Indonesian Journal of Electrical Engineering and Computer Science
Vol. 24, No. 2, November 2021, pp. 1228~1237
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v24.i2.pp1228-1237

A semantic web services discovery approach integrating multiple similarity measures and k-means clustering

Mourad Fariss, Naoufal El Allali, Hakima Asaidi, Mohamed Bellouki
Mohammed First University Oujda, FPD Nador, LMASI, Nador, Morocco

Article Info

Article history:

Received Jul 3, 2021
Revised Sep 17, 2021
Accepted Sep 22, 2021

Keywords:

K-means
Ontology
Semantic measure of similarity
SOA
Web service discovery

ABSTRACT

Web service (WS) discovery is an essential task for implementing complex applications in a service oriented architecture (SOA), such as selecting, composing, and providing services. This task is limited semantically in the incorporation of the customer's request and the web services. Furthermore, applying suitable similarity methods for the increasing number of WSs is more relevant for efficient web service discovery. To overcome these limitations, we propose a new approach for web service discovery integrating multiple similarity measures and k-means clustering. The approach enables more accurate services appropriate to the customer's request by calculating different similarity scores between the customer's request and the web services. The global semantic similarity is determined by applying k-means clustering using the obtained similarity scores. The experimental results demonstrated that the proposed semantic web service discovery approach outperforms the state-of-the approaches in terms of precision (98%), recall (95%), and F-measure (96%). The proposed approach is efficiently designed to support and facilitate the selection and composition of web services phases in complex applications.

Corresponding Author:

Mourad Fariss
LMASI, FPD Nador
Mohammed First University
Nador, BV Mohammed VI B.P. 524 Oujda 60000 Morocco
Email: m.fariss@ump.ac.ma

1. INTRODUCTION

Web service discovery is to find the relevant web services that satisfy the service customers' requirements. Owed to the increased number of published web services (WSs) on the Internet, the discovery stage offers an important number of candidate WSs for a given request. The WS discovery can be performed syntactically based on keywords or semantically based on WS description. The syntactic mechanism is limited to respond to the functional customer requirements. This makes introducing a new mechanism necessary, which involves the localization of WSs based on the capabilities they offer. Integration of semantic techniques in WSs can play an important role in incorporating different terminologies in WSs. Furthermore, adopting the right measure similarity to find most similar WSs during the discovery phase is complex task.

The traditional universal description, discovery, and integration (UDDI) only supports keywords for searching WSs; this search cannot find the whole relevant WSs for customers [1], [2]. Keywords are insufficient to express semantic concepts, semantically different concepts can have the same representation, which further influences the accuracy [3]. Hence, several approaches have been proposed to add the semantic concepts to the WS description as web service modeling language (WSML), web service description language with semantics (WSDL-S), ontology web language for services (OWL-S), to facilitate the discovery and selection tasks [4]. Many existing web service discovery approaches are based on operational signing and several matching mechanisms. We differentiate between logic-based reasoning [5] and non-logic-based techniques (i.e., graph matching, similarity measures [6], [7], graph matching, and data mining [8]) and hybrid discovery [9], [10] (i.e., methods that utilize logical and non-logical matching both).

Several works develop the notion of similarity in the discovery of web services [11]-[17]; some researchers have proposed and studied their similarity measures as in [12], [18], [19]. Others have worked with already existing measures like in [20]-[22]. Even though the existing of multiple similarity measures to calculate the similitude between the web service and the customer request, they are some particularities. In more detail, the web services are modeled using different formats, the enough published ontologies of WSs with OWL-S, the accurate information about the WS during the service discovery is required. According to these particularities, concentrating on one measure of similarity can lose the interests of other measures as well as can influence the results by its limitations.

This paper proposes a new solution for the similarity measure to calculate the similarity between a web service from a dataset of semantic web services and customers' requests. On solving this problem, others systematically have an avenue for improving results, such as selecting web services and the composition of web services. Consequently, a new approach to solve the problem of semantic web services discovery by decreasing the number of discovered WSs through increasing the precision. Our approach consists of calculating six similarity measures, namely; Euclidean distance, Manhattan distance, Wu and Palmer'similarity, Cosine similarity, Jaccard's index, and logical correspondence. and use the k-means clustering to designate the discovered web services most similar to the customer's request based on the similarity scores of each WS.

The remainder of the paper is structured as follows: section 2 includes the related works and the motivation, section 3 details the problem and the necessary background to solve it. Section 4 gives our proposed contribution. Section 5 discusses the experimental results. Section 6 closes the paper with the conclusion.

2. RELATED WORK AND MOTIVATION

We can mention a set of matchmakers for WSs developed in the literature, such as [23], that applied three functions to calculate the lexical specification similarity and showed the best performance for the classical vector space model. Surprisingly, semantic similarity metrics did not help improve the service interface mapping accuracy and recall. Classical term frequency–inverse document frequency (TF-IDF) heuristics outperformed other approaches in most cases. Due to the excessive generality of the WordNet ontology, many false correspondences have been found. The study concluded that the semantic similarity has no gain in precision due to the choice of the dataset, which is WSDL files that influence the results obtained.

The Condorcet-fuse system [24] is based on a majority voting scheme. More explicitly, a document d1 is classified before another document d2 in the combined list, if d1 is classified before d2 more times than d2 is classified before d1. The outranking approach [25] adjusts the majority voting model by setting a set of thresholds. Experimental evaluation notes that performance is very delicate to threshold values. The proposed approach used different similarity measures through the top-ranking ratio.

Fethallah *et al*. [26] focused on automatic service discovery based on the semantic web. As a solution, they proposed an approach that utilizes the service interface and the domain ontology to model web services. Then, they calculated the similarity score using a matching algorithm between the request and the web service model, which is based on the Wu and Palmer similarity measure. The dataset used to evaluate the approach is sampled from the ontology web language services-test collection (OWLS-TC) corpus version 2.2.1. This approach is characterized by its simplicity and its weak complexity, but still show a weakness in recall.

Wu *et al*. [27], a method was proposed to facilitate clustered web service discovery using WSDL document meta tags. The efficiency of clustering was maintained by the unstable distribution and classification of noise tags. In addition, the authors proposed automated web service clustering by tagging web services on a domain or UDDI search engine or ontology. However, syntax-based approaches with a lower performance drawback due to the complexity of natural language, a variety of semantic approaches are also suggested.

Renzis *et al*. [28] aims to suggest a solution based on applying case-based reasoning in selecting and discovering web services tasks. A similarity function determines the score of similarity among two cases. Furthermore, the approach combines the concepts of case-based reasoning (CBR) using WordNet and distributionally similar words using co-occurrences (DISCO) as a lightweight semantic basis. The result is case management, which can increase the visibility of the appropriate services to accomplish certain required features; this service has been returned as a proposed solution in approximately 90% of cases. However, the experiments are performed on small dataset of web services (62 services).

The work done in [29] allocated the web service discovery problem based on integration algorithms. The contribution consists of computing the best WSs following an outranking relationship. To evaluate their approach, the authors used the OWLS-TC version 2.2 benchmark. The results presented are impressive but remain low in terms of precision (66%). Fellah *et al.* [30], gave a web services discovery framework to perform the semantic interoperability of WSs in a multi-ontological environment. A proposed algorithm for computing the similarity of concepts between ontologies was developed, which comprises a combination of different local similarity measures, taking into account all items and semantic structures of the origin and objective ontologies. However, the dataset component is based on semantic annotations for WSDL and XML schema (SAWSDL) language, which is rarely used.

To solve the problem of web services discovery, researchers adopted the notion of the semantic web, as well as measures of similarity to determine the similarity between the customer request and the published WSs. These works are divided into two categories, as shown in Table 1. Some researchers have developed their similarity measures; others have worked with already existing measures. However, all these proposal approaches remain limited because of several factors like the choice of similarity measures suitable for the web services discovery problem and the precision that can be considered low for these approaches. These limitations have encouraged us to propose a semantic discovery web services approach with attractive precision without focusing on a specified similarity measure.

Table 1. Functional comparison of our proposal with the previous studies

	Syntactic/Semantic	Dataset used	Similarity measure/method
[23] (2006)	Syntactic	-	Term frequency-inverse document frequency
[25] (2008)	Semantic	Topic Distillation (TD) task of TREC-2004 Web track	Majority voting
[26] (2010)	Semantic	OWLS-TC V. 2.2	Wu&Palmer's Similarity
[27] (2014)	Syntactic	STag Dataset 1.0	Jaccard coefficient
[28] (2016)	Syntactic and Semantic	-	Case-based Reasoning
[29] (2017)	Semantic	OWLS-TC version 2.2	Cosine, Extended Jaccard, Jenson Shanon, information loss, logic matching.
[30] (2019)	Semantic	Benchmark of Ontology Alignment Evaluation Initiative OAEI	Individual similarity measure
Our approach	Semantic	OWLS-TC v4.0.	Euclidean distance, Manhattan distance, Wu and Palmer, Cosine, Jaccard's Index, and Logical Correspondence

3. PRELIMINARIES

This section provides the necessary background for understanding the remainder of this paper, including the web service discovery, the web service clustering, and the web service similarity.

3.1. Web service clustering

A data object cluster can be treated together as a group and seen as data compression. Calculating the similarity between objects is usually the first step in a clustering algorithm. Pre-clustering aims to decrease the search area. With computationally intensive semantic similarity calculation, the service matching process can be more beneficial in a specific group than in a large group of unrelated services. The similarity of the WSs is first calculated, then we can calculate the distance between the two services.

$$Distance\ (WSA, WSB) = 1/Service\_Sim(WSA, WSB) \quad (1)$$

Many algorithms were proposed to cluster data have recently emerged. They can be classified into exclusive clustering, hierarchical clustering, overlapping clustering, and probabilistic clustering [31], [32]. Our contribution exploits the k-means algorithm; it is an exclusive clustering and one of the most used algorithms for clustering. Based on the similarity, the clustering algorithms are used to group WSs.

K-means is one of the simplest forms of the unsupervised algorithm [33] used to solve clustering problems. The algorithm defines certain data by a specific number if the clusters are determined a priori [14]. Despite all its advantages [34], k-means has some weaknesses, but these later ones do not affect our contribution to studying the similarity score between consumers' queries and published WSs. Our approach consists of using clustering through classification and filtration rather than similarity.

3.2. Web service similarity

Semantic similarity is a metric across a set of data based on the similitude of their meaning. It presents the correspondence between two ontological concepts or taxonomies concepts, and it defines a distance among words by statistical means. The similarity between concepts is a quantitative measure of information determined based on their attributes and relations. It has a central role in an environment such as the semantic web, where data come from multiple sources and must flexibly be combined and integrated. Given many similarity measures, we focus on the six most used in the discovery web services domain. The following detail of each similarity measure respectively: Euclidean distance, Manhattan distance, Wu and Palmer, Cosine, Jaccard's index, and logical correspondence.

3.2.1. Euclidean distance

The Euclidean distance is a geometric problem standard measure. The usual distance between two points can be measured easily in two- or three-dimensional space. This measure is commonly employed to solve clustering problems and is considered the most desirable matching measure when the data is dense or continuous. It satisfies the four conditions above and is, therefore, an accurate scale. The Euclidean distance is also the standard distance measure utilized in the K-Mean algorithm.

To measure the Euclidean distance between two data elements d_1 and d_2 represented by their vectors $\overrightarrow{v_1}$ and $\overrightarrow{v_2}$ respectively. The equation is as follows:

$$D_E(\overrightarrow{v_1}, \overrightarrow{v_2}) = \sqrt{\left(\sum_{v=1}^{m} \left|w_{t,1} - w_{t,2}\right|^2\right)} \quad (2)$$

where the term set is $T = \{v_1, \dots, v_m\}$, and the *tfidf* value is utilized as term weights, that is $w_{t,1=}tfidf(d_1, v)$.

3.2.2. Manhattan distance

The Manhattan distance can be calculated as the sum of the absolute differences between the Cartesian coordinates of two points. We can say that it is the sum of the difference between the coordinates *x* and *y*. The distance between two points p_1 and p_2 , respectively with the coordinates (x_1, y_1) and (x_2, y_2), measured along the axes at the right angles is:

$$Manhattan\ distance = |x_1 - x_2| + |y_1 - y_2| \quad (3)$$

the distance Manhattan is frequently used in integrated circuits where the wires run only parallel to the *X* or *Y* axis. It is also named Manhattan length, straight distance, *L1* distance Minkowski or *L1* norm, taxi scale, or block distance.

3.2.3. Wu and Palmer'similarity measure

Wu and Palmer's [18] similarity measure generally uses information from the shortest path of two concepts, the specificity or prevalence of the two concepts in the ontology hierarchy, and their relations with other concepts. The authors propose a similarity measure to find the most specified common concept underlying the two measured concepts. The more specific common concept path length is computed by adding the IS-A links to the two compared concepts. The following equation present the formula of the Wu and Palmer' similarity:

$$S_{W\&P}(C_1, C_2) = 2H/(N_1 + N_2 + 2H) \quad (4)$$

where N_1 and N_2 are the numbers of IS-A links of C_1 and C_2 respectively with the most specific general concept *C*, and *H* is the number of IS-A links of *C* to the root of an ontology. This measure of similarity is between 1 and 0.

3.2.4. Cosine similarity

Cosine similarity is one of the successful similarity measures used with text documents in many information retrievals and clustering applications [21], [35]. The cosine similarity algorithm utilizes the angle between two vectors in the vector space to define the difference in content between two vectors [36]. It is essentially based on the consumer's preference and the level of variation between provided WSs. It determines a semantic WS that sequentially conforms to the consumer's context when it feeds the semantic WS back to the consumer to meet the consumer's different needs in the consumer's context. Such as $\overrightarrow{d_1}$ and $\overrightarrow{d_2}$ two documents, their cosine similarity is:

$$cos(\overrightarrow{d_1}, \overrightarrow{d_2}) = \overrightarrow{d_1}.\overrightarrow{d_2}/(|\overrightarrow{d_1}| \times |\overrightarrow{d_2}|) \quad (5)$$

since the cosine similarity algorithm concentrates on the difference of vectors' directions, it is not susceptible to their size. Hence, consumers mainly use it to determine whether the WS content is interesting.

3.2.5. Jaccard's index

The Jaccard Index (also named the Jaccard Similarity Coefficient) is a statistic utilized to compare the diversity and similarity of the sample set [37]. This Index is defined as present in (6), as the number of shared objects divided by the total number of objects, minus the number of common objects.

$$g^{(J)}(x_a, x_b) = x_a^T x_b / (\|x_a\|_2^2 + \|x_b\|_2^2 - x_a^T x_b) \tag{6}$$

3.2.6. Logical correspondence

There are four types of matches: exact, plugin, subsume and fail. The definition of each type is as follows:

- **EXACT**: if the concept RQout and ASout are in the same ontology class.
- **PLUGIN**: if the ASout concept in the ontology is a subclass of RQout, in this case, the RQout concept is more specific than the searched ASout concept.
- **SUBSUME**: if the class of RQout is more general than ASout, the ASout class of the ontology is a sub-class of RQout.
- **FAIL**: if there is no subsumption similarity between RQout and ASout in the ontology.

For example, to compare all the output concepts of the RQout query with the set of output concepts of the ASout note service, we apply:

$$LOGIC(RQout, ASout) = MIN_{PieRQout}\big(logMatch1(P_1, Asout)\big)$$

$$GMatch1(P_l, ASout) = MIN_{pieRQout}(logMatch1(P_l, ASout))$$

if the two concepts are declared complementary or distinct, the zero scores are given directly. Otherwise, we check the subsumption below and supply the score for the interval [0, 1] according to Table 2.

Table 2. Subsumption based score assignments

Relationship	Description	Score
EXACT	*RQout* and *ASout* are equivalent	1.0
PLUGIN	*RQout* is a parent of *ASout*	0.6
SUBSUME	*ASout* is a subclass of *RQout*	0.4
FAIL	No subsumption relation between *RQout* and *ASout*	0

4. PROPOSED APPROACH

To solve WSs discovery, our proposal introduces the notion of clustering and similarity score between the user's request and the available services. The similarity measures aim to compute the similarity score between the customer request and the services available in the UDDI register. This similarity becomes possible with the semantic aspect by utilizing OWL-S services, knowing that there are several methods of semantic description. The choice of an ontology language based on OWL-S is justified in our previous work [4], the most standardized and perhaps the most complete semantic WS deployed technology.

In our approach, we chose to utilize different similarity measures because of the difference in the precision and effectiveness of each measure. This led to applying the k-means clustering to filter and classify the discovered services according to their similarity scores to minimize the complexity of the selection phase. One of the principal difficulties of k-means clustering is to define in advance the number of clusters. A simple strategy to achieve this is the elbow method, which consists of varying K and following the evolution of the intra-class inertia. The idea is to visualize the "elbow," where the addition of a class does not correspond to anything in the data structuring.

The obtained discovered WSs belong to the class having the closest centroid to 1, which means that the WSs belonging to this class have similarity scores globally close to 1. Hence, they will be the discovered services that are the most similar to the sent query. Our proposed approach aims to solve WSs discovery using the notion of similarity and the k-means algorithm. The steps of this approach can be shown as follows:

- Step 1: consists of calculating the six measures of similarity employed in our approach; Euclidean distance, Manhattan distance, Wu and Palmer's similarity measure, Cosine similarity, Jaccard Index, and Logical Correspondence; between the customer request and the WSs available in the WSs registry. These similarity scores will be saved in the temporary database, which will be utilized in the second step. This phase allows us to surmount the problem of choosing suitable similarity measures for the WS discovery problem. And not focusing on one measure of similarity can make us overlook the benefits of other measures.

- Step 2: after creating the database of similarity scores achieved by the six similarity measures, we employ the k-means algorithm on these scores to identify the WSs classified in different clusters according to the degree of similarity between the consumer's request and the WSs. The use of k-means clustering is preceded by the specification of the K number of clusters utilizing the elbow method.
- Step 3: define the group of WSs most similar to the client request among the clusters obtained from the previous step. The suitable cluster is the one with a centroid closer to 1. This cluster must be the set of WSs discovered by the system and respond most effectively to the customer's request based on the similarities calculated by the different similarity measures.

The model presented in Figure 1 gives the steps of our proposed approach. The client request is sent to the discovery system, and the similarity database will be loaded by the similarity scores computed from the six similarity measures employed. Then, we apply the k-means algorithm to group the WSs into clusters and define the most similar cluster to the client request. This approach aims to increase the accuracy of WSs discovery and avoid the problems encountered with some similarity measures utilizing k-means clustering on the computed similarity scores. These advantages will influence the selection phase and the composition of WSs.

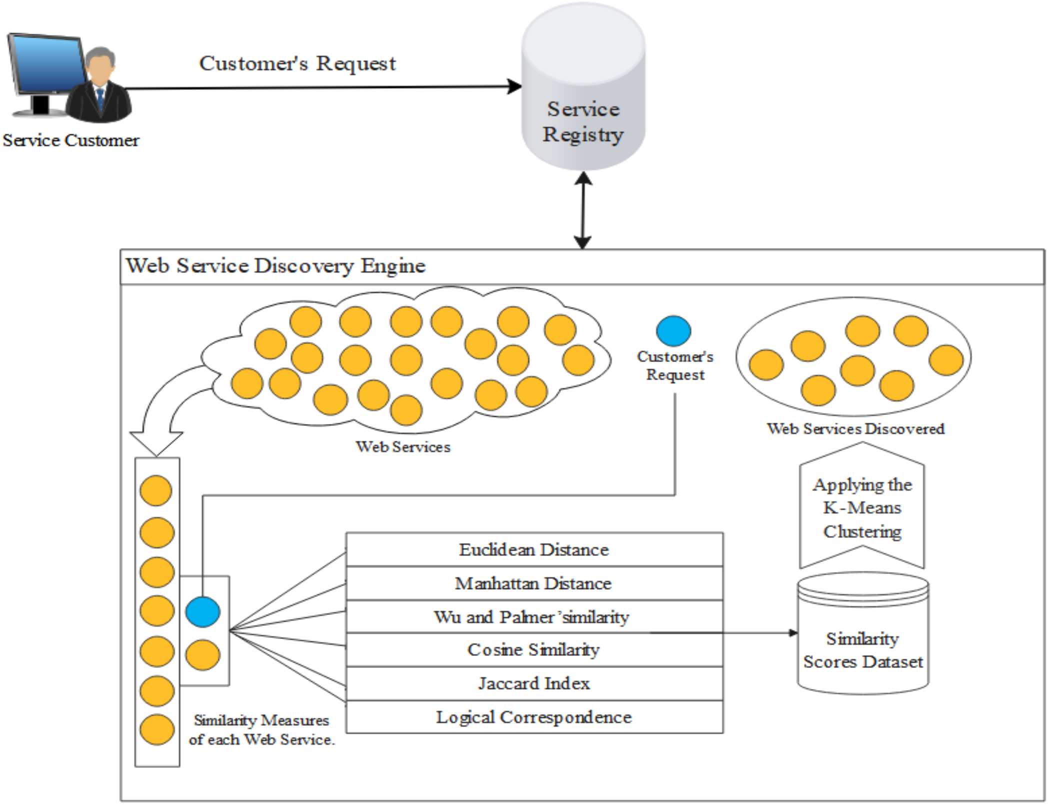


Figure 1. The proposed WS discovery approach

5. RESULTS AND DISCUSSION

5.1. Dataset and experimental setup

The technological environment employed to implement our proposed approach and other mechanisms and its evaluation is the JAVA language. The experimental computations ran on Windows 10 Intel Core i5 CPU (2.6GHz) and 8GB of RAM. Our approach was evaluated experimentally utilizing the service repository of the well-known OWLS-TC v4.0 test collection. There are 1083 SWSs written in OWL-S 1.1 and 42 queries. There are nine service domains: education, food, medicine, travel, economy, communication, weaponry, geography, and simulation. Table 3 shows the details of OWLS-TC v4.0. Some services seem in more than one category. Hence, if we consider just the first occurrence of each service, the number of services is 1083; if we consider repetitions across different categories, the number of services is 1140.

Table 3. Details of OWLS-TC v4.0

Domains Rainfall data	Num. of services	Num. of requests
Education	286	6
Medical Care	73	1
Food	34	1
Travel	197	6
Communication	59	2
Economy	395	12
Weapon	40	1
Geography	60	10
Simulation	16	3

5.2. Evaluation metrics

The performance of the proposed approach is evaluated utilizing tree metrics: precision, recall, and F-measure [38]. These metrics are the most commonly used to evaluate the performance of WS discovery approaches. Precision is the number of accurate results divided by the number of all results returned, while recall is the number of accurate results divided by the number of results to be returned, as presented in (7) and (8).

$$Precicion(P) = CorrectRelevantServicesFound/TotalRelevantServicesFound \quad (7)$$

$$Recall(R) = CorrectRelevantServicesFound/RelevantServicesshouldbeFound \quad (8)$$

There is an inverse correlation between the two equations, where it is possible to increase one value at the cost of reducing the other. In general, they are not treated separately. The F-measure combines precision and recall into a single equation, i.e., a composite harmonic mean, reduces the effects of large unusual values and amplifies the impacts of small ones, as shown in (9). As a weighted unanimous measure, the F-measure is much more critical than either precision or recall separately.

$$F - measure = \frac{2*Precision*Recall}{Precision+Recall} \quad (9)$$

5.3. Results' discussion

First, we performed the experiments with a query of the "car_price_service" dataset from the "Medical Care" domain, and we obtained the results displayed in Table 4. The latter shows the scores calculated by the first step for some control services. All the similarity measures used have different scores for each service, which explains the choice of a perfect similarity measure among the different measures employed. As shown by the "Euclidean distance", the similar service is "BookNonMedicalTransport" with a score of 0.94, but for the "Wu and Palmer" similarity measure, it is "BookMedicalTransport" with a score of 0.99.

We used the k-means method to determine the services with the best similar scores to remove this variation, using the different scores of the similarity measures. Nevertheless, before utilizing the k-means method, we must define the number of K classes utilizing the elbow method. Figure 2 indicates that the actual number of clusters is K=4.

The WSs will be classified into four classes. By calculating the centroids of each class, we achieve the results displayed in Table 5, which represents the centroids of each class and the number of WSs. The discovered WSs are the services in the class whose centroid is closest to 1. According to our approach, we can assume that the most similar class of WSs is class 3, which contains 72 WSs and gives a 98.63% precision depending on the studied domain.

Table 4. Details similarity scores for WSs discovered from the medical care domain

Service name	Euclidean distance	Manhattan distance	Wu and Palmer'similarity Measure	Cosine similarity	The Jaccard index	Logical Correspondence
BookNonMedicalTransport	0.94	0.86	0.90	0.94	0.79	1
hiking_destination	0.73	0.54	0.59	0.89	0.58	0.4
getAddressOfLocation	0.30	0.36	0.00	0.01	0.27	0
BookMedicalTransport	0.86	0.84	0.99	0.86	0.87	0.6
auto_pricecolor	0.76	0.63	0.58	0.83	0.84	0.6

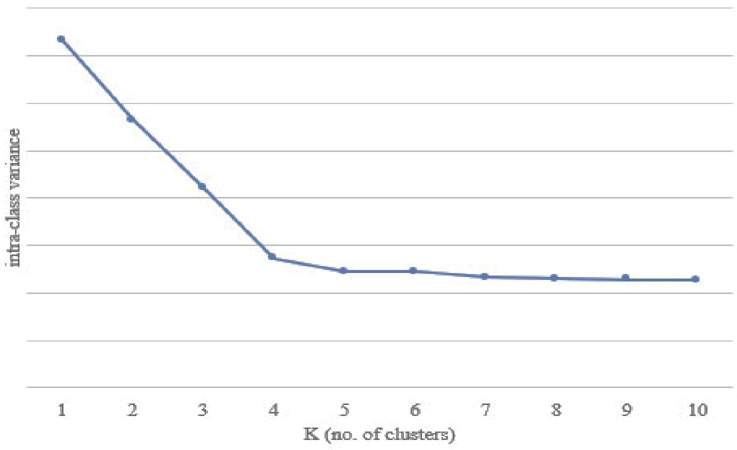
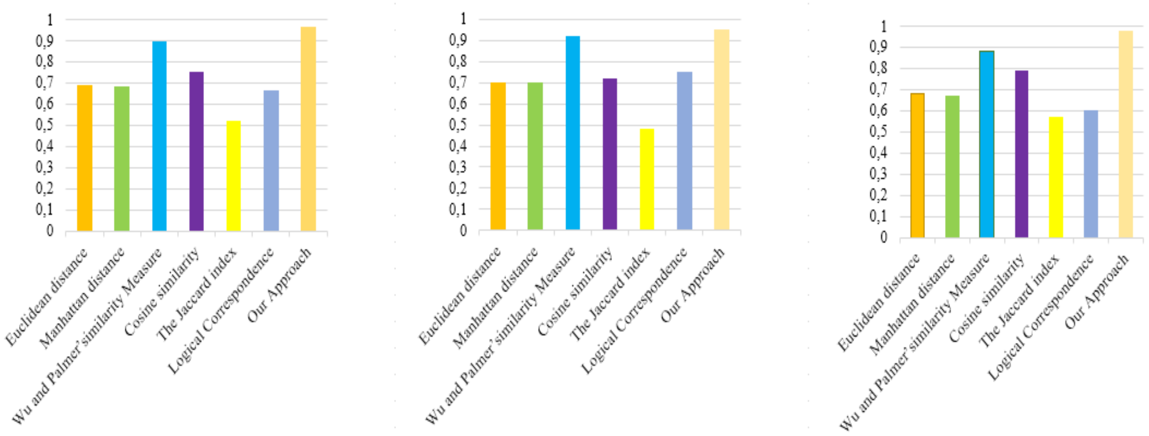


Figure 2. The elbow method showing the optimal K

Table 5. The Centroids of Classes Obtained

Class	Variance intra-class	Number of services	Central service
1	0.433	278	title_comedyfilm
2	0.574	489	auto_technology
3	**0.756**	**72**	**hospital_biopsy**
4	0.293	246	governmentmissileweapon_funding

It was tested for several queries and WSs to evaluate our approach, and its results were compared to the other similarity measures. Experiments presented in Figure 3 show that our approach is more performed than other methods for WS discovery. To analyze these results, we can say that the other similarity measures calculate the similarity between the request and the WSs. However, our approach benefits from their results to build a more real service similarity value. Most works in WS discovery used the similarity measures already mentioned to show the efficiency and effectiveness of our approach. We compared the results obtained with the other similarity measures studied; the results show the success of our approach by increasing the precision (98%) and recall (95%), as well as the F-Measure (96%) of the WSs discovered, compared to the approaches that use the different similarity measures. This success is due to using the k-means algorithm on the similarity scores obtained by many similarity measures, then the elimination of possible disadvantages for any measure. Although, decrease the number of WSs directed at the WS selection phase.



(a) (b) (c)

Figure 3. Discovery results of six similarity measures with our approach; (a) F-measure, (b) recall, (c) precision

On the other hand, to validate our results against other approaches in the literature. We implemented two methods previously mentioned in the related works section [29], [30]. For the approach proposed in [29], we utilized the best interval judged by the authors. Figure 4 shows a comparison of precision, recall, and F-measure for the mentioned methods. The obtained outcomes confirm the effectiveness of our approach compared to the studied approaches.

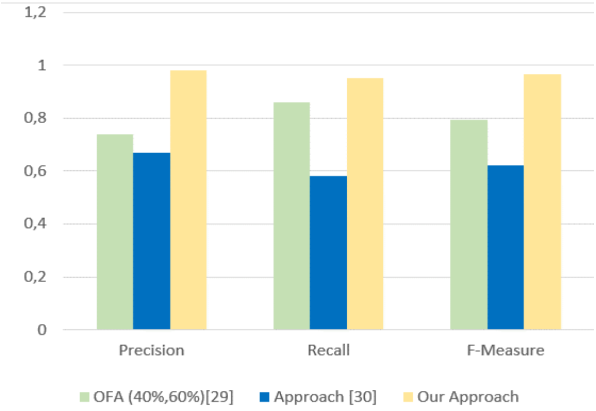


Figure 4. Evaluation metrics for our approach

6. CONCLUSION

To improve the discovery process of SWSs, we used an OWL-S and six similarity measures, namely Euclidean distance, Manhattan distance, Wu and Palmer, Cosine similarity, Jaccard index, and logical matching, between the WSs stored in the registry and the consumer's request. Then, we apply the k-means clustering algorithm in order to obtain the most similar services to the consumer's request, not based on a single similarity measure (the usual case of the Euclidean distance) but based on the six similarity measures used. The carried out experiments have shown the high performance of this approach in terms of precision (98%) and recall (95%), as well as the F-Measure (96%). We assume that a practical WSs discovery approach can help customers make their distributed applications more efficient. Moreover, other similarity measures can enrich the proposed approach depending on the customers' needs or the expected efficiency.

REFERENCES

[1] N. Balaji, S. R. Murugaiyan, M. S. Saleembasha, R. Baskaran, and P. Dhavachelvan, "Enhancements for UDDI using User Preferential Web Service Selection Model based on SLA," *Int. J. Eng. Technol.*, vol. 5, no. 5, pp. 4057-4067, 2013.

[2] C. Dhasarathan, V. Thirumal, and D. Ponnurangam, "Data privacy breach prevention framework for the cloud service," *Secur. Commun. Networks*, vol. 8, pp. 982-1005, 2015, doi: 10.1002/sec.1054.

[3] P. Khutade and R. Phalnikar, "QoS based web service discovery using oo concepts," *Int. J. Adv. Technol. Eng. Res.*, vol. 2, no. 6, pp. 81-86, 2012.

[4] M. Fariss, N. El Allali, H. Asaidi, and M. Bellouki, "Review of Ontology Based Approaches for Web Service Discovery," *International Conference on Advanced Information Technology, Services and Systems*, pp. 78-87, 2019, doi: 10.1007/978-3-030-11914-0_8.

[5] L. Li and I. Horrocks, "A Software Framework for Matchmaking Based on Semantic Web Technology," *Int. J. Electron. Commer.*, vol. 8, no. 4, pp. 39-60, 2004, doi: 10.1080/10864415.2004.11044307.

[6] S. Y. Lin, C. H. Lai, C. H. Wu, and C. C. Lo, "A trustworthy QoS-based collaborative filtering approach for web service discovery," *J. Syst. Softw.*, vol. 93, pp. 217-228, 2014, doi: 10.1016/j.jss.2014.01.036.

[7] P. Plebani and B. Pernici, "URBE: Web Service Retrieval Based on Similarity Evaluation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 11, pp. 1629-1642, Nov. 2009, doi: 10.1109/TKDE.2009.35.

[8] G. Cassar, P. Barnaghi and K. Moessner, "Probabilistic Matchmaking Methods for Automated Service Discovery," *IEEE Transactions on Services Computing*, vol. 7, no. 4, pp. 654-666, Oct.-Dec. 2014, doi: 10.1109/TSC.2013.28.

[9] M. Klusch, B. Fries, and K. Sycara, "OWLS-MX: A hybrid Semantic Web service matchmaker for OWL-S services," *J. Web Semant.*, vol. 7, no. 2, pp. 121–133, 2009, doi: 10.1016/j.websem.2008.10.001.

[10] M. Klusch and P. Kapahnke, "OWLS-MX3: An adaptive hybrid semantic service matchmaker for OWL-S," in *Third International Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web (SMR2)*, vol. 525, 2009.

[11] W. Lu, Y. Cai, X. Che, and Y. Lu, "Joint semantic similarity assessment with raw corpus and structured ontology for semantic-oriented service discovery," *Personal and Ubiquitous Computing*, vol. 20, no. 3, pp. 311-323, 2016, doi: 10.1007/s00779-016-0921-0.

[12] M. Fang, D. Wang, Z. Mi, and M. S. Obaidat, "Web service discovery utilizing logical reasoning and semantic similarity," *Int. J. Commun. Syst.*, vol. 31, no. 10, pp. 1-13, 2018, doi: 10.1002/dac.3561.

[13] R. A. H. M. Rupasingha, I. Paik, and B. T. G. S. Kumara, "Calculating web service similarity using ontology learning with machine learning," *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, 2015, pp. 1-8, doi: 10.1109/ICCIC.2015.7435686.

[14] H. Frigui, "Clustering: Algorithms and Applications," *2008 First Workshops on Image Processing Theory, Tools and Applications*, 2008, pp. 1-11, doi: 10.1109/IPTA.2008.4743793.

[15] N. Arunachalam, A. Amuthan, C. Kavya, M. Sharmilla, K. Ushanandhini and M. Shanmughapriya, "A survey on web service clustering," *2017 International Conference on Computation of Power, Energy Information and Commuincation (ICCPEIC)*, 2017, pp. 247-252, doi: 10.1109/ICCPEIC.2017.8290371.

[16] S. Chandrasekaran, V. B. Srinivasan, and L. Parthiban, "Efficient Web Service Discovery and Selection Model," *Int. J. Futur. Revolut. Comput. Sci. Commun. Eng.*, vol. 4, no. 2, 2018.

[17] K. Venkatachalam and N. K. Karthikeyan, "Effective Feature Set Selection and Centroid Classifier Algorithm for Web Services Discovery," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 5, no. 2, pp. 441-450, Feb. 2017, doi: 10.11591/ijeecs.v5.i2.pp441-450.

[18] Z. Wu and M. Palmer, "Verb Semantics and Lexical," *32nd Annu. Meet. Assoc. Comput. Linguist.*, 1994, pp. 133-138, doi: 10.3115/981732.981751.

[19] J. Wu and Z. Wu, "Similarity-based Web service matchmaking," *2005 IEEE International Conference on Services Computing (SCC'05) Vol-1*, vol. 1, 2005, pp. 287-294, doi: 10.1109/SCC.2005.93.

[20] F. Chen, C. Lu, H. Wu, and M. Li, "A semantic similarity measure integrating multiple conceptual relationships for web service discovery," *Expert Syst. Appl.*, vol. 67, pp. 19-31, 2017, doi: 10.1016/j.eswa.2016.09.028.

[21] C. Corley and R. Mihalcea, "Measuring the semantic similarity of texts," in *Proceedings of the ACL workshop on empirical modeling of semantic equivalence and entailment*, 2005, pp. 13-18, doi: 10.3115/1631862.1631865.

[22] H. Fethallah, M. Mohammed, and B. Amine, "Semantic Web service Discovery Based on Fuzzy Dominated Scores," in *Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication*, 2015, pp. 1-6, doi: 10.1145/2816839.2816881.

[23] N. Kokash, "A Comparison of Web Service Interface Similarity Measures," *Proceedings of the 2006 conference on STAIRS 2006: Proceedings of the Third Starting AI Researchers' Symposium*, 2006, pp. 220-231.

[24] M. Montague and J. A. Aslam, "Condorcet fusion for improved retrieval," *Proceedings of the eleventh international conference on Information and knowledge management*, 2002, pp. 538–548, doi: 10.1145/584879.584881.

[25] M. Farah and D. Vanderpooten, "An outranking approach for information retrieval," *Information Retrieval*, vol. 11, no. 4, pp. 315-334, 2008, doi: 10.1007/s10791-008-9046-z.

[26] H. Fethallah, C. Amine, and B. Amine, "Automated Discovery of Web Services : an Interface Matching Approach Based on Similarity Measure," in *Proceedings of the 1st International Conference on Intelligent Semantic Web-Services and Applications*, 2010, pp. 1-4, doi: 10.1145/1874590.1874603.

[27] J. Wu, L. Chen, Z. Zheng, M. R. Lyu, and Z. Wu, "Clustering Web services to facilitate service discovery," *Knowl. Inf. Syst.*, vol. 38, no. 1, pp. 207-229, 2014, doi: 10.1007/s10115-013-0623-0.

[28] A. De Renzis, M. Garriga, A. Flores, A. Cechich, and A. Zunino, "Case-based Reasoning for Web Service Discovery and Selection," *Electronic Notes in Theoretical Computer Science*, vol. 321, 2016, doi: 10.1016/j.entcs.2016.02.006.

[29] H. Fethallah, S. M. Ismail, M. Mohamed and T. Zeyneb, "An outranking model for web service discovery," *2017 International Conference on Mathematics and Information Technology (ICMIT)*, 2017, pp. 162-167, doi: 10.1109/MATHIT.2017.8259711.

[30] A. Fellah, M. Malki, and A. Elci, "A similarity measure across ontologies for Web services discovery," *International Journal of Information Technology and Web Engineering*, vol. 11, no. 1, pp. 22-43, 2016.

[31] J. Han, M. Kamber, J. Pei, *Data mining concepts and techniques*. San Francisco, USA: Moraga Kaufman, 2011.

[32] A. H. Nasution, Y. Murakami, and T. Ishida, "Generating similarity cluster of Indonesian languages with semi-supervised clustering," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 1, pp. 531-538, 2019, doi: 10.11591/ijece.v9i1.pp531-538.

[33] P. M. Prihatini, I. K. G. D. Putra, I. A. D. Giriantari, and M. Sudarma, "Complete agglomerative hierarchy document's clustering based on fuzzy Luhn's gibbs latent dirichlet allocation," *International Journal of Electrical and Computer Engineering*, vol. 9, no. 3, pp. 2103-2111, 2019, doi: 10.11591/ijece.v9i3.pp2103-2111.

[34] D. S. Maylawati, T. Priatna, H. Sugilar, and M. A. Ramdhani, "Data science for digital culture improvement in higher education using K-means clustering and text analytics," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 5, pp. 4569-4580, 2020, doi: 10.11591/ijece.v10i5.pp4569-4580.

[35] L. Muflikhah and B. Baharudin, "Document Clustering Using Concept Space and Cosine Similarity Measurement," *2009 International Conference on Computer Technology and Development*, 2009, pp. 58-62, doi: 10.1109/ICCTD.2009.206.

[36] W. B. Zulfikar, M. Irfan, M. Ghufron, Jumadi, and E. Firmansyah, "Marketplace affiliates potential analysis using cosine similarity and vision-based page segmentation," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 6, pp. 2492-2498, 2020, doi: 10.11591/eei.v9i6.2018.

[37] P. D. Ibnugraha, L. E. Nugroho, and P. I. Santosa, "An approach for risk estimation in information security using text mining and jaccard method," *Bulletin of Electrical Engineering and Informatics*, vol. 7, no. 3, pp. 393-399, 2018, doi: 10.11591/eei.v7i3.847.

[38] A. Althaf Ali and R. M. Shafi, "Test-retrieval framework: Performance profiling and testing web search engine on non factoid queries," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 14, no. 3, pp. 1373-1381, 2019, doi: 10.11591/ijeecs.v14.i3.pp1373-1381.