering. It is based on an algorithm developed by Mostafa, Quiroga, and Palakal, [3] which employs a word distribution weighing scheme and some heuristics to extract terms. The process is defined below:

(1) Extract common terms that are among the top weighted terms throughout a certain subset of documents in the collection.
(2) Compute term–term associations based on the extracted list of terms and the doc-term matrix of the collection.
(3) Apply a distance threshold and cluster the terms with centroids.

The method for cluster generation is *generateClusters* with parameters as follow: (1) *domain*: the knowledge domain; (2) *R*: consider the top R ranked tokens in each document; (3) *D*: the percentage of documents that must contain a token ranked above R for the token to be selected; (4) *theta*: vectors must be theta far away from all existing centroids to be considered a new centroid; and (5) *numberClusters*: number of clusters requested. There are other parameters for formatting the result. Lucas II returns a list of clusters and their member terms.

4.3. Document classification

This Web service classifies an online document (URL) into a proper cluster after computing its similarity scores with a set of term clusters:

(1) Convert the list of term clusters into cluster-term vectors.
(2) Retrieve the document content and use the unique terms in the clusters to render its doc-term vector (binary, frequency, or TF*IDF representation).
(3) Compute the similarity score between the document and each of the clusters using Dice or Cosine algorithm.
(4) Sort the clusters based on similarity scores and choose the top cluster.

The method for document classification is *classifyURL* with parameters as follow: (1) *domain*: the knowledge domain where DF values can be obtained; (2) *docURL*: the URL of a document; (3) *clusterString*: a list of term clusters that can also be generated through the cluster generation Web service; (4) *repAlgorithm*: representation model [0 Binary | 1 Term Frequency | 2 TF-IDF]; and 5) *classAlgorithm*: classification algorithm [0 Dice | 1 Cosine]. The service returns the best-matched cluster and the similarity score of the document to each cluster.

For more information about Lucas II Web services, please refer to http://tara.slis.indiana.edu:8080/lucas2/lucas2.html.

5. Web service clients

There are multiple ways to invoke these Web services. Based on our Web service description and the SOAP protocol, new client interfaces can be easily built according to users' preferences. One way is to use a Java application. A sample Java code for the cluster generation service can be downloaded at http://tara.slis.indiana.edu:8080/lucas2/LucasClient2.java.

Another way is to use JSP/Servlet to enable accessing these Web services on any Web browser. As shown in Fig. 2, a user can select options through the Web interface and submit the requests to a JSP/Servlet component, which then communicates with the "classifyURL" Web service and transfers the result back to the browser. This demo JSP client can be accessed at http://tara.slis.indiana.edu:8080/lucas2/lucas2class.jsp.

6. Conclusion

Our system implementation has demonstrated that IR algorithms can be effectively turned into Web services that can be accessed in a variety of ways. This flexibility enables easier integration of IR systems and/or algorithms without duplicating efforts. Future work involving LUCAS will integrate its Web services with more sophisticated IR and digital

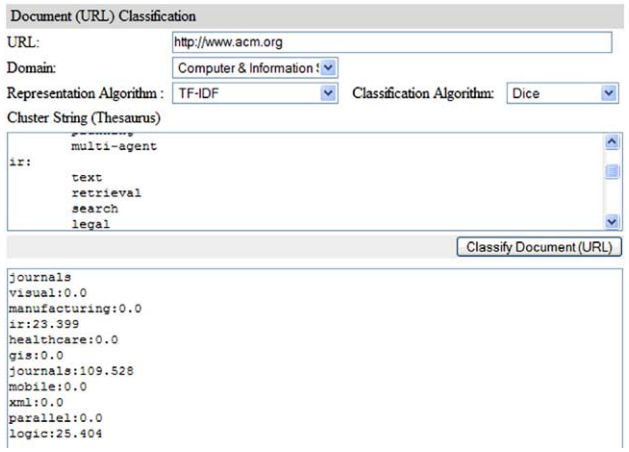

Fig. 2. A JSP/Servlet client for document classification.

library operations. In fact, the term generation and classification services have been successfully integrated into one of our digital library projects called ENABLE to generate index terms and classify Web pages automatically. For more information about the ENABLE project, please visit http://enable.slis.indiana.edu.

Acknowledgment

This work was partially supported through a grant from the National Science Foundation Award#:0333623.

References

[1] Chen, H. (1999). Semantic research for digital libraries. *D-Lib Magazine, 5*(10).
[2] Fu, Y., & Mostafa, J. (2004). Toward information retrieval Web services for digital libraries. *IEEE/ACM Joint Conference on Digital Libraries 2004.* Tucson, Arizona.
[3] Mostafa, J., Quiroga, L., & Palakal, M. (1998). Filtering medical documents using automated and human classification methods. *Journal of the American Society for Information Science, 49*(14).
[4] Truner, M., Budgen, D., & Brereton, P. (2003). Turning software into a service. *IEEE Computer, 36*(10).