A Novel Web Service Discovery Method Combining Semantic Interface Similarity and Context Similarity

Chenghua Lu*
College of Pearl River, Tianjin University of Finance and Economics, Tianjin, China
Email: iamluchenghua@sina.com

Abstract. Web service discovery is to find the most relevant services to satisfy the requester queries by means of similarity matching between Web services and a service requirement. We propose a novel Web service discovery method based on semantic similarity measure combining I/O similarity and context similarity. A new similarity measurement between terms is put forward as the basis of Web service similarity. Similarity between terms is represented by the similarity of two concepts containing them, and is calculated by means of the length of the shortest path between two concepts. Three kinds of linked relations are considered, i.e. ISA, HASA and antonymy relations. The experiments demonstrate that the proposed approach can improve the accuracy of concept similarity measure. Service discovery based on the similarity measure has better query performance than previous discovery methods in the precision, recall and F-measure.

1. Introduction
Web service is a self-contained, modular software pieces performing a set of business functions that can be described, published, located, and invoked via the Web using open standards. Its architecture implements Service Oriented Architecture (SOA) [1], which includes three component roles (requesters, providers and registries). Service providers publish their services (actually service descriptions) in registries. Service requesters discover the advertised services among a registry, and in turn select and contract them. Service registry serves as a repository consisting of vast Web service description documents. It should provide mechanisms to help providers make their service information available, and requesters find the most appropriate services to fulfill their requirements.

Encouraged by Service-Oriented Computing (SOC) paradigm, an increasing number of Web services have been published via the Web. However, Web services are currently not as broadly utilized as expected. One of the main obstacles is the Web service discovery, i.e., identifying the most relevant services in response to requester queries by using similarity matching between Web services and a service requirement [1, 2, 3].

At present, service registries are commonly based on the Universal Description, Discovery and Integration (UDDI) protocol. The corresponding service discovery systems are commonly based on syntactic key-words search engine. Unfortunately, UDDI-based registries suffer from a lack of knowledge. Providers and requesters possibly have very different knowledge and perspectives about the same service. It is unrealistic to expect providers and requesters to share the common-sense understanding of an application domain, or even there exists a service that fulfills exactly the requirement. To address the shortcoming of UDDI registries and boost automated discovery, Semantic Web Service (SWS) springs up [4, 5]. It is based on the use of ontology for service description with well-defined semantics and facilitates the autonomous publication, discovery, and execution of Web services at the semantic level. For example, for the SWS providing weather information, we utilize a
weather ontology that describes semantic relationship between weather features (temperature, humidity, visibility, precipitation); e.g. precipitation correlates humidity.

We focus on the effective Web services discovery mechanism. In this paper, we proposed a novel Web service discovery method based on semantic similarity measure. On one hand, besides the I/O interface, we extract context consisting of a set of terms from the text description. The context of service can provide more detailed information than I/O interface. Thus, it enhances more relevant services can be retrieved. On the other hand, a new term similarity measurement is defined to estimate the semantic similar degree between terms as the basis of semantic Web service similarity. Similarity can be represented by the similarity of the two concepts containing them, where concept is defined with a taxonomy having various linked relations in a typical hierarchical ontology. Three kinds of linked relations between concepts, i.e. ISA relation, HASA relation and antonymy relation are processed respectively. Furthermore, a similarity measurement for semantic Web service combining I/O similarity and context similarity is presented based on Information Retrieval (IR) techniques, e.g. part-of-speech tagging with CLAWS POS tagger and stemming with Porter’s algorithm.

The remainder of the paper is organized as follows. Section 2 summarizes the related work. Semantic similarity measurement between terms and the semantic-based web service discovery methods are presented in Section 3 and Section 4 in details, respectively. Experiment studies are given in Section 5. Section 6 concludes this paper and presents the future work.

2. Related Work
Due to the significance of service discovery for SOA paradigm, many service discovery methods have been developed [6, 7]. In order to be discovered by Web service search engines, providers publish their Web services (actually the service description documents). WSDL (Web Services Description Language) is a well-adopted standard for non-semantic service description. It provides a syntactic model for describing the interface of Web services. Otherwise Ontology Web Language for Services (OWL-S) and Web Service Modeling Ontology (WSMO) are more advanced protocols to include semantic descriptions. Ontologies serve as the key mechanism to globally define and reference common understanding in a distributed environment. For instance, WordNet is a hierarchical ontology commonly used in Web service discovery. Non-semantic services are usually discovered on syntactic level where textual service metadata in WSDL documents is searched. Non-semantic service discovery leads to the retrieval of much more irrelevant services. Semantic Web service discovery performs a higher level matchmaking utilizing the annotated power of ontologies to improve the performance and enable automation of Web service discovery [6].

Since original Web service descriptions can’t be used for searching, service metadata has to be extracted from Web service descriptions. Various kinds of metadata can be used for service discovery, such as input/output interfaces [7, 8], process model [9] and textual descriptions [10]. I/O interfaces and the textual description of Web services are the mostly used metadata, and so we. In recent work [7, 11], a discovery algorithm based on concept-to-concept similarity measure between I/O interfaces of two services is proposed, and WordNet lexical database is exploited [8]. Process metadata expresses how Web services implement. Kapitsaki [10] matched the Web service description based on context adaptation cases, which are derived from the proposed semantic Web service profile.

Precisely speaking, I/O interfaces show the functional aspects of services and textual descriptions tell us the detailed context information of services. Both of them play an indispensable role in service search; therefore, combining them with an appropriate proportion is an issue worth studying.

Crucially, term similarity measure is the foundation of semantic Web service similarity. Similarity between two terms can be represented by the similarity of the two concepts containing them. Generally speaking, there are three kinds of methods for concept similarity measurement, i.e. hierarchical (or a tree structure) ontology (e.g. WordNet) methods [12], information theory (or corpus-based) methods [13] and web search engine (e.g. Google) methods [14]. Roy Rada et al. [12] proposed that the minimum number of edges connecting two concepts in the hierarchical ontology tree can measure the semantic distance of two concepts. Li et al. [13] introduced a method combining a number of information sources, i.e. structural semantic information from a lexical taxonomy and information content from a corpus. Concept similarity is computed by combining the length of shortest
path and concepts depth. But this method doesn’t distinguish different linked relations between concepts. This will be improved in our paper. Cilibrasi et al. [14] presented a theory of similarity between words based on information distance and Kolmogorov complexity. It uses the World Wide Web as the database and Google as the search engine. It defines the similarity between words with the Normalized Google Distance from the WWW using Google counts. In summary, due to the data sparseness of the information-theory-based methods and the instability of search-engine-based methods, the hierarchical ontology-based method is the mostly used one. Therefore, we exploit WordNet as lexical database and define a new semantic similarity measure to estimate the similar degree between terms as the basis of Web service similarity.

The main contribution of this study is as following. We present a similarity measurement for semantic Web service combining I/O similarity and context similarity. At first, a new semantic similarity measure between terms is defined for hierarchical ontology with various linked relations between concepts. Three kinds of linked relations between concepts, i.e. ISA relation, HASA relation and antonymy relation, are processed respectively rather than only ISA relation of other researches [7, 13]. Different from the previous work, we propose an extended ontology-based service representation by taking advantage of natural language processing techniques including part of speech tagging with CLAWS POS tagger and stemming with Porter’s algorithm. The whole Web service similarity is obtained by combining I/O similarity and context similarity. Experimental results have demonstrated the better accuracy of our method in the retrieval of relevant services.

3. Semantic Similarity Measure between Terms

WordNet is a semantic dictionary—a lexical database, developed at Princeton by a group led by Miller. WordNet 3.0 is used in this study. It partitions the lexicon into nouns, verbs, and so on. They are organized into synonym sets, called synset. A synset represents a concept in which all words have the similar meaning. Consequently, similarity between two words, i.e. terms in our paper, can be represented by the similarity of the two concepts containing them.

Nouns and verbs in WordNet are organized in a tree-like hierarchical structure going from many specific terms at the lower levels to a few generic terms at the top. WordNet provided rich linking relations between concepts, e.g. ISA relation, HASA relation and antonymy relation. Assume $C_1$ and $C_2$ are respectively the corresponding concepts which containing the terms $t_1$ and $t_2$. Similarity between $t_1$ and $t_2$ is equivalent to the similarity of $C_1$ and $C_2$. In the following, we will introduce the process of concept similarity measurement. Before this, it is necessary to introduce two definitions about “the shortest path” and “the semantic distance” as follows.

**Definition 1:** The shortest path between $C_1$ and $C_2$ in the tree-like hierarchical ontology is defined as the summation of path to the lowest common parent concept, including all the upward and downward paths.

**Definition 2:** The semantic distance between $C_1$ and $C_2$ is defined as the summation of weights in the shortest path linked them from the tree-like hierarchical ontology.

Figure 1 shows a hierarchical ontology segment about “book” domain extracted from WordNet. It demonstrates three linking relations between concepts, i.e., ISA, HASA and antonymy relations.
Figure 1. A hierarchical ontology segment extracted from WordNet

In Figure 1, each node represents a concept. For instance, the shortest path between “songbook” and “magazine” is the path <songbook, book, publication, magazine>. Most researches only search the ISA relation to get the shortest path. It is commonly accepted that different relations make a different contribution to the concept similarity. Therefore, besides ISA relation, HASA and antonymy relations are also searched to obtain the shortest path. Different linking relation is assigned an appropriate weight, which represents the semantic distance between two adjacent concepts. A higher weight indicates a larger semantic distance, i.e., a lower similarity between two concepts, and vice versa.

Since the ontology is too huge, we just enumerate a portion of concepts. The root is “entity” and its depth is 0. An arrow with apostrophe is used to connect the indirect parent and sub-concepts. It can be found that concepts at upper layers of the hierarchy have more generalized semantics and less similarity, whereas concepts at lower levels have more concrete semantics and a stronger similarity. Hence, the depth of a concept should be taken into account. Furthermore, in the same layer, concepts with more sub-concepts (i.e. a larger local density) have larger similarity than that with a fewer sub-concepts. The local density includes two parts, i.e., the number of sibling concepts for \( C_1 \) and \( C_2 \) respectively, and it should be taken into consideration on assessing the concept similarity. Assume that the shortest path of \( C_1 \) and \( C_2 \) is shown below, where \( P_i \) is the concept constituted the shortest path.

\[
Path(C_1, C_2) = \langle C_1, P_1, P_2,...,P_s,...,P_e, C_2 \rangle
\]

Before calculating the similarity of \( C_1 \) and \( C_2 \), we first assign a weight value to each edge in the shortest path to scale down the similarity at upper layers and scale up the similarity at lower layers. In other words, the edge weights represent the semantic distance between concepts.

(1) For ISA relation

\[
Weight < P, P_{i+1}> = \frac{e^{-\frac{\log(s+1)}{\log n_p + 1}}}{\log n_p + 2}
\]
Therein, \( n_p \) is the number of sibling concepts for \( P_i \), and \( n_c \) is the number of sibling concepts for \( P_{i+1} \). \( h \) is the depth of the edge (e.g., in Figure 1, the root is “entity,” and the depth of the edge between “reference book” and “book” is 11, that is the depth of “book”). When the depth of an edge increases to infinity, its weight will decrease to 0, and the same to \( n_p \) and \( n_c \). When \( h \) equals to 0, \( n_p \) and \( n_c \) equals to 1, the weight is 0.5. Hence, the weight of an ISA relation scales to \([0, 0.5]\)

(2) For HASA relation

\[
\text{Weight} < P_i, P_{i+1} >= \frac{\varepsilon^{-\log(h+1)}}{\log n_p + 1} \cdot (1 - \frac{1}{\log n_c + 2})
\]

Therein, \( h, n_p \) and \( n_c \) are the same as above (e.g., in Figure 1, there is a HASA relation between “songbook” and “chapter”). HASA relation has different implications than ISA relation. For ISA relation, the sub-concept inherits all properties and semantics of the parent concept. However, HASA relation implies that the parent concept comprises of the sub-concept, i.e., the sub-concept is a part of the parent concept. Therefore, the weight increases with increasing \( n_c \). Otherwise, the larger \( n_p \) is, the semantic mass is more adequate so that the weight decreases with increasing \( n_p \). If an ISA edge and a HASA edge have the same depth and local density, the semantic distance of concepts with HASA relation should be longer than that with ISA relation. From this perspective, we use formula (2) to estimate the weight of a HASA edge. The range of formula (2) is \([0, 1]\).

(3) For antonymy relation

\[
\text{Weight} < P_{i+1}, P_i >= 1 + \frac{\varepsilon^{-\log(h+1)}}{\log n_p + 1}
\]

The antonymy relation is special and exists between sibling concepts. Suppose \( P_i \) was the lowest common parent concept of \( C1 \) and \( C2 \). Antonymy relation only exists between \( P_i \) and \( P_{i+1} \). We add an edge between them instead of the original path \( < P_i, P_p, P_{i+1} > \). Formula (3) is used to estimate the weight of antonymy relation and its range is \([1, 2]\). The semantic distance of antonymy relation is always greater than an ISA or a HASA relation. This can fully reflects the great influence of antonymy relation to the concept similarity measure. With the increase of \( h \) and \( n_p \), the antonymy degree is lower (e.g., in Figure 1, there is an antonymy relation between “amateur” and “professional”).

The semantic similarity of concepts is measured by a notion of semantic distance between them. We first sum up the weight in the shortest path denoted as \( W \), i.e., the semantic distance between two concepts. The computation is shown in formula (4).

\[
W = \sum_{P_i, P_{i+1} \in \text{Path}(C1, C2)} \text{Weight}(P_i, P_{i+1})
\]

(4)

Semantic distance and semantic similarity are inversely related measurements. The shorter the semantic distance is, the greater the semantic similarity. When the semantic distance is 0 (i.e. \( t_1 \) and \( t_2 \) are containing in the same concept, \( C1 = C2 \)), the similarity should be 1. When the semantic distance increases to infinity, the similarity should decrease to 0. The similarity ranges from 0 to 1. We use a negative exponential function of semantic distance to compute the similarity, as shown in formula (5). Similarity between terms \( t_1 \) and \( t_2 \) equals to the similarity of concepts \( C1 \) and \( C2 \) which contain them.

\[
\text{SimTerm}(t_1, t_2) = \text{SimCon}(C1, C2) = e^{-W}
\]

(5)

4. Semantic-Based Web Service Discovery

We proposed a novel semantic-based Web service discovery method combining I/O similarity and context similarity. Firstly, we extract I/O interfaces and context respectively from the available service and service request, denoted by \( IO_s \), \( IO_R \), \( T_s \) and \( T_R \). Secondly, we respectively implement the measurement of I/O similarity and context similarity on the strength of WordNet. Ultimately, combine both of them and calculate the similarity of available service and service request, returning a list of candidate services for the requester.
4.1. Context Similarity Measurement

From an IR-based viewpoint, in order to establish a match between a request and a Web service, the search mechanism exploits some natural language processing techniques to extract different service metadata from Web service description. Accordingly, a definition is firstly given as follows.

**Definition 3:** For the consistency, a Web service and a service request are all defined as a three tuple: $S = \langle Input, Output, Context \rangle$. The representation of service capability is on the basis of input, output and context. Input/output interfaces provide abstract functional aspects of services and context denotes the detailed information of services.

Before the semantic Web service discovery process, we first filter the unsatisfactory services by matching the numbers of I/O interfaces between available service and request. The following conditions should be satisfied. In this way, the computational efficiency of the semantic-based Web service discovery is improved.

1. $\text{Num}(I_0) \leq \text{Num}(I_k)$, $\text{Num}(I_0)$ and $\text{Num}(I_k)$ are respectively the number of input parameters of available service and service request.

2. $\text{Num}(O_0) \geq \text{Num}(O_k)$, $\text{Num}(O_0)$ and $\text{Num}(O_k)$ are respectively the number of output parameters of the available service and service request.

In this section, we extract the context by using some natural language processing techniques, i.e. part of speech tagging with CLAWS POS tagger and stemming with Porter’s algorithm. With respect to the similarity measurement of term sets, some researchers have used the method of bipartite graph matching [11]. However, traditional bipartite graph matching is only appropriate for the case when the term sets have the same length. In this paper, we exploited a bidirectional matching method appropriate for the case when the term sets have a different length. The computational process of context similarity is as follows.

1. We carry out the part of speech tagging onto the text description of the available service and request by using CLAWS POS tagger (http://ucrel.lancs.ac.uk/claws/trial.html), and then extract two parts of speech (i.e., nouns and verbs) according to the part of speech label. After that, every term is transformed into its root form with Porter’s stemming algorithm (http://facweb.cs.depaul.edu/mobasher/classes/csc575/porter.html). Stemming reduces inflected (or sometimes derived) terms to their stem, base or root form. Some high frequency terms are removed (e.g. “service” and “they”), which almost exist in every text description. By means of the preprocessing, we get the context consisting of nouns and verbs term sets as follows, represented by $T_S$ and $T_R$.

$$T_S = \{N_S, V_S\}, T_R = \{N_R, V_R\}$$ (6)

2. Calculate the similarity between all terms in the term sets of available service and request to get the similarity matrix, denoted by $MN(N_S, N_R)$ and $MV(V_S, V_R)$. The similarity measure between terms has been discussed in Section 3. Afterwards, calculate the similarity of term set between available service and service request according to formula (7). Therein, $|N_S|$, $|N_R|$, $|V_S|$, and $|V_R|$ are the number of terms in term set of nouns and verbs.

$$SimN(N_S, N_R) = \frac{1}{2} \left( \frac{1}{|N_S|} \sum_{k=1}^{|N_S|} \max_{1 \leq l \leq |N_R|} (\text{SimTerm}(N_{Sk}, N_{Rl})) + \frac{1}{|N_R|} \sum_{l=1}^{|N_R|} \max_{1 \leq k \leq |N_S|} (\text{SimTerm}(N_{Rl}, N_{Sk})) \right)$$

$$SimV(V_S, V_R) = \frac{1}{2} \left( \frac{1}{|V_S|} \sum_{k=1}^{|V_S|} \max_{1 \leq l \leq |V_R|} (\text{SimTerm}(V_{Sk}, V_{Rl})) + \frac{1}{|V_R|} \sum_{l=1}^{|V_R|} \max_{1 \leq k \leq |V_S|} (\text{SimTerm}(V_{Rl}, V_{Sk})) \right)$$

$$\text{SimTerm}(N_{Sk}, N_{Rl}), \text{SimTerm}(V_{Sk}, V_{Rl}) \in MN(N_S, N_R), \text{SimTerm}(V_{Rl}, V_{Sk}) \in MV(V_S, V_R)$$ (7)

3. Finally, context similarity is computed by combining the similarity of nouns and verbs term sets with appropriate weights, as shown in formula (8).

$$SimT(T_S, T_R) = \frac{n \cdot SimN(N_S, N_R) + v \cdot SimV(V_S, V_R)}{n + v}$$ (8)

Generally speaking, nouns make the greatest contribution to text similarity measure, and followed
by verbs. Likewise, the service text description is also a short text so that the three weights are similarly set as follows: \( n=0.55 \) and \( v=0.45 \). Furthermore, we consider the absence of a certain component, e.g. there are verbs in \( T_S \) but not in \( T_R \). For this case, we set \( l=0.9 \) and \( v=0 \). This is the same to other components. Moreover, when \( T_S \) and \( T_R \) have the same kinds of components, \( l=1 \).

4.2. I/O Interface Similarity Measurement
Our method of term set similarity measure is applied to I/O interface similarity measure. The weight between input interface and output interface are both set as 0.5, as shown in formula (9).

\[
\text{SimIO}(I_O, O_R) = \frac{1}{2}(\text{SimI}(I_S, I_R) + \text{SimO}(O_S, O_R))
\]

(9)

From the aforementioned work, we have got the context similarity and I/O interface similarity between available service and service request. The two parts are combined with an appropriate weight \( \alpha \) as shown in formula (10).

\[
\text{SimSR}(S, R) = \alpha \cdot \text{SimIO}(I_O, O_R) + (1-\alpha) \cdot \text{SimT}(T_S, T_R)
\]

(10)

5. Experiment Studies

5.1. Concept Similarity Evaluation
We evaluate our concept similarity measure by comparing it with human rating in the Miller-Charles [15] dataset which is the commonly used benchmark dataset. Miller and Charles asked a group of 38 human to rate 30 word pairs for similarity on a scale from 0 to 4. These 30 word pairs are a subset of Rubenstein-Goodenough’s original dataset of 65 word pairs. It consists of 10 high level, 10 intermediate level and 10 low level word pairs in terms of semantic similarity. Much work uses 28 word pairs of them, not including the two pairs “shore-woodland” and “cemetery-woodland”. For a convenient comparison, we also make use of these 28 word pairs in our experiment.

Both Pearson correlation coefficient and Spearman correlation coefficient are exploited as the evaluation criterion. Pearson correlation coefficient is used for the normalized similarity correlation of each method and human rating. Considering the compared methods may vary with ranges of similarity, the Spearman correlation coefficient is used to measure the rank correlation between each method and human rating. The larger the rank correlation, the computed result gets closer to the human rating.

We compare our method with other three methods [13, 16, 17]. Li et al. [13] introduced a method of semantic similarity measure by combining multiple information sources including the structural semantic information from a lexical taxonomy and information content from a corpus. Concept similarity is obtained by combining the shortest path length and concepts depth. However, this method doesn’t distinguish different linked relations between concepts, and we have done. Liu et al. [17] proposed to calculate the length of the path between two concepts in the weighted hierarchy to measure their similarity. They consider ISA and HASA relations but not antonymy relation. Bollegala et al. [16] defined semantic similarity by using page counts and text snippets retrieved from a Web search engine for two words. Table 1 shows the experimental results for every method’s calculation correlation to Miller-Charles’ dataset (in column MC replica).

It is observed that our proposed similarity measure outperforms other three methods. We search for three kinds of relations from WordNet in obtaining the shortest path. Different linking relations are assigned different weights. The method [16] also gets a higher correlation coefficient. However, this method has a great instability due to the instability of Google server. In addition, the text snippets are difficult to extract and have many noises resulting in inaccurate results. Our method has the potential to remove this problem with the stable and newest-version lexical database WordNet.
Table 1. Semantic similarity comparisons for various methods on Miller-Charles’s 28 word pairs

| Word pairs          | MC replica | Li’s measure | Liu’s measure | Search engine measure | Our measure |
|---------------------|------------|--------------|---------------|-----------------------|-------------|
| chord-smile         | 0.13       | 0.01         | 0.00          | 0.01                  | 0.08        |
| rooster-voyage      | 0.08       | 0.00         | 0.00          | 0.05                  | 0.09        |
| noon-string         | 0.08       | 0.01         | 0.00          | 0.00                  | 0.07        |
| glass-magician      | 0.11       | 0.04         | 0.00          | 0.05                  | 0.04        |
| monk-slave          | 0.55       | 0.26         | 0.12          | 0.24                  | 0.14        |
| coast-forest        | 0.42       | 0.10         | 0.00          | 0.15                  | 0.16        |
| monk-oracle         | 1.10       | 0.10         | 0.00          | 0.80                  | 0.28        |
| lad-wizard          | 0.42       | 0.26         | 0.10          | 0.23                  | 0.15        |
| forest-graveyard    | 0.84       | 0.04         | 0.00          | 0.44                  | 0.22        |
| food-rooster        | 0.89       | 0.00         | 0.00          | 0.02                  | 0.20        |
| coast-hill          | 0.87       | 0.19         | 0.00          | 0.36                  | 0.55        |
| car-journey         | 1.16       | 0.00         | 0.00          | 0.17                  | 0.18        |
| crane-implement     | 1.68       | 0.25         | 0.07          | 0.06                  | 0.64        |
| brother-lad         | 1.66       | 0.26         | 0.10          | 0.13                  | 0.63        |
| bird-crane          | 2.97       | 0.38         | 0.44          | 0.85                  | 0.56        |
| bird-cock           | 3.05       | 0.72         | 0.80          | 0.87                  | 0.58        |
| food-fruit          | 3.08       | 0.02         | 0.00          | 0.94                  | 0.40        |
| brother-monk        | 2.82       | 0.72         | 0.80          | 0.27                  | 0.59        |
| asylum-madhouse     | 3.61       | 0.72         | 0.80          | 0.79                  | 0.84        |
| furnace-stove       | 3.11       | 0.05         | 0.00          | 0.88                  | 0.48        |
| magician-wizard     | 3.50       | 0.98         | 1.00          | 1.00                  | 1.00        |
| journey-voyage      | 3.84       | 0.72         | 0.80          | 1.00                  | 0.90        |
| coast-shore         | 3.70       | 0.59         | 0.75          | 0.97                  | 0.76        |
| implement-tool       | 2.95       | 0.68         | 0.77          | 0.50                  | 0.85        |
| boy-lad             | 3.76       | 0.71         | 0.79          | 0.96                  | 0.86        |
| automobile-car      | 3.92       | 1.00         | 1.00          | 0.92                  | 1.00        |
| midday-noon         | 3.42       | 0.99         | 1.00          | 0.99                  | 1.00        |
| gem-jewel           | 3.84       | 0.96         | 1.00          | 0.82                  | 1.00        |
| **Pearson**         | 1.00       | 0.891        | 0.850         | 0.870                 | **0.907**   |
| **Spearman**        | 1.00       | 0.751        | 0.750         | 0.850                 | **0.912**   |

5.2. Semantic Web Service Discovery Evaluation

We compare our method with other three methods [7, 8, 11]. With respect to the impact weight of context similarity in formula (12), we choose to use $\alpha=0.6$ experimentally. We employ the well-known service retrieval dataset OWL-S TC v3.0. Web services are mostly retrieved from public IBM UDDI registries. It provides 1007 semantic Web services from seven different domains, i.e., Education, Medical, Food, Travel, Communication, Economy, and Weapon. Moreover, it provides a set of 29 test queries which are associated with relevance sets to conduct performance evaluation experiments.

5.2.1. Performance criteria. Precision, recall and F-measure are standard, well-known measures for evaluating service retrieval performance. Precision is the number of returned services that are actually relevant for the corresponding request divided by the number of all returned services. Recall is the number of returned services that are actually relevant for the corresponding request divided by the number of all relevant services that should have been returned. The F-measure score can be interpreted as a weighted average of the precision and recall. It is shown in formula (11), where an F-measure score reaches its best value at 1 and worst value at 0.

$$\text{Precision} = \frac{\text{Relevant} \cap \text{Returned}}{\text{Returned}}, \text{Recall} = \frac{\text{Relevant} \cap \text{Returned}}{\text{Relevant}}, \text{F-measure} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$  \hspace{1cm} (11)

5.2.2. Results under different thresholds. A parameter $\varepsilon$ is used as the threshold of service similarity to be considered as relevant to the request. This experiment aims to find a suitable value of $\varepsilon$ for each
domain. $\varepsilon$ is set as 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9. We calculate the precision, recall and F-measure score of seven domains. The result is shown in Figure 2, 3, 4 respectively.

![Figure 2. Precision under different $\varepsilon$](image1)

![Figure 3. Recall under different $\varepsilon$](image2)
We can find that precision increases with $\varepsilon$ increasing and recall decreases with $\varepsilon$ increasing in every domain. Figure 4 demonstrates that each domain has a different optimal threshold $\varepsilon$ at which reaching the best F-measure score. Meanwhile, the experimental results indicate that Web services of different domains have different characteristics, mainly embodied in I/O interfaces and context.

5.2.3. Comparison with other methods. We will compare our approach with other three methods, i.e., URBE [8], SimRSP[11], and SDMA [7]. URBE is an approach for Web service retrieval based on the evaluation of similarity between Web service interfaces. SimRSP is an approach for Web service query based on concept relaxation which employs the semantic relation of concepts from hierarchical ontology. Both URBE and SimRSP use only I/O interface information and employ a bipartite graph matching approach to compute the similarity between term sets. SDMA uses only context information. In view of the concept similarity measure, URBE, SimRSP and SDMA just compute the ISA relation between concepts. URBE measures concept similarity by means of information content method. SimRSP and SDMA use the shortest path method to compute concept similarity.
In Figure 5, there is only the F-measure score in SDMA and without precision and recall. It is observed that our measure outperforms the methods of URBE and SimRSP both in the Precision and Recall. Our proposal performs overall better than the other three methods and achieves the best F-measure score. Our semantic-based Web service discovery method benefits from both considering various linked relations between concepts and the integration of I/O interfaces and context.

6. Conclusions and Future Work
In this paper, we presented a vision for Web service discovery based on semantic similarity measure which combines I/O similarity and context similarity. Experiments have been done to evaluate the performance of our proposed method. The result has shown that the concept similarity-assessment method we developed is more precise than other similar methods, greatly facilitating the Web service discovery process. As a future work, we will extend our work in two directions. In the first place, with the service documents longer and more complicated, the computational efficiency is an issue deserving to be studied. In a second place, we are planning to develop a Web service discovery system based on our algorithm and improve its self-adaptation for the unexpected changes especially in heterogeneous environment.

7. Acknowledgements
The work was supported by the major project research topic in advance from college of Pearl River, Tianjin University of Finance and Economics in 2019 (No. ZJZD19-05).

8. References
[1] Xia H, Hu C, Xiao F, et al. An efficient social-like semantic-aware service discovery mechanism for large-scale Internet of Things[J]. Computer Networks, 2019, 152: 210-220.
[2] Ma X J, Chen J D, Li K. Method of Combining I/O with Information Content for Discovering Semantic Web Services[J]. Computer Systems & Applications, 2016, 25(2): 141-145.
[3] Azmy M R, Muhamad W. Advanced Technologies to Support Service Discovery in Service-Oriented Systems[C]//2020 International Conference on Information Technology Systems and Innovation (ICITSI). IEEE, 2020: 300-305.
[4] Cheng B, Li C, Chen J. A Web Services Discovery Approach Based on Interface Underlying Semantics Mining[J]. IEEE Transactions on Knowledge & Data Engineering, 2017, PP(99):1-1.
[5] Gomes P, Cavalcante E, Batista T, et al. A semantic-based discovery service for the Internet of Things[J]. Journal of internet services and applications, 2019, 10(1): 1-14.
[6] BENSALAH AZIZOU Z, Boudries A, Amad M. Decentralized service discovery and localization in Internet of Things applications based on ant colony algorithm[J]. International Journal of Computing and Digital Systems, 2020, 9(5): 941-950.

[7] Farrag T A, Saleh A I, Ali H A. Semantic web services matchmaking: Semantic distance-based approach [J]. Computers & Electrical Engineering, 2013, 39(2):497-511.

[8] Plebani P, Pernici B. URBE: Web Service Retrieval Based on Similarity Evaluation[J]. IEEE Transactions on Knowledge & Data Engineering, 2009, 21(11):1629-1642.

[9] Shafiq O, Alhajj R, Rokne J. Log based Business Process Engineering using Fuzzy Web Service Discovery[J]. Knowledge-Based Systems, 2014, 60(2):1-9.

[10] Kapitsaki G M. Web service matchmaking for the development of context-aware applications[J]. Iet Software, 2012, 6(6):536-548.

[11] Ou W J, Zeng C, Xiang X M, et al. Efficient Web Service Query Approach Based on Concept Relaxation[J]. Chinese Journal of Computers, 2011, 34(12):2381-2390.

[12] Rada R, Mili H, Bicknell E, et al. Development and application of a metric on semantic nets[J]. IEEE Transactions on Systems Man & Cybernetics, 1989, 19(1):17-30.

[13] Li Y, Bandar Z A, Mclean D. An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources[J]. Knowledge & Data Engineering IEEE Transactions on, 2003, 15(4):871-882.

[14] Cilibrasi R L, Vitanyi P M B. The Google Similarity Distance[J]. IEEE Transactions on Knowledge & Data Engineering, 2004, 19(3):370-383.

[15] Miller G A, Walter G. Charles. Contextual correlates of semantic similarity[J]. Language Cognition & Neuroscience, 1991, 6(1):1-28.

[16] Bollegala D, Matsuo Y, Ishizuka M. A Web Search Engine-Based Approach to Measure Semantic Similarity between Words [J]. IEEE Transactions on Knowledge & Data Engineering, 2010, 23(7):977-990.

[17] Liu W, Quan X, Feng M, et al. A short text modeling method combining semantic and statistical information[J]. Information Sciences An International Journal, 2010, 180(20):4031-4041.