# A Service Concept Recommendation System for Enhancing the Dependability of Semantic Service Matchmakers in the Service Ecosystem Environment

Hai Dong, Farookh Khadeer Hussain, and Elizabeth Chang

Digital Ecosystems and Business Intelligence Institute, Curtin University of Technology

Enterprise Unit 4, De Laeter Way, Technology Park, Bentley, WA 6102, AUSTRALIA

**Abstract**— A Service Ecosystem is a biological view of our business and software environment, which is comprised of a Service Use Ecosystem and a Service Supply Ecosystem. Service matchmakers play an important role in ensuring the connectivity between the two ecosystems. Current matchmakers attempt to employ ontologies to disambiguate service consumers' service queries by semantically classifying service entities and providing a series of human computer interactions to service consumers. However, the lack of relevant service domain knowledge and the wrong service queries could prevent the semantic service matchmakers from seeking the service concepts that can be used to correctly represent service requests. To solve this issue, in this paper, we propose the framework of a service concept recommendation system, which is built upon a semantic similarity model. This system can be employed to seek the concepts used to correctly represent service consumers' requests, when a semantic service matchmaker finds that the service concepts that are eventually retrieved cannot match the service requests. Whilst many similar semantic similarity models have been developed so far, most of them focus on distance-based measures for the semantic network environment and ignore content-based measures for the ontology environment. For the ontology environment in which concepts are defined with sufficient datatype properties, object properties, and restrictions etc., the content of concepts should be regarded as an important factor for concept similarity measures. Hence, we present a novel semantic similarity model for the service ontology environment. The technical details and evaluation details of the framework are discussed in this paper.

**Index Terms**— semantic similarity models, semantic service matchmakers, service concept recommender system, Service Ecosystem, service ontology

## 1 INTRODUCTION

The term "*Service Ecosystem*" has emerged from Veryard's book [1], in which the author utilized a biological view to analyse the components of our business and software environment. The Service Ecosystem contains both human users and artefacts which can be divided into two major components: 1) services that would be meaningful and viable in the ecosystem; and 2) devices that enable the release and delivery of these services into the ecosystem. Services and devices in the Service Ecosystem are heterogeneous by nature and cover nearly all commercial and electronic services and devices in human society. Moreover, in accordance with the notion of demand/use and supply, Veryard separated the ecosystem into four components with respective internal activities described as follows:

1. Service Use Ecosystem, in which services are demanded, used, and use of services are architected and configured, as well as service publications being subscribed by human users and artefacts;
2. Service Supply Ecosystem, in which human users and artefacts architect, configure, publish, provide or deliver services through stable interfaces,
3. Device Use Ecosystem, in which devices are configured, installed, connected or called, and device behavior and system behavior are predicted;
4. Device Supply Ecosystem, in which devices are architected, provided and managed.

In addition, there are several factors that influence the above activities within each ecosystem. Availability is one of them, which is the ecological principle of the Service Supply Ecosystem. Availability refers to the accessibility, ease-of-use and reliability of services for supplying [1]. Furthermore, availability also impacts upon the connectivity between the Service Use Ecosystem and the Service Supply Ecosystem.

In order to ensure the availability of service supply, one important factor is to enable a service entity to be precisely matched with a service request. Whilst currently there are plenty of commercial search engines available for the service matchmaking, there is a widespread problem that occurs in most of the search engines, namely, an ambiguous service request cannot match an appropriate service entity. This is also the reason why many ontology-based service matchmakers have been developed. These service matchmakers utilize service domain ontologies for classifying service entities, by associating service entities with appropriate service ontology concepts, which enriches the semantical extent of service entities [2-9]. Moreover, these matchmakers provide a particular service domain knowledge for service consumers, in order to disambiguate their service queries. This is realized by: 1) keyword-based automatic matching between a service request and one or more service concepts [2, 3, 6-9]; and 2) service consumers choosing the service concepts that can best present their service requests from the matched concepts by means of a series of Human Computer Interactions (HCIs) with service ontologies [4]. As a consequence, when a service query finally matches one or more service concepts, the service entities associated with the service concepts can be matched and retrieved by the ontology-based service matchmakers.

However, the principle of the ontology-based service matchmakers gives rise to the following research question – *If a* *service consumer does not have relevant knowledge about his/her service request, and thus his/her service query could be incorrect or incomplete at the beginning, are the semantic service matchmakers able to disambiguate his/her service query?* The answer could be '*no*', because incorrect or incomplete service queries and the lack of relevant service domain knowledge could lead to the ontology-based service matchmakers being incapable of finding appropriate service concepts for the service consumer.

In order to resolve this issue, first of all, whereas the eventually retrieved service concept(s) cannot match the service consumer's service request, this concept(s) is obtained by the HCIs between the service consumer and an ontology-based service matchmaker. Therefore, we can premise that there is some extent of overlap between the retrieved service concept(s) and the service concept(s) that can correctly represent the service consumer's service request, as a result of the impact of the service consumer's subjective perception on the retrieval of this service concept(s). As a consequence, we propose to study the ontology-based service matchmaking from the perspective of measuring the similarity of concept-concept, instead of measuring the similarity of query-concept utilized by those semantic matchmakers. This task can be accomplished by utilizing a semantic similarity model. This model can be used to seek semantically similar service concepts for a given concept in a service ontology environment, which can be used as a remediation when a semantic matchmaker finds that the retrieved service concept cannot match a given service request.

As a matter of fact, many semantic similarity models have been developed in the field of information retrieval (IR) [10-12], natural language processing (NLP) [13-16], medicine [17], and bioinformatics [18-20] etc. However, there is a crucial issue here that needs research attention – most of these models are designed for the semantic network environment but not for the semantically-rich ontology environment. In order to address the above issues, in this paper, we propose a service concept recommendation system based on a novel semantic similarity model, with the purpose of enhancing the dependability of the semantic service matchmakers. This model integrates the approaches from the perspective of a concept content-oriented measure and an ontology structure-oriented measure.

The remainder of the paper is structured as follows: we give a brief overview of related works in Section 2. In Section 3, we deliver the system architecture of the service concept recommendation system. In Section 4, a lightweight ontology model is presented in order to allow ontologies to be adapted to the forthcoming hybrid semantic similarity model. In Section 5, we present the hybrid semantic similarity model. Section 6 contains the evaluation of the model. Conclusions are drawn and future work is proposed in Section 7.

## 2 RELATED RESEARCH WORK

In this section, we will conduct a general survey of the recent works related to semantic service matchmakers, service recommendation systems and semantic similarity models.

### 2.1 Semantic Service Matchmakers

Semantic service matchmakers are more concerned with matching a set of service advertisements with a service request by means of ontology description languages, such as Resource Description Framework (RDF), Web Ontology Language (OWL) etc. in order to determine which service can possibly fulfil the request. The major approach is to use semantic descriptions to create the semantic-rich frameworks for advertised services and service requests. The following are the typical examples in this domain.

Various researchers employ ontologies to describe service descriptions for service matchmaking.

Dong et al. [4] designed a semantic service search engine that uses domain-specific service ontologies to classify and describe services. A service is represented by a RDF/OWL-annotated service metadata. The service metadata will then be matched with the concepts of a service domain ontology in terms of computing the similarity between the service descriptions of the service metadata and the concept descriptions of the service concepts. Therefore, the services are described by the conjunction of the matched service concepts. When a service request is sent to the search engine, the search engine will use several matchmaking algorithms to determine the semantical similarity between the service request and the service concepts. The matched service concepts will be ranked and displayed to the service consumer. The service consumer can then choose the appropriate service concepts via the view-based search. Eventually, the service metadata that match the selected service concepts are retrieved from the search engine.

Bianchini et al. [2, 3] proposed an ontology-based hybrid matchmaking approach. First of all, a service can be described by the conjunction of Description-Logic (DL)-based concepts from a service ontology. Thus the DL-based classification is used to precisely establish the kind of matching between a request and an advertised service, by deducing their relationship in the ontology. Then, the similarity between the request and its partially matched services are computed and ranked. Similar to Bianchini et al.'s approach, Chiat et al. [21] worked on making use of DL to match semantic web services with service requests.

Kawamura et al. [7] designed a Semantic Service Matchmaker (SSM) in order to search services in the Universal Description Discovery Integration (UDDI) Business Registry. The services are coded with the Web Service Semantic Profile (WSSP), inspired by the DARPA Agent Markup Language for Services (DAML-S) Service Profile, in order to encode semantic information into the Web Service Description Language (WSDL) of services. Then, a constraint filter is used to determine whether a service request can be subsumed by a service.

A series of researches related to service resource matchmaking in the grid environment, have been undertaken, as a result of the feature that the resources and services in the grid environment are distributed.

Tangmunarunkit et al. [8] proposed such a service matchmaker, which is comprised of ontologies, domain background knowledge and matchmaking rules. There are three categories of ontologies as follows: 1) resource ontologies for describing resources on the Grid; 2) resource request ontologies for describing requests; 3) policy ontologies for describing the authorization and usage policy of resources. In addition, matchmaking rules are designed with the ontologies and the ontology-annotated domain background knowledge, in order to match resource providers with requests. Analogously, Harth et al. [6] designed an ontology-based matchmaker for performing resource selection on the Grid, by using terms defined in ontologies to form the loose coupling between resources and request descriptions. In addition, Vega-Gorgojo et al. [9] made use of ontologies to describe Computer Supported Collaborative Learning (CSCL) tools, with the purpose of enabling matchmaking learning services in a CSCL system. Furthermore, Han and Berry [5] developed a heuristic algorithm enabling agents to find similar neighbors in grids. This is realized by building ontologies for the resources owned by agents and then matching between resource ontologies in order to find similar agents.

The above approaches all focus on the semantic disambiguation of services and requests. However, one challenge emerges here – once a service consumer is unable to provide a semantically clarified service request, and instead sends a heterogeneous request (this sort of phenomenon often occurs in the commercial search engines), how do these approaches address the problem? In fact, the ambiguous service request could reduce the performance of the semantic service matchmakers, on account of low-precise matching between service concepts and service requests.

### 2.2 Service Recommendation Systems

A recommendation/recommender system is a software application that aims to present information items that are likely to be of interest to users, in order to support users in their decision making, when interacting with large-scale information bases [22, 23]. Based on the understanding of users' interest, this system uses various techniques to filter useless information and retain useful information [24]. Recommender techniques have been widely employed in many fields, such as the selling of goods [25], movie and news recommendations etc. [26, 27]. Recently, much work has been done in the field of service recommendation, which is discussed as follows:

Costa et al. [28] proposed an ontology-based, context-aware service recommendation system, which uses ontologies to represent users' profiles and service domain knowledge, and find the similarity between ontology-represented users' profiles and services. Analogously, Shahvalady et al. [29] proposed a user profile ontologies-based recommendation approach in the Web service domain. The major difference between this and Costa et al.'s work is that Shahvalady et al. take into account the factor of quality of service (QoS) for service recommendation.

Kuo et al. [24] proposed a location-based service (LBS) recommendation system for mobile commerce, which recommends location-based services based on user preference. Moreover, the authors divide user preference into short-term preference and long-term preference, and allow adjustments in the former by considering real-time user feedbacks, and adjustments in the latter by using user histories.

Blake and Nowlan [30] proposed a methodology to proactively recommend candidate Web services to service consumers within their daily routine. This is realized by: 1) capturing documents by monitoring operational sessions, file actions and system messages; 2) extracting relevant strings from documents; and 3) determining relevant services from service repositories by matching with extracted strings.

Sellamin et al. [31] proposed a recommendation methodology for Web service discovery in distributed registries. Based on a query and a user profile, the methodology can find an appropriate registry from distributed registries by matching them with the user profile, and run the query over the registry to recommend proper Web services.

Han et al. [32] proposed a recommendation system for service selection in the Cloud computing environment. After a service request is passed to a Web portal, Cloud services with the optimal attributes of QoS and service-rank, integrating quality of virtual hypervisors used by different Cloud service platforms, user feedback and cost for providing better services, are selected from Cloud providers.

There are two limitations in the existing service recommendation systems:

1. Most of the systems focus on using predefined users' profiles to find relevant services. Consequently, the performance of the systems heavily relies on the quality of users' profiles. Nevertheless, for the Service Ecosystem, we are concerned with the issue that users do not have enough knowledge regarding their service requests. This issue normally arises when users first come to a system without profiles or users retrieve new services irrelevant to their profiles. Obviously, the users' profiles-based service recommendation mechanism cannot work in such situations.
2. Most of the systems focus on Web/Cloud service recommendation. As introduced in Chapter 1, services in the Service Ecosystem concern all available service entities in human society, and Web services occupy only a limited proportion of these. In this research, we are more concerned with generic services rather than Web services. Therefore, these systems do not propose any means for generic services recommendation in the Service Ecosystem.

In order to resolve the two issues observed in this survey, we propose an innovative service recommendation system, which predicts semantically similar services based on capturing interactions between users and semantic service matchmakers, in order to enhance the dependability of matchmakers in the Service Ecosystem.

### 2.3 Semantic Similarity Models

The current semantic similarity models are manifold and can be classified into three main categories according to the utilized information – edge (distance)-based models [10, 12, 33-37], node (information content)-based models [15, 38, 39] and hybrid models [13, 40-42]. In the rest of the section, we will briefly introduce the three categories and the typical models in each category, and analyse their limitations.

**Edge (Distance)-based Models**. The edge-based models are based on the shortest path between two nodes in a definitional network that is a type of hierarchical semantic net, in which all nodes are linked with *is-a* relations. The models are based on the assumption that all nodes are evenly distributed and are of similar densities and the distance between any two nodes is equal. They can also be applied to a network structure.

One typical definition of the edge-based model is provided by Rada [10], and is described as:

$$\text{Distance } (c_1, c_2) = \text{Minimum number of edges seperating } c_1 \text{ and } c_2 \tag{1}$$

and the similarity between $c_1$ and $c_2$ is given by

$$sim_{rada}(c_1, c_2) = 2 \times Max \text{ - Distance}(c_1, c_2) \tag{2}$$

where $Max$ is the maximum depth of the definitional network.

**Node (Information Content)-based Models.** Information content-based models are used to judge the semantic similarity between concepts in a definitional network, based on measuring the similarity by taking information content into account. These models can avoid the defect of the edge counting approaches which cannot control variable distances in a dense definitional network [15].

Resnik [15] developed such a model whereby the information shared by two concepts can be indicated by the concept which subsumes the two concepts in the taxonomy. Then, the similarity between the two concepts $c_1$ and $c_2$ can be mathematically expressed as follows:

$$sim_{\mathrm{Re}snik}(c_1, c_2) = \max_{c \in S(c_1, c_2)}[-\log P(c)] \tag{3}$$

where $S(c_1, c_2)$ is the set of concepts that subsume both $c_1$ and $c_2$, and P($c$) is the possibility of encountering an instance of concept $c$.

Lin [38]'s semantic similarity model is the extension of Resnik's model, which measures the similarity between two nodes as the ratio between the amount of commonly shared information of the two nodes and the amount of information of the two nodes, which can be mathematically expressed as follows:

$$sim_{Lin}(c_1, c_2) = \frac{2 \times sim_{\mathrm{Re}snilk}(c_1, c_2)}{IC(c_1) + IC(c_2)} \tag{4}$$

**Hybrid Models**. Hybrid models combine the features of edge-based and node-based models for determining the similarity measure. Jiang et al. [40] developed a hybrid model that uses node-based theory to enhance the edge-based model. Their method takes into account the factors of local density, node depth and link types. The weight between a child concept $c$ and its parent concept $p$ can be measured as:

$$wt(c, p) = (\beta + (1-\beta)\frac{\bar{E}}{E(p)})(\frac{d(p)+1}{d(p)})^{\alpha}(IC(c) - IC(p))T(c, p) \tag{5}$$

where $d(p)$ is the depth of node $p$, $E(p)$ is the number of edges in the child links, $\bar{E}$ is the average density of the whole hierarchy, $T(c, p)$ represents the link type, and $\alpha$ and $\beta$ ($\alpha \geq 0$, $0 \leq \beta \leq 1$) are the control parameters of the effect of node density and node depth towards the weight.

The distance between two concepts is defined as follows:

$$\text{Distance}(c_1, c_2) = \sum_{c \in \{path(c_1,c_2) - LS(c_1,c_2)\}} wt(c, p(c)) \tag{6}$$

where $path(c_1, c_2)$ is the set that contains all the nodes in the shortest path from $c_1$ to $c_2$, and $LS(c_1, c_2)$ is the lowest concept that subsumes both $c_1$ and $c_2$.

In some special cases, such as when only the link type is considered as the factor of weight computing ($\alpha$=0, $\beta$=1, and $T(c, p)$ =1), the distance algorithm can be simplified as follows:

$$\text{Distance}(c_1, c_2) = \text{IC}(c_1) + \text{IC}(c_2) - 2 \times sim(c_1, c_2) \tag{7}$$

where $\text{IC}(c) = \text{-log P}(c)$, and $sim(c_1, c_2) = \max_{c \in LS(c_1,c_2)}[\text{-log P}(c)]$.

Finally, the similarity value between two concepts $c_1$ and $c_2$ is measured by converting the semantic distance as follows:

$$sim_{Jiang}(c_1, c_2) = 1 - \text{Distance}(c_1, c_2) \tag{8}$$

The testing results show that the parameters $\alpha$ and $\beta$ do not heavily influence the similarity measure computation [40].

It can be observed that the above models are all designed for the semantic network environment that is a graphic notation for representing knowledge in patterns of interconnected nodes and arcs [43]. A typical example of a semantic network is WordNet®. However, when applied in the ontology environment, the above models could meet some challenges, which are addressed as follows [44]:

It is important to note that semantic networks are not ontologies. An ontology can define the semantics of objects with datatype properties, object properties and restrictions etc, e.g., Resource Description Framework Schema (RDFS) and Web Ontology Language (OWL). A semantic network cannot represent such rich semantics as can be represented by an ontology. Therefore, the above models from the literature can be difficult to apply in the ontology environment.

These models all ignore the factor of the contents of nodes in the concept similarity measure, due to the feature of nodes in the semantic network environment, in which a node is a single word without adequate properties. Nevertheless, in the ontology environment, ontology concepts are defined with sufficient data-type and object-type properties and restrictions, and the combinations of these properties can be regarded as the crucial identifications for the concepts. Therefore, we can deduce that the concept similarity measure in the ontology environment should emphasize the factor of the content of concepts, and as a result of this being ignored, these concept similarity models may meet challenges.

These models primarily focus on computing the relative positions of nodes in the semantic network environment, especially in definitional networks, in which nodes are only linked by *is-a* relations. Jiang et al.'s model is an exception, but they do not propose any means to obtain the weight of local density, node depth and link types. In the ontology environment, the relations (object properties) between concepts are diverse and customizable. Consequently, the semantic similarity models could find it difficult to cope with the multi-relational ontology environment.

In order to address the crucial issues, we design a semantic similarity model whose purpose is to measure concept similarities in the ontology environment.

## 3 SERVICE CONCEPT RECOMMENDATION SYSTEM

In this paper, we propose a framework of a service concept recommendation system by taking into account the heterogeneous nature of services in the Service Ecosystem. By incorporating the framework into a semantic service matchmaker, the service concept recommendation system can recommend service concepts to service consumers who enter incomplete or incorrect queries to the matchmaker, as a result of a lack of relevant domain knowledge about their service requests, in order to enhance the dependability of the semantic service matchmaker in the Service Ecosystem.

### 3.1 System Architecture and Workflow

In Fig. 1, we present the system architecture and workflow of the proposed service concept recommendation system. The system workflow is comprised of the following four steps:

**Step 1.** Service concept selection. A service consumer selects a service ontology concept from the service ontologies stored in a service knowledgebase service, by a series of interactions with a semantic service matchmaker.

**Step 2.** Service concept obtainment. The service concept recommendation methodology will obtain the selected service concept from the semantic service matchmaker.

**Step 3.** Semantically similar service concept retrieval. The service recommendation system will make use of a semantic similarity model to find the semantically similar concepts from the service ontologies to which the selected concept belongs, based on the concept.

**Step 4.** Service concept recommendation. The semantically similar service concepts will be displayed to the service consumer and ranked according to their similarly values to the selected concept. When the service consumer selects a concept from the ranking list, Step 1 to Step 4 will be repeated until he/she eventually finds a concept that can best fit his/her query intention.

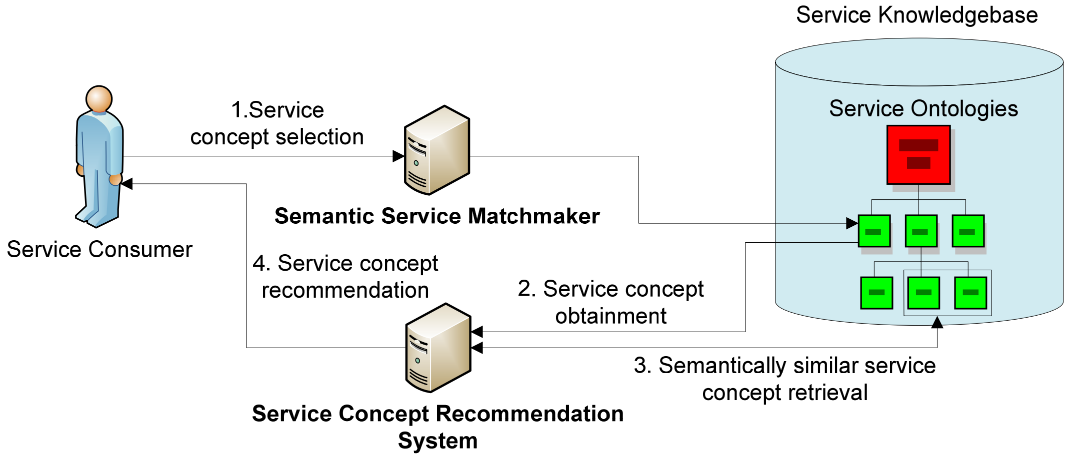


Fig. 1.System architecture and workflow of the service concept recommendation system

### 3.2 Use Scenario

The prototype of the service concept recommendation system has been trialled within the Customized Semantic Service Search Engine (CSSSE) project in a case study which is developing a semantic service search engine for the transport service domain [4]. The following scenario describes a service consumer carries out a service search task, making use of the recommendation system to find the correct concept(s) that can represent his/her query intention.

As shown in Fig .2, the service consumer searches for a service regarding underwater filming (which can be represented by the "Diving Photography" concept in the transport service ontology in CSSSE). Due to a lack of relevant service domain knowledge, s/he enters the query word "filming in water" to the search engine, and the search engine retrieves a "Diving" concept from the service ontology. By observing the "Service Descriptions" property of the "Diving" concept (shown in the upper left side of the screen in Fig. 2), the service consumer finds that the concept cannot be used to completely represent his/her query intention. The proposed concept recommendation system then recommends a list of ranked semantically similar concepts (shown under the label "Relevant Concepts" in the lower left side of the screen in Fig. 2) based on our proposed semantic similarity model. By observing the "Service Descriptions" property of each concept, the service consumer may find the "Diving Photography" concept. Once the "Diving Photography" concept has been clicked, its associated services are retrieved and displayed to the service consumer (shown on the right side of the screen in Fig. 2).

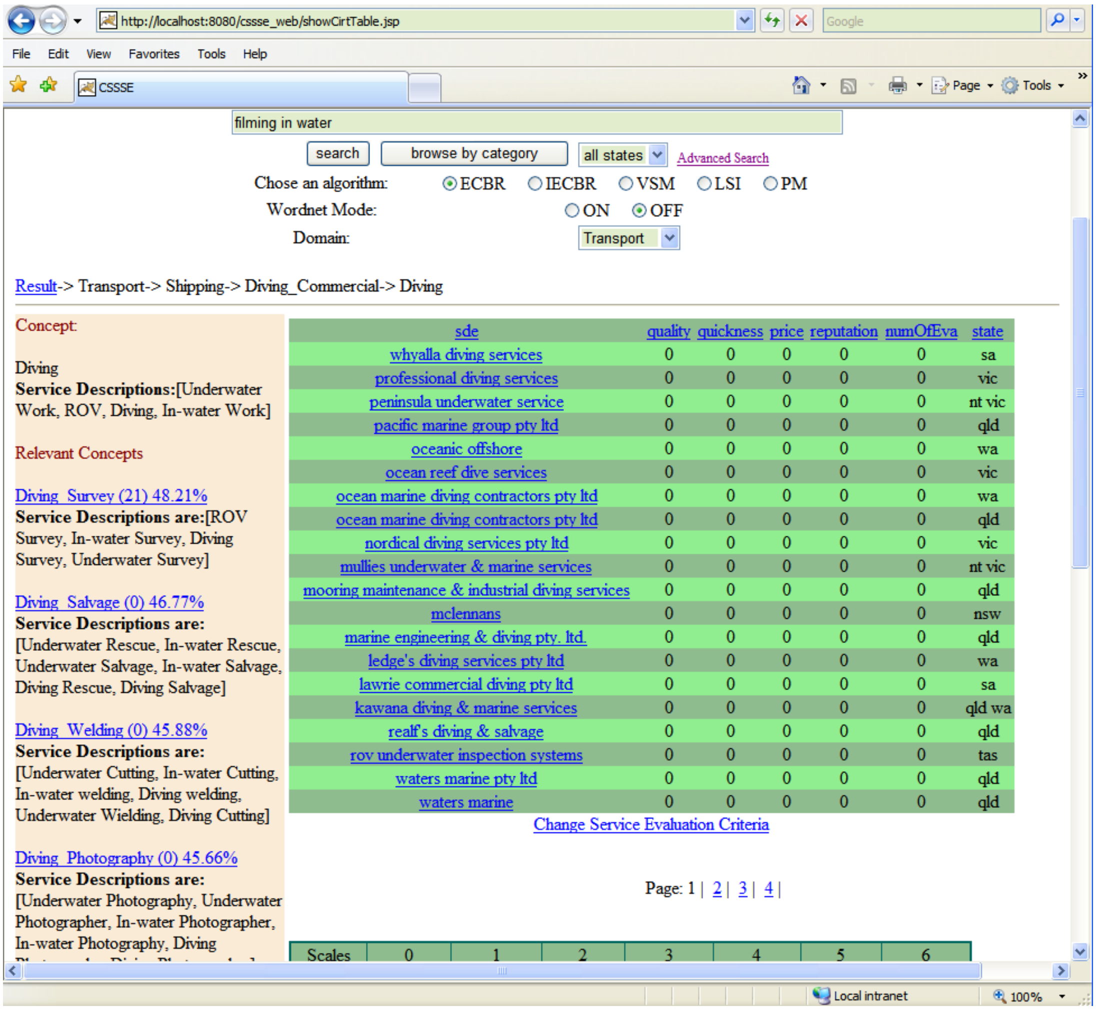

Fig. 2. Screenshot of a use scenario of the service concept recommendation system

## 4 CONVERTING AN ONTOLOGY TO A LIGHTWEIGHT ONTOLOGY

Before we present our semantic similarity model, we need to convert the ontology environment to a so-called lightweight ontology environment in which our model can be applied.

An ontology is defined by Gruber [45] as an explicit specification of conceptualization, which comprises objects (concepts) and relationships among objects (concepts).

We define the lightweight ontology in which the proposed concept similarity algorithm can be employed as follows [44]:

**Definition 1.** *Lightweight ontology*

A lightweight ontology refers to an ontology in which concepts are related only with an *is-a* relation that is a generalization/specification relationship between the upper generic concept and lower specific concept. The lower concept inherits all the properties of the upper concept.

The *is-a* relation commonly appears in semantic web documents (SWD), e.g., it can be represented as </rdfs:subClassOf> in RDFS and OWL. However, it is crucial to note that there are various types of relationships in an SWD. It is a challenge to convert a normal ontology to a lightweight ontology. To overcome this challenge, we present another definition – pseudo-concept $\varsigma$ as follows:

**Definition 2.** *Pseudo-concept* $\varsigma$

We define a pseudo-concept $\varsigma$ for a concept c as a combination of (c, $\delta \rightarrow \gamma_\delta$, $o \rightarrow \gamma_o$, $\lambda_o$), where c is the name (or Uniform Resource Identifier (URI)) of the concept c, $\delta$ is the datatype property(s) of the concept c, $\gamma_\delta$ is the restriction(s) of the datatype property $\delta$, o is the object property(s) of the concept c, $\gamma_o$ is the restriction(s) of the object property o, and $\lambda_o$ is the name(s) of the object concept(s) that o relates to. In addition, $\rightarrow$ refers that the similarity of the restriction(s) is determined by the affiliated property(s). In other words, although two different properties have the same restrictions, the properties are still regarded as different.

The theorems regarding the detailed conversion between a concept and a pseudo-concept can be referenced from our submitted work [46]. Subsequently, we use a simple example in order to illustrate the process of converting an ontology to a lightweight ontology and converting a concept to a pseudo-concept.

Fig. 3 presents an example of ontology – a pizza ontology, which originates from the cognominal ontology given by the Protégé-OWL tutorial [47]. We can see that there are seven concepts involved in this ontology linked by two different types of relations – *is-a* and hasTopping. To convert this pizza ontology to a lightweight ontology, we are required to remove the hasTopping relations and preserve the *is-a* relations.

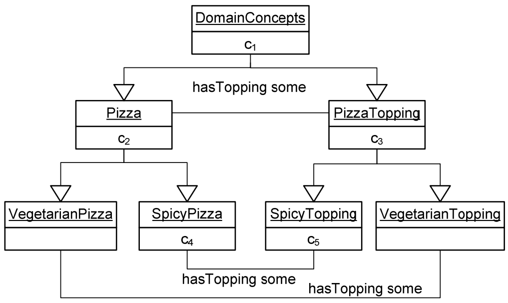


Fig. 3. An example of ontology – a pizza ontology

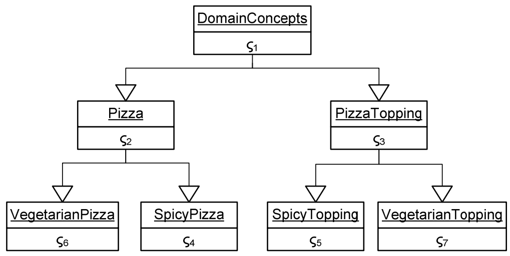


Fig. 4. The lightweight ontology for the pizza ontology

According to the two definitions above, the lightweight ontology can be found in Fig. 4, in which each pseudo-concept is represented below:

$\varsigma_1$= {DomainConcepts}
$\varsigma_2$= {Pizza, hasTopping→some, PizzaTopping}
$\varsigma_3$= {PizzaTopping}
$\varsigma_4$= {SpicyPizza, hasTopping→some, SpicyTopping}
$\varsigma_5$= {SpicyTopping}
$\varsigma_6$= {VegetarianPizza, hasTopping→some, VegetarianTopping}
$\varsigma_7$= {VegetarianTopping}

In terms of the definition of the lightweight ontology and pseudo-concept, an ontology that comprises datatype-property-defined concepts and object-property-defined relationships can be converted to a lightweight ontology that comprises pseudo-concepts linked by *is-a* relationships. There are two advantages of the ontology-lightweight ontology conversion, which can be described as follows:

1. The possibility of measuring concept similarity by measuring similarity of object properties and datatype properties of concepts. As discussed in Section 2.2, the existing semantic similarity models ignore the content-based concept similarity measure. Since concepts encapsulate object properties and datatype properties of into pseudo-concepts, the semantic similarity between two concepts can be partially obtained by measuring the content similarity between their converted pseudo-concepts.
2. The convenience of applying existing semantic similarity models to concept similarity measures in the ontology environment. As discussed in Section 2.2, the existing semantic similarity models focus mostly on definitional networks with hierarchical structures, which cannot be directly used to ontologies with more complicated structures. Since a lightweight ontology is a hierarchical structure linked by single-type relationships, by converting an ontology to a lightweight ontology, the existing semantic similarity models can be directly employed for partially measuring concept similarity in the ontology environment.

## 5 HYBRID SEMANTIC SIMILARITY MODEL

It is well-known that many available IR approaches can be utilized to compute concept similarity based on the content of a pseudo-concept. However, we observe an issue here – although the similarity of two concepts can be measured by contrasting the content of their pseudo-concept, this approach is not sufficient to reveal the extent of their similarity. The reason for this is that an ontology can be represented as a graph in which each concept is a node and relations are arcs among the nodes, and the similarity of two nodes also relates to the structure of the graph and the relative distance between the two nodes [10, 48]. Jiang et al.'s model [17] inspires us to attempt to integrate the factor of content of pseudo-concepts and the factor of structure of lightweight ontology to compute the extent of the similarity between two concepts.

In this section, we present a hybrid semantic similarity model integrating the two factors above. Our proposed hybrid model involves two sub-models. The first sub-model measures concept similarity based on the content of pseudo-concept, by means of the cosine correlation approach. The second sub-model measures concept similarity based on the structure of a lightweight ontology graph, by means of an approach originating from the enhanced topic-based vector model (eTVSM) [48]. The product of the two sub-models is two concept-concept matrixes. Then we integrate the two matrixes to obtain a new concept-concept matrix that indicates the extent of similarity between concepts. To illustrate the working mechanism of the hybrid model, we will compute the concept similarity values for the pizza ontology in Fig.2.

### 5.1 Content-based Semantic Similarity Model

As described earlier, a pseudo-concept can be regarded as a textual description of a concept. In this section, we propose to make use of the cosine correlation model to compute the extent of similarity between each pair of concepts of an ontology based on the extent of their pseudo-concepts.

The main reason for applying the cosine correlation model is to construct a concept-concept matrix for an ontology in which each element is the similarity value between the two corresponding concepts. To construct the concept-concept matrix, first of all we need to build an $m \times n$ matrix $A$. Each of the $m$ unique terms in the collection is assigned a row in the matrix, while each of the $n$ concepts in the collection is assigned a column in the matrix. An element $a_{ij}$ where $A= [a_{ij}]$ indicates the weight between term $t_i$ and concept $c_j$. The weight can be obtained by the term frequency-inverse document frequency (tf-idf) method [49], which can be represented mathematically as follows:

$$a_{ij} = \frac{freq_{i,j}}{\max_l freq_{l,j}} \bullet \log \frac{N}{n_i} \tag{9}$$

where $freq_{i,j}$ is frequency of term $t_i$ appearing in a pseudo-concept $\varsigma_j$, $max_l freq_{l,j}$ is the total number of terms appearing in a pseudo-concept $\varsigma_j$, $N$ is the total number of pseudo-concepts and $n_i$ is the number of pseudo-concepts where term $t_i$ appears.

Therefore, the elements of a column $a_{ij}$ ($i =1...m$) of the matrix $A$ are the weights between all terms and a concept $c_j$. Then we normalize each column of the matrix $A$ to 1 as shown below:

$$\left| \sum_{i=1}^{m} a_{ij} \right| = 1 \tag{10}$$

Once the normalized matrix $A$ is acquired, we can obtain a concept-concept matrix C through the product of the transpose of the matrix $A$ and itself as shown below:

$$C = A^T A \tag{11}$$

Here we explain the purpose of normalizing each column of matrix $A$. According to Equation (11), each element of the matrix $C$ can be obtained as shown below in Equation (12), which is the scalar product of two columns. It is educed that the range of $c_{kl}$ is between 0 and 1 and the maximum of $c_{kl}$ is the product of two same columns, which is 1 according to Equation (12).

$$c_{kl} = \sum_{i=1}^{m} a_{ki} \times a_{il} \tag{12}$$

Obviously, matrix $A$ can be regarded as the assembly of similarity values between arbitrary couples of concepts based on the content of their pseudo-concepts, which can be represented by Equation (13). Here, each column of matrix $C$ can be viewed as the vector of a concept, and the similarity value of two concepts is the scalar product of corresponding concept vectors. Because concept vectors are normalized, the scalar product is equal to the cosine of the angle between the two concept vectors.

$$C = [c_{kl}] = [\cos(\vec{c}_k, \vec{c}_l)] = [sim_c(c_k, c_l)] \tag{13}$$

Finally, all pair-wise concept similarity values from the ontology example presented in Section 3 are given in Table 1.

TABLE 1
PSEUDO-CONCEPT CONTENT-BASED CONCEPT SIMILARITY VALUES FOR THE PIZZA ONTOLOGY

| | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ | $c_7$ |
|---|---|---|---|---|---|---|---|
| $c_1$ | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $c_2$ | 0 | 1 | 0.5 | 0.11 | 0 | 0.11 | 0 |
| $c_3$ | 0 | 0.5 | 1 | 0 | 0 | 0 | 0 |
| $c_4$ | 0 | 0.11 | 0 | 1 | 0.5 | 0.11 | 0 |
| $c_5$ | 0 | 0 | 0 | 0.5 | 1 | 0 | 0 |
| $c_6$ | 0 | 0.11 | 0 | 0.11 | 0 | 1 | 0.5 |
| $c_7$ | 0 | 0 | 0 | 0 | 0 | 0.5 | 1 |

### 5.2 Structure-based Semantic Similarity Model

As mentioned earlier, the structure-based approach originates from the topic similarity measure model for the topic map environment [48]. In our model, we employ this method in the environment of lightweight ontologies. As lightweight ontologies have only one type of relationship, the weights of relations can be viewed as equal and the issue of relation weights can be ignored in the measurement process. The process of computing concept similarity extent can be divided into two stages: 1) determining the pseudo-concept vectors based on a lightweight ontology structure; 2) obtaining a concept similarity matrix by means of the scalar product of the pseudo-concept vectors. The operational vector space dimensionality is specified by the number of pseudo-concepts in a lightweight ontology.

Let m be the number of pseudo-concepts in a lightweight ontology, and a set of pseudo-concepts can be represented as $\Theta = \{\varsigma_1 \ldots \varsigma_m\}$. In order to represent the lightweight ontology structure, we can use $G(\varsigma_i)$ to represent the generic concept of a pseudo-concept $\varsigma_i$ in an is-a relation. Returning to the lightweight ontology example from Fig. 2, the structure of the lightweight ontology can be represented as follows:

$G(\varsigma_1) = \{\}$
$G(\varsigma_2) = \{\varsigma_1\}$
$G(\varsigma_3) = \{\varsigma_1\}$
$G(\varsigma_4) = \{\varsigma_2\}$
$G(\varsigma_5) = \{\varsigma_3\}$
$G(\varsigma_6) = \{\varsigma_2\}$
$G(\varsigma_7) = \{\varsigma_3\}$

Since it is well-known that the *is-a* relation is transitive in ontologies, here we use $G^*(\varsigma_i)$ to represent all the generic concepts of a pseudo-concept $\varsigma_i$. Again, returning to the lightweight ontology example from Fig. 2, the structure of lightweight ontology can be represented as follows:

$G^*(\varsigma_1) = \{\}$
$G^*(\varsigma_2) = \{\varsigma_1\}$
$G^*(\varsigma_3) = \{\varsigma_1\}$
$G^*(\varsigma_4) = \{\varsigma_1, \varsigma_2\}$
$G^*(\varsigma_5) = \{\varsigma_1, \varsigma_3\}$
$G^*(\varsigma_6) = \{\varsigma_1, \varsigma_2\}$
$G^*(\varsigma_7) = \{\varsigma_1, \varsigma_3\}$

We use $\Theta_s$ to represent the set of specific concepts that are not generic concepts of any pseudo-concepts in a lightweight ontology. In our lightweight ontology example, the specific concept set comprises:

$\Theta_s = \{\varsigma_4, \varsigma_5, \varsigma_6, \varsigma_7\}$

On the other hand, the complement of $\Theta_s$ is the set of all generic concepts, which can be represented as $\Theta_g$. The generic concept set in our lightweight ontology example consists of:

$\Theta_g = \{\varsigma_1, \varsigma_2, \varsigma_3\}$

As mentioned previously, each pseudo-concept is assigned a vector with the dimensions towards all pseudo-concepts in a lightweight ontology. The approach of obtaining vectors can be divided into two steps: 1) obtaining vectors for the specific pseudo-concept set; 2) obtaining vectors for the generic pseudo-concept set.

First, we employ Equation (14) to obtain vectors for the specific pseudo-concept set. We assign the same weight to the dimensions of a vector $\vec{\varsigma}_i$ that have counterparts in its generic concept set $G^*(\varsigma i)$ and itself. The heuristics behind this can be found in [48].

$$\forall \varsigma_i \in \Theta_s : \vec{\varsigma}_i = (\varsigma_{i,1} \dots \varsigma_{i,m})$$

$$\text{with } \varsigma_{i,k} = \begin{cases} 1 & \text{if } \varsigma_k \in G^*(\varsigma_i) \vee i = k \\ 0 & \text{else} \end{cases} \tag{14}$$

Once a specific pseudo-concept vector has been obtained, we normalize the vector length to 1 by making use of the Equation (15), in order to make the weight of each pseudo-concept in the vector dependent on the number of generic concepts. The normalization also benefits the angle measure between two vectors.

$$\forall \varsigma_i \in \Theta_s : |\vec{\varsigma}_i| = 1 \tag{15}$$

Second, the generic pseudo-concept vector can be obtained by the sum of all its specific concepts related by the direct *is-a* relations as shown in Equation (16). Similar to Equation (15), the length of generic pseudo-concept vector needs to be normalized to 1 as shown in Equation (17). This is a recursive process in which the lower level generic pseudo-concept vectors are obtained first by obtaining all their specific concepts from $\Theta_s$ and then by normalizing the sum. Subsequently, the upper level generic pseudo-concept vectors are obtained by gaining all its specific concepts from the lower level generic pseudo-concept set and $\Theta_s$ and by normalizing the sum.

$$\forall \varsigma_i \in \Theta_g : \vec{\varsigma}_i = \sum_{\varsigma_k \in \Theta : \varsigma_i \in G(\varsigma_k)} \vec{\varsigma}_k \tag{16}$$

$$\forall \varsigma_i \in \Theta_g : |\vec{\varsigma}_i| = 1 \tag{17}$$

Once we have all the pseudo-concept vectors, we can obtain the pseudo-concept similarity matrix *L* by the scalar product of arbitrary pairs of vectors. Since one pseudo-concept corresponds to one concept, matrix *L* is also the assembly of similarity values between all corresponding pairs of concepts from an ontology. Meanwhile, the similarity value between two concepts can also be viewed as the cosine of the angle between the two corresponding pseudo-code vectors. The extent of similarity between two concepts can be obtained by using Equation (19) shown below:

$$sim_s(c_i, c_j) = L_{i,j} = \cos(\vec{\varsigma}_i, \vec{\varsigma}_j) = \vec{\varsigma}_i \vec{\varsigma}_j = \sum_{k=1}^{m} \varsigma_{i,k} \varsigma_{j,k} \tag{18}$$

Finally, the pair-wise concept similarity values from the pizza ontology in Fig. 2 are given in Table 2.

TABLE 2
LIGHTWEIGHT ONTOLOGY STRUCTURE-BASED CONCEPT SIMILARITY VALUES FOR THE PIZZA ONTOLOGY

| | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ | $c_7$ |
|---|---|---|---|---|---|---|---|
| $c_1$ | 1 | 0.83 | 0.83 | 0.76 | 0.76 | 0.76 | 0.76 |
| $c_2$ | 0.83 | 1 | 0.4 | 0.91 | 0.36 | 0.91 | 0.36 |
| $c_3$ | 0.83 | 0.4 | 1 | 0.36 | 0.91 | 0.36 | 0.91 |
| $c_4$ | 0.76 | 0.91 | 0.36 | 1 | 0.33 | 0.66 | 0.33 |
| $c_5$ | 0.76 | 0.36 | 0.91 | 0.33 | 1 | 0.33 | 0.66 |
| $c_6$ | 0.76 | 0.91 | 0.36 | 0.66 | 0.33 | 1 | 0.33 |
| $c_7$ | 0.76 | 0.36 | 0.91 | 0.33 | 0.66 | 0.33 | 1 |

### 5.3 Integrating the Products of the Two Models

Section 5.1 and Section 5.2 present two semantic similarity matrixes (*C* and *L*) based on the pseudo-concept content and lightweight ontology structure respectively. In this section, we leverage both of these matrixes in order to obtain a new matrix that is able to indicate the similarity among concepts in a more precise manner. We define a matrix *S* in which each element is the weighted arithmetic mean between counterparts in matrices *C* and *L* as shown in Equation (19). According to Equation (13) and Equation (18), the similarity value between two concepts is also the weighted arithmetic mean between the content-based semantic similarity value ($sim_c(c_i, c_j)$) and the structure-based semantic similarity value ($sim_s(c_i, c_j)$).

$$S_{i,j} = sim(c_i, c_j) = (1-\beta)sim_c(c_i, c_j) + \beta sim_s(c_i, c_j) = (1-\beta)C_{i,j} + \beta L_{i,j} (0 \le \beta \le 1) \tag{19}$$

It is noted that an appropriate $\beta$ needs to be determined in order to obtain the optimal performance of this model, which is introduced in the next section.

## 6 EVALUATION

In this section, in order to evaluate our hybrid semantic similarity model, we compare it with four typical semantic similarity model introduced in Section 2.2, including Rada's model, Resnik's model, Lin's model and Jiang's model in a large-scale service ontology. This comparison includes two sub-tasks as follows:

1. Obtaining the best performance from each model. In the IR field, when a query is sent to a search system, a list of results with similarity values is returned from the system. Then the search system needs to decide an optimal threshold value which is used to filter the irrelevant results with lower similarity values, in order to obtain the best performance. Analogously, in our subsequent experiments, as a result of that, the performance of each model was different with different threshold values. Hence, we need to find the optimal threshold value for each model where each model can achieve the best performance. Consequently, for each model, we decide to start the initial threshold value from 0, and to increase this by 0.05 each time until it reaches 0.95; we then obtain the performance data for each model at each time of the variation of the threshold value. Since the intervals of Rada's model and Resnik's model are between 0 and infinite, we use their normalized modules in this experiment. Similarly, with the purpose of obtaining the optimal $\beta$ value, we test the performance of our model from $\beta$=0 to $\beta$=1, with an increment of 0.1 at each time.
2. Comparing the best performance of each model. We then compare the performances of the five models at their optimal threshold values based on the performance indicators to be introduced in Section 6.1.

### 6.1 Performance Indicators

In order to horizontally compare our proposed hybrid model with these models, we utilize the four most widely used performance indicators from the traditional IR field as the evaluation metrics. The performance indicators are defined as follows:

**Precision.** Precision in the information retrieval is used to measure the preciseness of a search system [50]. Precision for a single concept refers to the proportion of matched and logically similar concept in all concepts matched to this concept, which can be represented by Equation (20) below:

$$\text{Precision(S)} = \frac{\text{Number of matched and logically similar concepts}}{\text{Number of matched concepts}} \tag{20}$$

With regard to the whole collection of concepts in an ontology, the total precision is the sum of precision for each concept normalized by the number of concepts in the collection, which can be represented by Equation (21) below:

$$\text{Precision(T)} = \frac{\sum_{i=1}^{n} \text{Precision}(\text{S}_i)}{n} \tag{21}$$

**Mean average precision.** Before we introduce the definition of mean average precision, the concept of average precision should be defined. Average precision for a single concept is the average of precision values after truncating a ranked concept list matched by this concept after each of the logically similar concept for this concept. This indicator emphasizes the return of more logically similar concepts earlier, which can be represented as:

$$\text{Average precision(S)} = \frac{\text{Sum(Precision @ Each logically similar concept in a list)}}{\text{Number of matched and logically similar concepts in a list}} \tag{22}$$

Mean average precision refers to the average of the average precision values for the collection of concepts in an ontology, which can be represented as:

$$\text{Mean average precision} = \frac{\sum_{i=1}^{n} \text{Average precision}(S_i)}{n} \tag{23}$$

**Recall.** Recall in the information retrieval is used to measure the effectiveness of a search system [50]. Recall for a single concept is the proportion of matched and logically similar concepts in all concepts that are logically similar to this concept, which can be represented by Equation (24) below:

$$\text{Recall(S)} = \frac{\text{Number of matched and logically similar concepts}}{\text{Number of logically similar concepts}} \tag{24}$$

It is noted that the number of logically similar concepts can be determined only by a peer-reviewed approach in the current situation. With regard to the whole collection of concepts in an ontology, the total recall is the sum of recall for each concept normalized by the number of concepts in the collection, which can be represented by Equation (25) below:

$$\text{Recall(T)} = \frac{\sum_{i=1}^{n} \text{Recall}(S_i)}{n} \tag{25}$$

**Harmonic Mean.** Harmonic Mean in the information retrieval is used as an aggregated performance scale for the search engine [50]. In this experiment, Harmonic Mean is the average of precision and recall, which can be represented below as:

$$\text{Harmonic Mean} = \frac{1}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} \tag{26}$$

When the Harmonic Mean value reaches the highest, it means the integrated value between precision and recall reaches the highest at the same time.

**F-measure.** F-measure is another measure that combines precision and recall, and the difference is that users can specify the preference on recall or precision by configuring different weights [50]. In this experiment, we employ F-measure ($\beta$=2) that weights recall twice as much as precision, which can be represented as:

$$\text{F-measure } (\beta=2) = \frac{(1+\beta^2)\cdot \text{Precision} \times \text{Recall}}{\beta^2 \cdot \text{Precision+Recall}} = \frac{5 \times \text{Precision} \times \text{Recall}}{4 \times \text{Precision+Recall}} \tag{27}$$

All of the above indicators have the same limitation – they do not consider the number of non-logically similar concepts in a matched concept collection of a concept. Furthermore, if there is no logically similar concept in the matched collection, recall cannot be defined. To resolve this issue, we need another performance indicator – **Fallout**. In this experiment, fallout for a single concept is the proportion of non-logically similar concepts matched by this concept in the whole collection of non-logically similar metadata for this concept, which can be represented as:

$$\text{Fallout(S)} = \frac{\text{Number of matched and non-logically similar concept}}{\text{Number of non-logically similar concept}} \tag{28}$$

With regard to the whole collection of concepts, the total fallout value is the sum of the fallout value for each concept normalized by the number of concepts in an ontology, which can be represented as:

$$\text{Fallout(T)}=\frac{\sum_{i=1}^{n}\text{Fallout}(S_i)}{n} \tag{29}$$

In contrast to other performance indicators, the lower the fallout value, the better is the search performance.

## 6.2 Optimal Performance Obtainment

In order to test the performance of the existing models and our proposed hybrid model in a genuine service ontology environment, and to obtain the most accurate testing result, we design a large-scale ontology – a transport service ontology, by means of surveying more than 1000 Australian transport service companies' websites and referencing transport service domain knowledge from the Wikipedia (http://www.wikipedia.org/) and Open Directory Project (http://www.dmoz.org/). The transport service ontology is a four-layer ontology with 304 ontology concepts, covering nearly all existing service information in the transport domain. Each concept is defined by the domain-specific concept descriptions, which corresponds to an actual transport service. Fig..5 presents the abbreviated view of the transport service ontology. The in-depth information with regard to the transport service ontology can be referenced from [51, 52].

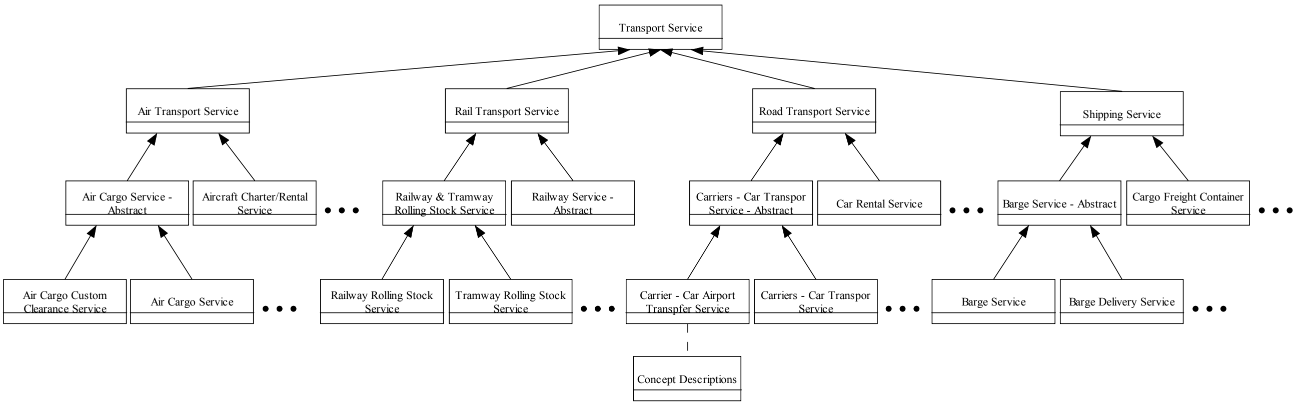


Fig. 5. Abbreviated view of the transport service ontology

Since the F-measure and F-measure ($\beta$=2) are two aggregated metrics, we decide to use them as the primary benchmarks for seeking the optimal threshold value. Fig. 6 and Fig. 7 respectively show the variation of Harmonic Mean values and the variation of F-measure ($\beta$=2) values of the five models on different threshold values plus our model on different $\beta$ values. Based on the two benchmarks, we choose two groups of optimal threshold values for each model, which can be found in the next section.

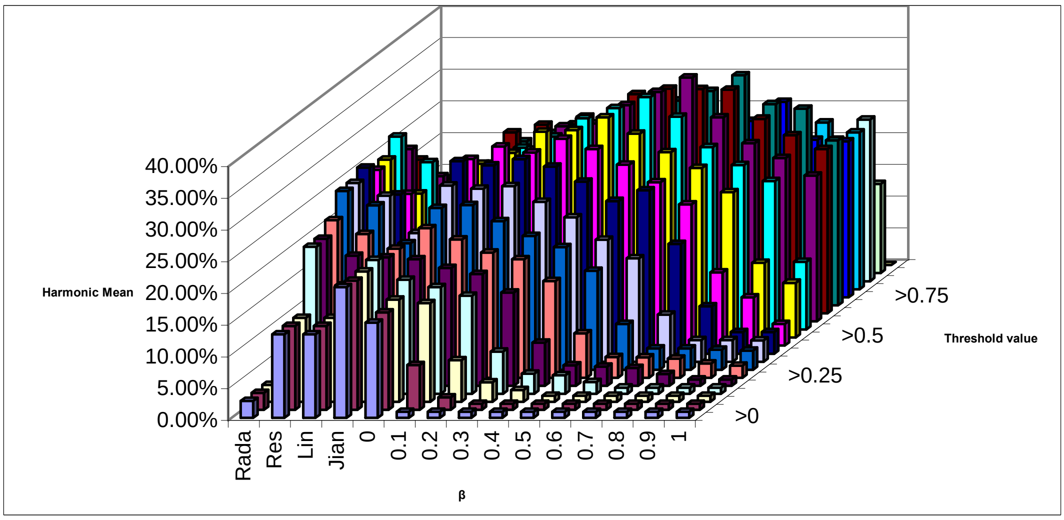


Fig. 6. Harmonic Mean values of Rada's model, Resnik's model, Lin's model, Jiang's model and the hybrid model on different threshold values

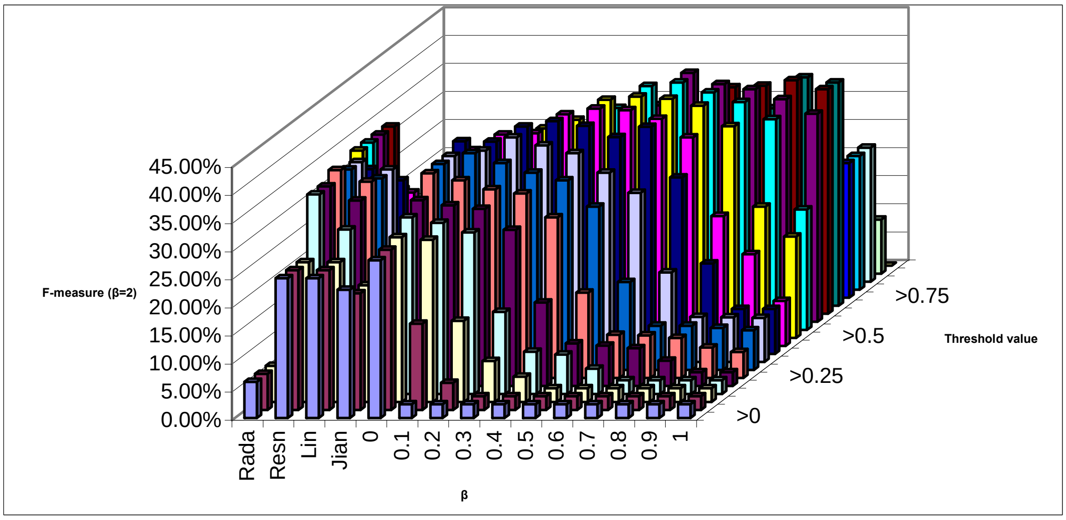


Fig. 7. F-measure (β=2) values of Rada's model, Resnik's model, Lin's model, Jiang's model and the hybrid model on different threshold values

### 6.3 Optimal Performance Comparison

Table 3 and Table 4 present the performance of the five models on the performance indicators based on their optimal threshold values respectively for their highest Harmonic Mean values and F-measure (β =2) values.

First of all, for Table 3, it can be observed that our model obtains the highest Recall and Harmonic Mean values, ranks second for Precision, and ranks third for the rest of the performance indicators, which indicates the outstanding comprehensive performance of the model. For the key factor – Harmonic Mean, the performance data of our model is nearly 50% higher than the second. Therefore, it can be deduced that the best performance of our model is better than the four candidate models in terms of the Harmonic Mean-based optimal threshold values.

Since a survey result indicates that search engine users are more concerned with Recall than Precision in the real environment, as a result of users' purposes in obtaining information [53], we employ the indicator of F-measure (β=2) which weights Recall as twice that of Precision. For Table 4, our model performs the best among the five models on the indicators of Precision, Recall and F-measure (β=2) and ranks second for Mean Average Precision and Fallout. Especially for the key metrics – F-measure (β=2), our model is nearly 50% higher than the four existing models.

To conclude this evaluation, we evaluate our proposed model by comparing its best performance with the four existing semantic similarity models and the results of this comparison show that our model performs better than these models on the key indicators. Thus, we can conclude that this experiment preliminarily proves the advantages of our model in the ontology environment.

The reason that the statistical data are relatively low for these models is that we determine the answer set for each concept based on human judgment. For a large number of concepts within the health service ontology, the answer sets are empty because they are unique; therefore, there are no logically similar concepts for them. These concepts lower the average performance of these models.

TABLE 3
PERFORMANCE OF THE FIVE MODELS ON THEIR OPTIMAL THRESHOLD VALUES FOR HARMONIC MEAN

| Models | Optimal threshold values | Precision | Mean Average Precision | Recall | Fallout | Harmonic Mean |
|---|---|---|---|---|---|---|
| Our model(β=0.3) | >0.3 | 36.36% | 69.33% | 59.26% | 1.87% | 45.07% |
| Rada's model | >0.5 | 13.57% | 44.00% | 52.41% | 11.89% | 21.55% |
| Resnik's model | >0.55 | 47.72% | 74.22% | 22.32% | 1.01% | 30.42% |
| Lin's model | >0.35 | 18.79% | 61.55% | 43.13% | 5.86% | 26.17% |
| Jiang's model | >0.15 | 22.97% | 90.68% | 19.55% | 0.80% | 21.12% |

TABLE 4
PERFORMANCE OF THE FIVE MODELS ON THEIR OPTIMAL THRESHOLD VALUES FOR F-MEASURE (Β =2)

| Models | Optimal threshold values | Precision | Mean Average Precision | Recall | Fallout | F-measure (β =2) |
|---|---|---|---|---|---|---|
| Our model(β=0.3) | >0.1 | 27.14% | 66.82% | 74.78% | 4.09% | 55.35% |
| Rada's model | >0.5 | 13.57% | 44.00% | 52.41% | 11.89% | 33.33% |
| Resnik's model | >0.25 | 16.06% | 45.92% | 54.82% | 13.06% | 36.97% |
| Lin's model | >0.25 | 14.31% | 47.30% | 54.58% | 12.46% | 34.92% |

| | | | | | | |
|---|---|---|---|---|---|---|
| Jiang’s model | >0 | 17.81% | 82.67% | 24.52% | 1.69% | 22.80% |

## 7 CONCLUSION AND FUTURE WORK

In this paper, we proposed a framework for a service concept recommendation system for service retrieval in the Service Ecosystem, which contains an innovative semantic similarity model for querying semantic similar service concepts based on user-selected service concepts in interactions with the semantic service matchmakers. The three major contributions of this research can be described as follows:

1. The proposed framework is able to recommend useful service concepts when semantic service matchmakers fail to find proper service concept(s) to represent service consumers’ service requests. This approach solves the problem that current semantic service matchmakers rely heavily on the quality of service queries and cannot work when a service consumer is unable to provide a semantically clarified service request, which may improve the dependability of semantic service matchmakers.
2. The proposed framework is able to retrieve semantically similar generic service concepts from specific Service Ecosystem domains by capturing interactions between service consumers and semantic service matchmakers, instead of using users’ profiles. This solves the problems of existing service recommendation systems that service consumers without profiles or service consumers who want to retrieve new items excluded in their profiles cannot have appropriate services recommended to them, and most of the service recommendation approaches focus on Web services and ignore generic services in the Service Ecosystem.
3. The designed semantic similarity model is able to compute the similarity between concepts in the ontology environment, by considering the two factors of content of concepts and structure of ontologies. This solves the problem of existing semantic similarity models whereby most of the models take into account the semantic network environment, especially the definitional network environment for similarity computation, and ignore the factor of semantic-rich content of concepts in the ontology environment. Furthermore, by means of an experiment in a large-scale ontology, the proposed semantic similarity model performs more convincingly than do the four candidate models chosen from existing semantic similarity models.

For our future work, firstly we will enhance our semantic similarity model in order to improve its overall performance; secondly, we plan to implement our semantic similarity model in other large-scale knowledge bases; thirdly, we will consider incorporating HCIs into the service concept recommendation system in order to improve its efficiency.

## ACKNOWLEDGMENT

We would like to thank all relevant DEBII staff for their assistance, especially our programmer – Wei Liu who took on the responsibility of implementing the prototype of the service concept recommender system. We also would like to express our gratitude to all the anonymous reviewers of our manuscript for their valuable suggestions for improving this paper.

## REFERENCES


[1] R. Veryard, *Plug and Play: Towards the Component-Based Business*. London: Springer, 2000.

[2] D. Bianchini, V. D. Antonellis, and M. Melchiori, "Flexible semantic-Based service matchmaking and discovery," *World Wide Web,* vol. 11, pp. 227-251, 2008.

[3] D. Bianchini, V. D. Antonellis, B. Pernici, and P. Plebani, "Ontology-based methodology for e-service discovery," *Information Systems* vol. 31, pp. 361 - 380, 2006.

[4] H. Dong, F. K. Hussain, and E. Chang, "A service search engine for the industrial digital ecosystems," *IEEE Transactions on Industrial Electronics,* Accepted.

[5] L. Han and D. Berry, "Semantic-supported and agent-based decentralized grid resource discovery," *Future Generation Computer Systems,* vol. 24, pp. 806-812, 2008.

[6] A. Harth, Y. Hey, H. Tangmunarunkity, and S. Deckeryz, "A semantic matchmaker service on the grid," in *WWW 2004*, New York, 2004, pp. 326-327.

[7] T. Kawamura, J.-A. D. Blasio, T. Hasegawa, M. Paolucci, and K. Sycara, "Preliminary report of public experiment of semantic service matchmaker with UDDI business registry " in *ICSOC 2003*, M. E. Orlowska, S. Weerawarana, M. P. Papazoglou, and J. Yang, Eds. Berlin: Springer-Verlag, 2003, pp. 208-224.

[8] H. Tangmunarunkit, S. Decker, and C. Kesselman, "Ontology-based resource matching in the Grid -the Grid meets the Semantic Web " in *The Semantic Web - ISWC 2003*, D. Fensel, K. Sycara-Cyranski, J. Mylopoulos, and K. Sycara, Eds. Berlin: Springer Verlag, 2003.

[9] G. Vega-Gorgojo, M. L. Bote-Lorenzo, E. Gómez-Sánchez, Y. A. Dimitriadis, and J. I. Asensio-Pérez, "A semantic approach to discovering learning services in grid-based collaborative systems," *Future Generation Computer Systems,* vol. 22, pp. 709-719, 2006.

[10] R. Rada, H. Mili, E. Bicknell, and M. Blettner, "Development and Application of a Metric on Semantic Nets," *IEEE Transactions on Systems, Man and Cybernetics,* vol. 19, pp. 17-30, 1989.

[11] R. K. Srihari, Z. F. Zhang, and A. B. Rao, "Intelligent indexing and semantic retrieval of multimodal documents," *Information Retrieval,* vol. 2, pp. 245-275, 2000.

[12] M. Sussna, "Word Sense Disambiguation for Free-text Indexing Using a Massive Semantic Network," in *The Second International Conference on Information and Knowledge Management (CIKM ’93)*, Washington, 1993, pp. 67-74.

[13] Y. Li, Z. A. Bandar, and D. McLean, "An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources," *IEEE Transactions on Knowledge and Data Engineering,* vol. 15, pp. 871-882, 2003.

[14] D. Lin, "Automatic retrieval and clustering of similar words," in *the 17th COLING*, Austin, 1998, pp. 768-774.

[15] P. Resnik, "Semantic similarity in a taxonomy: an information-based measure and its application to problems of ambiguity in natural language," *Journal of Artificial Intelligence Research,* vol. 11, pp. 95-130, 1999.

[16] R. Rosenfield, "A maximum entropy approach to adaptive statistical modelling," *Computer Speech and Language,* vol. 10, pp. 187-228, 1996.

[17] O. Steichen, C. D. Bozec, M. Thieu, E. Zapletal, and M. C. Jaulent, "Computation of semantic similarity within an ontology of breast pathology to assist inter-observer consensus," *Computers in Biology and Medicine,* vol. 36, pp. 768–788, 2006.

[18] R. M. Othman, S. Deris, and R. M. Illias, "A genetic similarity algorithm for searching the Gene Ontology terms and annotating anonymous protein sequences," *Journal of Biomedical Informatics,* vol. 41, pp. 65-81, 2008.

[19] T. Pedersen, S. V. S. Pakhomov, S. Patwardhan, and C. G. Chute, "Measures of semantic similarity and relatedness in the biomedical domain," *Journal of Biomedical Informatics,* vol. 40, pp. 288-299, 2006.

[20] J. L. Sevilla, V. c. Segura, A. Podhorski, E. Guruceaga, Jose´ M. Mato, L. A. Martı´nez-Cruz, F. J. Corrales, and A. Rubio, "Correlation between Gene Expression and GO Semantic Similarity," *IEEE/ACM Transaction on Computational Biology and Bioinformatics,* vol. 2, pp. 330-338, 2005.

[21] L. C. Chiat, L. Huang, and J. Xie, "Matchmaking for Semantic Web services," in *2004 IEEE International Conference on Services Computing (SCC'04)*, Shanghai, 2004, pp. 455-458.

[22] N. Leavitt, "Recommendation technology: Will it boost E-Commerce," *Computer,* vol. 39, pp. 13-16, 2006.

[23] P. Resnick and H. R. Varian, "Recommender systems," *Communications of the ACM,* vol. 40, pp. 56-58, 1997.

[24] M.-H. Kuo, L.-C. Chen, and C.-W. Liang, "Building and evaluating a location-based service recommendation system with a preference adjustment mechanism," *Expert systems with applications,* vol. 36, pp. 3543-3554, 2009.

[25] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Computing,* vol. 7, pp. 76-80, 2003.

[26] D. Billsus, C. A. Brunk, C. Evans, B. Gladish, and M. Pazzani, "Adaptive interfaces for ubiquitous Web access," *Communication of ACM,* vol. 45, pp. 34-38, 2002.

[27] B. N. Miller, I. Albert, S. K. Lam, J. A. Konstan, and J. Riedl, "MovieLens unplugged: Experiences with an occasionally connected recommender system," in *the 8th International Conference on Intelligent User Interfaces* Miami, Florida, USA 2003, pp. 263 - 266.

[28] A. C. M. Costa, R. S. S. Guizzardi, G. Guizzardi, and J. G. P. Filho, "COReS: Context-aware, Ontology-based Recommender system for Service recommendation," in *The 19th International Conference on Advanced Information Systems Engineering (CAiSE'07)*, Trondheim, Norway, 2007.

[29] S. H. Shahvalady, M. Mohsenzadeh, A. G. Neiat, and F. Mahmoodi, "A web service recommender system using user ontology," in *Seventh International Conference on Computer Science and Information Technologies (CSIT 2009)*, Yerevan, Armenia, 2009, pp. 234-238.

[30] M. B. Blake and M. F. Nowlan, "A web service recommender system using enhanced syntactical matching," in *2007 IEEE International Conference on Web Services (ICWS 2007)*, Salt Lake City, Utah, USA, 2007, pp. 575-582.

[31] M. Sellami, S. Tata, Z. Maamar, and B. Defude, "A recommender system for web services discovery in a distributed registry environment," in *2009 Fourth International Conference on Internet and Web Applications and Services (ICIW 2009)*, Venice/Mestre, Italy, 2009, pp. 418-423.

[32] S.-M. Han, M. M. Hassan, C.-W. Yoon, and E.-N. Huh, "Efficient service recommendation system for cloud computing market," in *the 2nd International Conference on Interaction Sciences (ICIS 2009)*, Seoul, Korea, 2009, pp. 839-845.

[33] G. Hirst and D. St-Onge, "Lexical chains as representations of context for the detection and correction of malapropisms," in *WordNet: An Electronic Lexical Database,*, C. Fellbaum, Ed. Cambridge: The MIT Press, 1998.

[34] C. Leacock and M. Chodorow, "Combining local context and WordNet similarity for word sense identification," in *WordNet: An Electronic Lexical Database* Cambridge: MIT Press, 1998, pp. 265-83.

[35] J. Lee, M. Kim, and Y. Lee, "Information retrieval basedon conceptual distance in IS-A hierarchies," *Journal of Documentation,* vol. 49, pp. 188-207, 1993.

[36] R. Richardson and A. F. Smeaton, "Using WordNet in a Knowledge-Based Approach to Information Retrieval," Dublin: Dublin City University, 1995.

[37] Z. Wu and M. Palmer, "Verb semantics and lexical selection," in *the 32nd Annual Meeting of the Associations for Computational Linguistics*, Las Cruces, 1994.

[38] D. Lin, "An Information-theoretic definition of similarity," in *15th International Conference on Machine Learning (ICML-98)*, Madison, 1998, pp. 296-304.

[39] G. Pirro, "A semantic similarity metric combining features and intrinsic information content," *Data & Knowledge Engineering,* vol. In Press, Corrected Proof, 2009.

[40] J. J. Jiang and D. W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," in *International Conference on Research in Computational Linguistics (ROCLING X)*, Taiwan, 1997, pp. 19-33.

[41] A. Maguitman, F. Menczer, H. Roinestad, and A. Vespignani, "Algorithmic detection of semantic similarity," in *WWW 2005*, Chiba, 2005, pp. 107-116.

[42] V. S. Zuber and B. Faltings, "OSS: a semantic similarity function based on hierarchical ontologies," in *IJCAI*, Hyderabad, 2007, pp. 551-556.

[43] J. F. Sowa, "Semantic Networks," in *Encyclopedia of Artificial Intelligence*, 2 ed, S. C. Shapiro, Ed.: Wiley, 1992.

[44] H. Dong, F. K. Hussain, and E. Chang, "A hybrid concept similarity measure model for the ontology environment," in *On the Move to Meaningful Internet Systems: OTM 2009 Workshops*, Vilamoura, Portugal, 2009, pp. 848-857.

[45] T. Gruber, "A translation approach to portable ontology specifications," *Knowledge Acquisition,* vol. 5, pp. 199-220, 1995.

[46] H. Dong, F. K. Hussain, and E. Chang, "A context-aware semantic similarity model for ontology environment," *Concurrency and Computation: Practice and Experience,* Submitted.

[47] N. Drummond, M. Horridge, R. Stevens, C. Wroe, and S. Sampaio, "Pizza ontology v1.5," 2007. Available at http://www.co-ode.org/ontologies/pizza/2007/02/12/

[48] D. Kuropka, "Modelle zur repräsentation natürlichsprachlicher dokumente. ontologie-basiertes information-filtering und -retrieval mit relationalen datenbanken," in *Advances in Information Systems and Management Science*, J. Becker, H. L. Grob, S. Klein, H. Kuchen, U. Müller-Funk, and G. Vossen, Eds. Berlin: Logos Verlag Berlin, 2004.

[49] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation,* vol. 28, pp. 11-21, 1972.

[50] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Harlow: ACM Press, 1999.

[51] H. Dong, F. K. Hussain, and E. Chang, "A human-centered semantic service platform for the digital ecosystems environment," *World Wide Web,* vol. 13, pp. 75-103, 2010.

[52] H. Dong, F. K. Hussain, and E. Chang, "Focused crawling for automatic service discovery, annotation and classification in industrial digital ecosystems," *IEEE Transactions on Industrial Electronics,* Accepted.

[53] L. T. Su, "The relevance of recall and precision in user evaluation," *J Am Soc Inf Sci,* vol. 45, pp. 207-217, 1999.