ServiceXplorer: A Similarity-based Web Service Search Engine

Anne H.H. Ngu, Scott Julian
Department of Computer Science
Texas State University
San Marcos, Texas 78666, USA
{angu, sjulian}@txstate.edu

Jiangang Ma, Quan Z. Sheng, Lina Yao
School of Computer Science
The University of Adelaide
Adelaide, SA 5005, Australia
{mike, qsheng, lina}@adelaide.edu.au

ABSTRACT

Finding relevant Web services and composing them into value-added applications is becoming increasingly important in cloud and service based marketplaces. The key problem with current approaches to finding relevant Web services is that most of them only provide searches over a discrete set of features using exact keyword matching. The Earth Mover's Distance (EMD) algorithm has been used successfully in multimedia databases for partial match retrieval among images with uneven feature distributions. The only downside of EMD is that it is computationally expensive and can result in slower performance for very large data sets. We demonstrate in this paper that by utilizing well known indexing scheme such as inverted file and R-tree indexes over Web services attributes, EMD can be used efficiently to find partial matches between a query and a database of Web services. We also demonstrate that our EMD-based approach performs better in precision than the Vector Space Model in retrieving relevant services.

Categories and Subject Descriptors

H.3.5 [**On-line Information Services**]: Web-based services; H.3.4 [**Systems and Software**]: Distributed systems

General Terms

Management, Performance

Keywords

Web Services, Search Engine, Earth Mover's Distance, Web Services Discovery

1. INTRODUCTION

Similarity-based search of Web services has been a challenging issue over the years. Consider in the case of a composite service where one of its component service fails, we need to search and find the best replacement for it. Moreover, similarity search is also a crucial task in many other practical applications, such as data mining within cluster applications, and Google Maps search to locate businesses.

Currently, most existing services search engines such as Woogle [1], WSExpress [6], and Titan systems [5] adopt keyword based search strategy because it is easy to implement and work well in some cases. However, the traditional keyword-based service search engines still have several limitations when they are applied to Web services. First, due to the heterogeneous naming conventions used by different service providers, keyword-based approach may result in a large term dictionary especially when scaling to massive number of Web services. Second, as Web service descriptions are typically comprised of limited terms, the resultant term vectors will become extremely sparse in the space of distinct terms used by all services. Matching services based on traditional similarity measures (e.g., cosine similarity) leads to poor results as large sparse term vectors are less likely to coincide on common terms. Furthermore, traditional keyword-based search engines compute a similarity score by only matching the keywords at the same positions in both the query and the document vectors, without considering the impact of the keywords at neighboring positions.

To overcome the limitations mentioned above, we have developed an EMD-based similarity search algorithm [3], for finding similar Web services based on partial matching of Web service attributes (e.g., operations, messages, description, service name) or a combination of them. EMD has been widely employed in multimedia databases [4] to better approximate human visual perception, speech recognition, and document retrieval because EMD can effectively capture the differences between the distributions of two objects' main features, and allow for partial matching. This means if keywords in a query only match partially the keywords in a service record, EMD-based similarity search is still able to find the most similar matching pairs of words. To overcome the limitation of EMD-based computation complexity, we exploit filtering techniques to minimize the total number of actual EMD computations. Furthermore, we have developed a *generalized independent minimization lower bound* as a new EMD filter for partial matching technique that is suitable for searching Web services. The filter is then incorporated into a k-NN algorithm for producing top-k results.

In this paper, we demonstrate an EMD-based Web services search engine: `ServiceXplorer`, which implements the techniques presented in [3]. ServiceXplorer adopts a two-step approach for EMD based Web service search: *filter* and *refine*. The filter step selects potential candidate Web service records from a database by removing those unqualified Web service records. In this step, an indexing scheme (inverted file index, R-tree) is used to find the nearest leaf nodes to the query when a query is evaluated. At the end of this step, all records belonging to the found nodes are considered to be the potential candidates, forming a potential candidate set (PCS). The refine step computes the similarity between the query and Web service records in PCS based on EMD distances, and picks up the top-k most similar results. In the following sections, we overview the design and implementation of ServieXplorer and report the proposed demonstration.

2. SERVICEXPLORER: AN OVERVIEW

In this section, we overview the system implementation and the techniques on EMD-based similarity search of Web services.

Functional-related (FR) Attributes					Non-FR Attributes		
ID	Name	Category	Function	Description	RT (s)	AVA (%)	REL (%)
1	B.S. Stillwell Ford	New cars	getCarPrice	Finance facilities at competitive rates	1.2	86	88
2	City Holden	Used cars	getCarColor	Minimize the hassle of a second hand vehicle	0.8	92	80

Table 1: Web service records

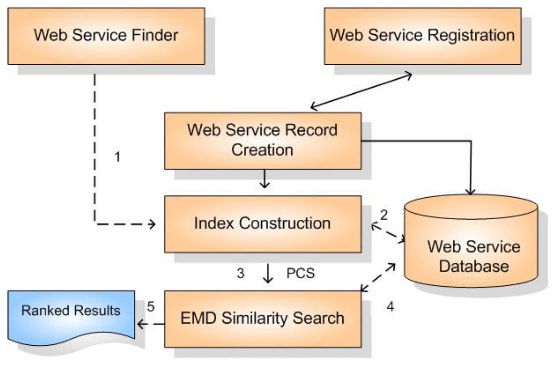


Figure 1: System architecture of ServiceXplorer

2.1 System Architecture

The proposed architecture (see Figure 1) consists of four main components: *Web service registration*, *Web service record creation*, *index construction*, and *EMD similarity search*. The Web service registration component provides a web-based GUI where users can perform several activities such as uploading a WSDL file, and entering short descriptions and categories/tags for the uploaded WSDL file. Currently, we store the information extracted from WSDL files in a Web service database which is implemented in MySQL. The `WSRecords` is the main relation which is being referred by all the other relations. It stores the unique service key, service name, and the description. Each WSDL can have multiple messages, operations, QoS, and categories, which are stored in `WSOperation`, `WSMessage`, `WSQoS`, and `WSCategory` respectively.

The index construction component is responsible for creating the inverted file index for the registered Web services. The inverted file index list is implemented as a normalized relation with both WSDL unique term and the service key as the primary key. Note that we do not keep track of term frequency since the inverted index is used purely for filtering purpose, not for ranking relevancy of a term in a Web service document. For the non-functional quality of service (QoS) attributes such as reliability and availability, their values can be expressed as a range. Therefore, R-tree indexes are created on `WSQoS` by using the *create spatial index data* definition statement in MySQL. The collection of QoS is done by a separate monitoring component which is not within the scope of this paper. The interested readers are referred to [2].

2.2 EMD-based Similarity Search

In [3], we introduce an EMD-based optimization algorithm to support similarity search. Formally, given a database **DB** containing Web service records and a query q, a similarity search returns all records $r \in \mathbf{DB}$, such that $dis(q, r) \leq \varepsilon$, where $dis(q, r)$ is a distance function, and ε is the specified similarity threshold. In particular, it includes three main sub-search tasks: i) modeling of Web services, ii) defining similarity measure, and iii) designing a new lower bound to ensue correct search results.

First, we model Web services as *records* stored in a relational database to facilitate effective similarity search. Thus, all Web service records form a Web service database **DB**. Formally, a Web service record $r \in \mathbf{DB}$ is represented by a tuple $r = \langle n^r, c^r, f^r, d^r, \mathbf{QoS}^r \rangle$, where n^r, c^r, f^r, and d^r contains the content of the attribute "name", "category", "operation", "description", and $\mathbf{QoS}^r = \{QoS_1^r, QoS_2^r, \ldots\}$ represents QoS such as response time (RT), availability (AVA) and reliability (REL). Some examples of Web service records are shown in Table 1.

Second, we use EMD as a distance measure, $dis_{EMD}(q, r)$, between a query q and a record r, which describes their similarity:

$$EMD(\mathbf{q}, \mathbf{r}) = \frac{\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} f_{ij}^* d_{ij}}{\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} f_{ij}^*}$$

where $\mathbf{f}^*$ is the optimal flow that involves computing the optimal solutions by using the following linear program (LP) with variable f_{ij}:

$$\begin{array}{ll} \text{minimize} : \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} f_{ij} d_{ij} & \\ \text{subject to} : & \\ \forall 1 \leq i \leq n_1 : \sum_{j=1}^{n_2} f_{ij} \leq w_{kw_i} & (1.1) \\ \forall 1 \leq j \leq n_2 : \sum_{i=1}^{n_1} f_{ij} \leq w_{aw_j} & (1.2) \\ \forall 1 \leq i \leq n_1, 1 \leq j \leq n_2 : f_{ij} \geq 0 & (1.3) \\ \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} f_{ij} = \min \left(\sum_{i=1}^{n_1} w_{kw_i}, \sum_{j=1}^{n_2} w_{aw_j} \right) & (1.4) \end{array}$$

Third, we have developed a filtering technique using a new lower bound LB_{GIM} to minimize the total number of actual EMD computations. The LB_{GIM} is a conditional lower bound depending on cases of LP introduced above, and we theoretically show that the LB_{GIM} lower bounds the general EMD. The detailed proof can be found in [3].

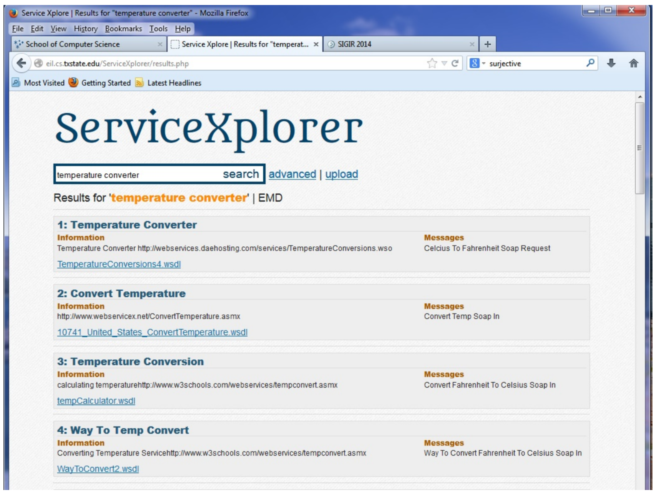
3. DEMONSTRATION

In this section, we demonstrate the functionality of ServiceXplorer and report experimental results. ServiceXplorer can find desired services by providing users with simple search interfaces and effective search algorithms with EMD. In particular, we demonstrate the search functions through three main search scenarios: *service registration*, *simple similarity search*, and *advanced similarity search*. The system can be accessed from: `http://eil.cs.txstate.edu/ServiceXplorer`.

Service Registration. Service providers can register their services through the registration interface, provided by ServiceXplorer. Through the interface, the users can choose appropriate services, specify the function descriptions of the selected services, and provide related tags. The selected services will be stored in the system after pressing the upload button. Once registered in the system, the services are further refined and stored as records in a database for effective similarity search.

Simple Similarity Search. In this scenario, ServiceXplorer handles the similarity search of Web services by using EMD as the underlying similarity distance only. In the simple similarity search interface, a user can type a single keyword or multiple keywords, and our system will return the relevant services to the user. From Figure 2 we can see that using EMD similarity strategy, there is a higher probability that the top results are always the most relevant ones. For example, for the query of `Temperature Converter`, the second top result is `Convert Temperature` while using VSM (vector space model), that particular service is only ranked the 23rd.

Advanced Similarity Search. ServiceXplorer also offers an advanced similarity search that enables users to locate services by selecting different index structures, specifying QoS parameters and comparing the search performance with that of VSM. For example, if a user chooses EMD search with index structure, she can simply click the radio button (`With`) to



Service Xplore | Results for "temperature converter" - Mozilla Firefox

File Edit View History Bookmarks Tools Help

School of Computer Science | Service Xplore | Results for "temperat... | SIGIR 2014

eil.cs.txstate.edu/ServiceXplorer/results.php

surjective

Most Visited | Getting Started | Latest Headlines

ServiceXplorer

temperature converter search advanced | upload

Results for 'temperature converter' | EMD

1: Temperature Converter

Information
Temperature Converter http://webservices.daehosting.com/services/TemperatureConversions.wso
TemperatureConversions4.wsdl

Messages
Celcius To Fahrenheit Soap Request

2: Convert Temperature

Information
http://www.webservicex.net/ConvertTemperature.asmx
10741_United_States_ConvertTemperature.wsdl

Messages
Convert Temp Soap In

3: Temperature Conversion

Information
calculating temperaturehttp://www.w3schools.com/webservices/tempconvert.asmx
tempCalculator.wsdl

Messages
Convert Fahrenheit To Celsius Soap In

4: Way To Temp Convert

Information
Converting Temperature Servicehttp://www.w3schools.com/webservices/tempconvert.asmx
WayToConvert2.wsdl

Messages
Way To Convert Fahrenheit To Celsius Soap In

Figure 2: Scrrenshot on using ServiceXplorer

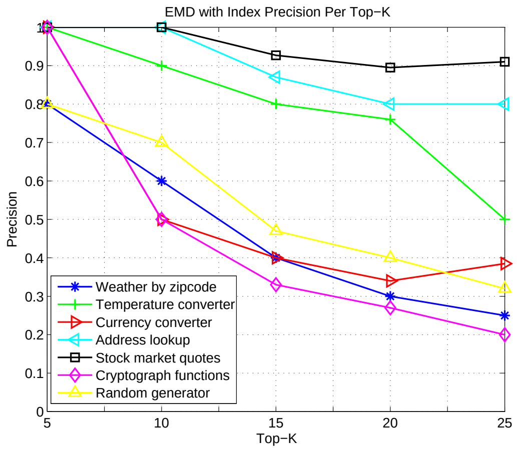


Figure 3: EMD with top-k

select the index. The input query will be processed against the inverted index list by filtering unqualified services and then executed against services records in the database to locate desired Web services using EMD similarity search.

Experimental Results. We evaluated the performance of the EMD algorithm against the Vector Space Model (VSM). The experiments were conducted by using over 4,500 real Web services collected from the Web. All our experiments were conducted on a Dell Optiplex 760 running Windows 7, E8400 Core 2 Duo processor with 4GB RAM. We set the maximum number of results to be returned to 40 in all our experiments. The similarity coefficient threshold is set to the default zero.

For the VSM algorithm, we used the MySQL implementation of full-text search. We submitted the queries `weather by zipcode`, `temperature converter`, `currency converter`, and `address lookup` respectively to the ServiceXplorer and recorded the precision of search using EMD in conjunction with an inverted index, and VSM. Figure 3 and 4 show the precision of EMD and VSM at top-k increments of 5, 10, 15, 20, 25. Our experiments show that using EMD in conjunction with an inverted index has an average precision of 0.613 whilst the average precision of using VSM is only 0.369. Thus EMD is more effective than VSM for searching Web services.

We also evaluated the scalability of the EMD algorithm when coupled with inverted file and the R-tree indexes. Figure 5 compares the number of EMD calculations with indexes using the same set of queries as the previous test. Without any indexing scheme, the total number of EMD calculations is 4,500 (same as the number of WS records in the database). With indexing, the largest PCS (potential candidate set) is just 85 with the `address lookup` query. Our experiment shows that the scalability of the EMD algorithm

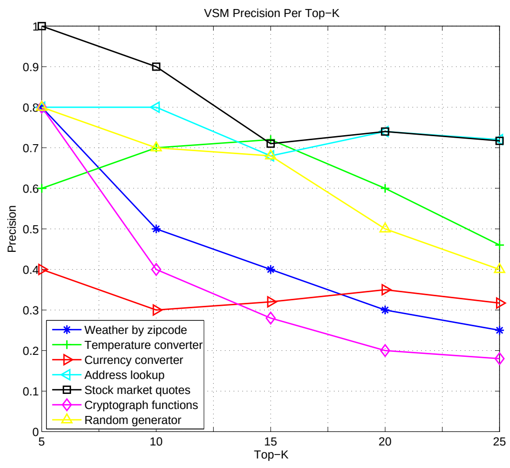


Figure 4: VSM with top-k

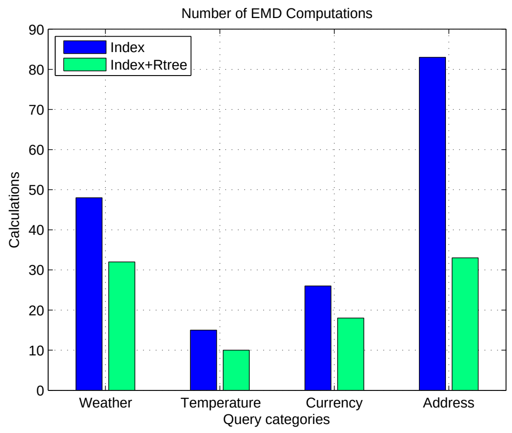


Figure 5: Scalability of indexed EMD

can be achieved by applying a filtering step through the combination of various indexing mechanisms.

4. CONCLUSION

In this paper, we have presented ServiceXplorer, a system that exploits an EMD-based algorithm for effective similarity search of Web services. Our experimental studies on real Web services show that by incorporating an inverted file index, we can significantly reduce the number of EMD computations without affecting the precision of the search. Interested readers are referred to the project website[1] for more details.

5. REFERENCES

[1] X. Dong et al. Similarity Search for Web Services. In *Proc. of the 30th Intl. Conf. on Very Large Data Bases (VLDB)*, 2004.

[2] Y. Liu, A. H. Ngu, and L. Zeng. QoS Computation and Policing in Dynamic Web Service Selection. In *Proc. of the 13th Intl. World Wide Web Conf. (WWW)*, 2004.

[3] J. Ma, Q. Z. Sheng, K. Liao, Y. Zhang, and A. H. Ngu. WS-Finder: A Framework for Similarity Search of Web Services. In *Proc. of the 10th Intl. Conf. on Service Oriented Computing (ICSOC)*, 2012.

[4] Y. Rubner, C. Tomasi, and L. Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. *Intl. Journal of Computer Vision*, 40(2):99–121, 2000.

[5] J. Wu, L. Chen, Y. Xie, and Z. Zheng. Titan: A System for Effective Web Service Discovery. In *Proc. of the 21st Intl. World Wide Web Conf. (WWW)*, 2012.

[6] Y. Zhang, Z. Zheng, and M. Lyu. WSExpress: A QoS-aware Search Engine for Web Services. In *Proc. of IEEE Intl. Conf. on Web Services (ICWS)*, 2010.

[1]`http://eil.cs.txstate.edu/ServiceXplorer`