REVIEW ARTICLE

Clustering and Association Rules for Web Service Discovery and Recommendation: A Systematic Literature Review

Waeal J. Obidallah[1,3] [ORCID] **· Bijan Raahemi[1,2] · Umar Ruhi[2]**

Abstract

The purpose of this study is to identify, summarize, and systematically compare various clustering and association rule techniques for web service discovery and recommendation, identify the most common data sets used in the extant literature, and highlight current trends and future research directions. Following the methodology of Kitchenham and Charters (Guidelines for performing Systematic Literature reviews in Software Engineering, 2007) for a systematic literature review (SLR), a set of research questions are designed. Six digital databases are searched. A total of 4581 papers were initially retrieved, and a rigorous two-stage scanning process resulted in 66 relevant papers. Based on the selection criteria and data extraction, 57 final studies were selected. These papers are summarized and compared, and the relevant information is extracted to answer the research questions. The synthesis resulted in knowledge of currently proposed methods for web service discovery and recommendation based on clustering and association rule techniques. Furthermore, it identifies algorithms, similarity measures, evaluation metrics, and data sets. Also identifies challenges, research gaps, trends, and future directions. We propose a classification of web service discovery and recommendation methods and map the 57 final selected papers into these classes. This review will help researchers to understand the current state-of-the-art in clustering and association rules techniques for web service discovery and recommendation, and also recognize trends and future directions for improvement. Future studies should broaden the basis of discovery and recommendation by including various types of web service descriptions including plain text that are currently used in web APIs. An opportunity for improvement by utilizing modern techniques based on big data analytics and social network analysis.

Keywords Systematic literature review · Web service discovery · Web service recommendation · Clustering · Association rules

Introduction

Web services constitute the foundational building blocks of service-oriented architectures (SOAs) for the creation of software applications in a distributed computing paradigm [2]. Web services are a set of loosely coupled software components that are developed, described, published, discovered, used, and reused to achieve explicit functionalities [3]. There are different types of web services, which provide different functionalities over the Internet. They include SOAP-based, REST-based, XML-RBC-based, and JSON-RBC-based web services. The most popular forms are REST-based and SOAP-based web services. Recently, enterprises leverage their assets by widely publishing their web services or application programming interfaces (APIs), allowing scaling of their functionalities over the Internet and the use of different business models. With the growing number of web services,

✉ Waeal J. Obidallah
  wobidall@uottawa.ca

  Bijan Raahemi
  braahemi@uottawa.ca

  Umar Ruhi
  umar.ruhi@uottawa.ca

1 Knowledge Discovery and Data Mining Lab, University of Ottawa, Ottawa, Canada

2 Telfer School of Management, University of Ottawa, Ottawa, Canada

3 College of Computer and Information Sciences, Imam Muhammad Ibn Saud Islamic University, Riyadh, Saudi Arabia

the demand for discovery and recommendation mechanisms for web services has also increased.

Web service discovery is the process of finding and locating existing web services based on the service requester's requirements. It must match services' functional and non-functional descriptions [4, 5], as well as retrieve web service descriptions (e.g., WSDL and WADL) published by a service provider. The functional requirements refer to the elements that indicate the system's capabilities. Functional characteristics of web services include interfaces, operations, and protocol bindings. These are typically described in the service profile [5], [6]. Non-functional requirements refer to the elements that indicate the system's performance parameters, which may include quality of service (QoS) considerations, and service policies (e.g., security features and service cost) [5, 6].

Web service recommendation systems assist web service consumers in the process of web service selection where QoS parameters play a significant role in ranking web services [7]. Because many web services have similar functionalities, the service requester must perform service selection without prior knowledge regarding the list of web service candidates. Web service recommendation systems provide leverage by suggesting and ranking web services to help service requesters during the selection process. Web service recommendation systems have been used to improve web service discovery systems in different studies.

Data mining and machine learning techniques have been used to improve the process of web service discovery and recommendation. More specifically, clustering and association rule techniques are used to boost the discovery and recommendation of web services. Clustering algorithms are used to cluster web services into different clusters to minimize the search space of web services based on various similarity measures. Furthermore, association rule algorithms are used to find the relationships and correlations between services and users to build a recommendation model.

This SLR aims to identify, explore, summarize, and synthesize clustering and association rule techniques used for web service discovery and recommendation. The final objective is to provide the reader with the current state of clustering and association rule techniques utilized in the process of web service discovery and recommendation by identifying the methods, algorithms, similarity measures, data sets, and evaluation metrics. Furthermore, this SLR aims to understand and demonstrate whether there are algorithms, data sets, and similarity measures that are used much more frequently compared to others. The motivation and the need to conduct this SLR are explained as follows:

- The discovery problem is inherently difficult because of the large scale of SOA systems. This includes web services, which are dynamic, changing, and uncertain in

nature. The scope of services in SOA changes rapidly as services are added and removed or modified. Developers abandon older standards as new standards emerge. Web service discovery and recommendation are critical to the success of web services, because such services are useless if users cannot discover them;

- The use of web services in the industry is gaining attention. This allows developers to efficiently use, discover, and select web services through discovery and recommendation systems. Thus, it is critical to identify techniques that facilitate web service discovery and recommendation;

- Data mining, text mining, and machine learning techniques provide solutions to facilitate the process of web service discovery and recommendation where clustering, classification, and association rule algorithms have been applied. The focus of clustering and association rules is based on the identification of algorithms, similarity measures, data sets, and evaluation metrics. These favor the building of practical web service discovery and recommendation systems. This helps future researchers to make sense of the web service discovery and recommendation landscape regarding the use of clustering and association rules;

- Over the past decade, several clustering and association rule techniques for web services have been proposed to facilitate the process of web service discovery and recommendation. Thus, there is a need to investigate the characteristics and details of such techniques using a systematic method and quality requirements; and

- Data mining and analytics is a quickly evolving field, and this SLR provides useful information and synthesis regarding the current state of web service discovery and recommendation research using clustering and association rule techniques, particularly due to the lack of comprehensive SLR in this area.

The remainder of the paper is organized as follows. Section 2 provides background regarding web services and the need for discovery and recommendation. In Sect. 3, we describe the research method indicating the search selection, data extraction, quality assessment, and evaluation strategies and the SLR phases. In Sect. 4, we report the results of this study by analyzing the final selected papers and providing answers to the research questions. Section 5 presents our discussion and observations. Section 6 discusses possible threats to the validity of this SLR. Finally, we conclude the paper in Sect. 7 by summarizing some of the outcomes of this study.

Background

web services are enabling technologies for E-Business that provide fast, efficient, and reliable services to customers over the Internet [8]. Web services are self-contained, modular applications with business functionality and logic that are described, published, located, discovered, and invoked over a network on the World Wide Web. Most web services are described using the standard WSDL or according to the REST paradigm [5]. However, service providers publish their web services' functional descriptions in different languages and formats. According to [5], seekda.com, which was a web portal for publishing web services, reported 30 k web services in November 2011. It is also reported by the public directory website programmableweb.com (PW) that it has approximately 16 k single or composite RESTful web services as of March 2013 [5]. The numbers of web services rapidly increased the number of places where web service providers publish their services, including public directories, web portals, and API Marketplaces, in various languages and formats.

The process of locating existing web services based on the service requester's functional and non-functional requirements [4, 5] and retrieving the web service descriptions previously published by a service provider is known as web service discovery. The standard discovery mechanism for SOAP-based web services is the use of UDDI, which is an XML-based registry that provides service providers with the ability to publish and register their web services by providing their specifications. UDDI is used to provide web services with a way to be discovered by service consumers based on keyword matching [9]. There is no discovery standard for REST-based web services. The discovery process starts with web service providers advertising their web services in web service repositories by publishing their web service description (e.g., a WSDL file). In the web service data layer [10], service providers share their service data (web service descriptions and specifications, which can be in different formats) to be stored for any inquiry from service users. Services' potential users send a request to query (keyword-based if UDDI) the repositories at the web service data layer by specifying their web service requirements. The service matcher matches the request with the available web services and recommends a set of web services that match the requester's needs. The final step is the selection and invocation of the best web service that meets the user's needs.

Specialized search engines, including Titan, Seekda, and Woogle,[7] provide an advanced solution to web service discovery. However, most web service search engines rely on keyword-based matching. Therefore, the search may suffer from a lack of keywords in the description. Web service files do not contain an adequate number of words for index terms

or features. Moreover, the small numbers in the web service files are erratic and unreliable [11]. The limited number of functionalities offered by web services can lead to insufficient terms in descriptions, which differs from regular text documents. Furthermore, a huge and irrelevant number of web services will be returned due to a broad search space. Clustering web services based on similarities can minimize the search space [12].

[1]Approach to web service recommendation help in understanding the user's requirements when selecting a web service. Although web service recommendation is challenging and time-consuming due to large search spaces, clustering techniques can minimize this space [13]. QoS information plays a major role in web service recommendation, and when it is limited, web service recommendation approaches usually either fail to make predictions or make inferior ones. However, QoS information is hard to obtain, because users often request very few of the available web services [14]. Furthermore, it is impractical, costly, and time-consuming for users to try all candidate web services and acquire QoS information [14–16]. Due to user disparity, QoS values evaluated by one user cannot be directly employed by another for web service recommendation. Therefore, it is necessary to either identify associations between users or cluster users based on their similarities.

Web service technology is an efficient and effective way to deliver services to users in e-business and e-commerce. Companies across the globe use web services; they develop and publish such services to leverage enterprise business models. However, the web service concept is unfamiliar to end users who continue to struggle in discovering and selecting the appropriate web services. End users need advanced mechanisms to assist in the discovery and selection of services. This has led many researchers to propose solutions to enhance web service discovery and recommendation. Data analytics, text mining, information retrieval, and machine learning techniques provide solutions to facilitate the process of web service discovery and recommendation, and clustering, classification, and association rule algorithms have been applied in the field of web service discovery and recommendation.

Research Methodology

We follow general guidelines and procedures described in [1, 17]. The three phases of SLR are (1) planning, (2) conducting, and (3) reporting phases, as illustrated in Fig. 1. The planning phase involves (1) identification of the need for SLR; (2) definition of research questions; (3) development of the review
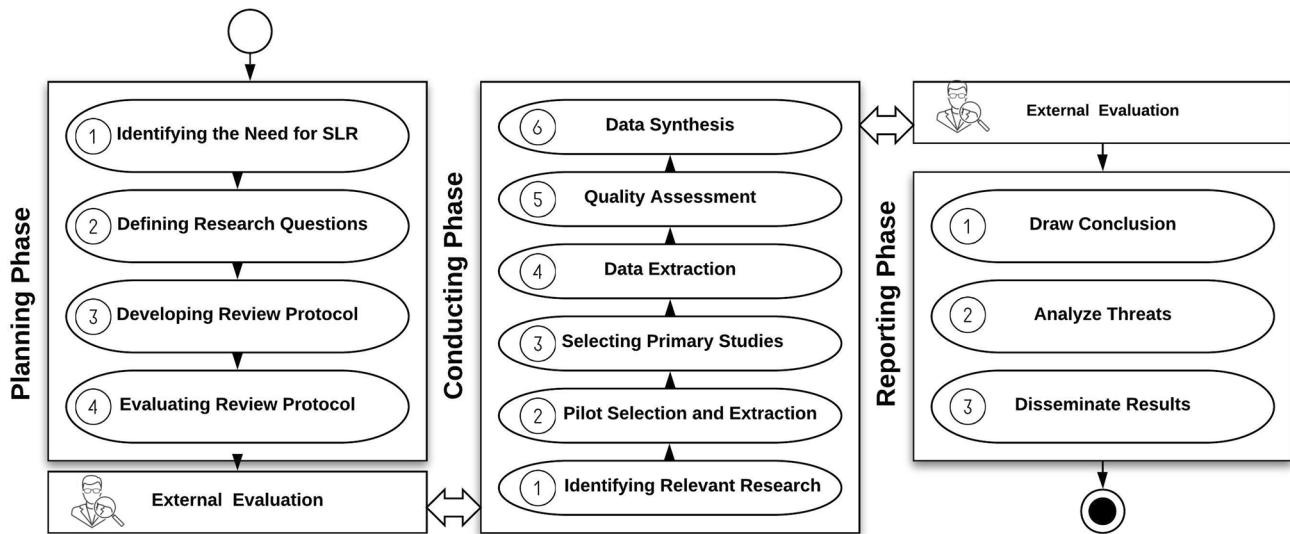
---

1 [7] http://db.cs.washington.edu/projects/woogle.html.

Fig. 1 Overall research methodology

Table 1 PICOC criteria to define scope, goals, and strings

Criteria	Describe/reflect
Population	Web services, web service discovery, web service recommendations
Intervention	Clustering, association rules
Comparison	Comparison of methods, algorithms, data sets, and evolution metrics
Outcome	Classification and comparison, trends, and future research directions
Context	Systematic investigation to consolidate peer-reviewed and academic research

protocol; and (4) evaluation of the review protocol. Three research questions are identified following criteria acknowledged by [1]. Based on search questions and search strings, six digital databases are searched. The results retrieved 4581 papers, which included several formats (long, short, and poster papers) and duplicates. A rigorous two-stage scanning and filtering process was followed according to predefined criteria. Using the Covidence online Web-based tool [18], 66 papers were selected based on the two screening stages. We conducted a manual search based on snowballing of 66 selected papers, which lead to the identification of two papers after applying the selection criteria. The 68 primary selected papers went through a quality assessment as suggested by [1, 19, 20]. We used the quality assessment to weight the relevance and the importance of studies to our research questions. The final activity identified 57 papers as the final selections. We collected the required information to answer the research questions, as well as analyze and summarize the results.

The Planning Phase

The planning phase identifies the research questions, search strategy, selection process, quality assessment, and data extraction. It also reviews protocol evaluation.

Research Question Definitions

We considered research questions from the viewpoints of population, intervention, comparison, outcome, and context (PICOC), as outlined in Table 1. The general goal and scope of the study were formulated through PICOC. We used population and intervention in search terms and keywords. The research questions addressed include:

RQ1   What techniques, methods, and algorithms are used in clustering and association rules for web service discovery and recommendation?

RQ2   What are the most common data sets used to validate the proposed clustering and association rule approaches to facilitate web service discovery and recommendation?

RQ3   What are the trends and future research directions related to the discovery and recommendation of web services?

RQ1 identifies:

- Techniques used in the investigated studies, including clustering based on web service discovery/recommen-

Table 2 Research questions and motivation

RQ	Research question	Motivation
RQ1	What clustering and association rule techniques, methods, and algorithms are used for web service discovery and recommendation?	Identify and classify existing techniques Identify and compare proposed methods with its similarity measures and evaluation metrics Determine and compare the used algorithms
RQ2	What are the most common data sets used to validate the proposed clustering and association rule approaches to facilitate web service discovery and recommendation?	Identify discovery benchmark data set collections and their locations Identify recommendation benchmark data set collections and their locations Determine the type, purpose, and method of collection of designated data sets
RQ3	What are the trends and future research directions related to the discovery and recommendation of web services?	Understand and disclose challenges Identify research gaps Identify trends and future directions

Table 3 Digital libraries selected and initially retrieved results

	Database name	URLs	Initially retrieved
1	ACM Digital Library	http://www.dl.acm.org/	637
2	Science Direct	http://www.sciencedirect.com/	62
3	Scopus	https://www.scopus.com/	2196
4	IEEE Xplore	http://ieeexplore.ieee.org/	342
5	Web of Science	https://webofknowledge.com/	71
6	Springer Link	http://link.springer.com/	1273
Total number of papers			4581

dation or association rule-based web service discovery/recommendation;

- The proposed method to facilitate the process of web service discovery and recommendation. Clustering and association rule techniques are used in the proposed method and supported by other techniques, such as natural language processing (NLP) and information retrieval (IR). Methods include the proposed processes/framework used to facilitate web service discovery and recommendation, similarity measures, and evaluation matrices to validate the work; and

- Clustering and association rule algorithms to support the object of the study. For example, in clustering, k-means, hierarchical, and density-based clustering algorithm (DBSCAN) algorithms are employed. In association rules, the Progressive Size Working Set (PSWS) and Apriori algorithms are used.

A summary of the research questions and their motivations is listed in Table 2.

Search Strategy

To avoid overlooking relevant papers, we used a generic search string, which included a broad number of articles in its initial results. We identified the search terms and keywords by following the five criteria in [1]. Based on the PICOC, we used search terms and keywords to construct search strings as follows:

$(P_1 \text{ OR } P_2 \dots \text{OR } P_n) \text{ AND } (I_1 \text{ OR } I_2 \dots \text{OR } I_n)$

P_n: population terms, I_n: intervention terms.

Different spellings were considered when constructing the search string using an asterisk (*) and the Boolean operators AND and OR. When databases allowed, the advanced search option inserted the complete search string. To extend the scope of the results, we considered the population and intervention in the search string. We included the comparison in the primary study selection process by filtering studies that did not contain methods, algorithms, data sets, and evaluation metrics. This compensated for the exclusion of the comparison in the search strings. The search string was as follows:

("Web Service*" OR "Web Service* Discovery" OR "Web Service* Recommend*")

AND

("Clustering" OR "Association Rule*").

We used search strings to search six digital libraries, as listed in Table 3 along with the number of initially retrieved papers from each digital library. Search strings were adjusted according to the requirements of the selected database search engines. Prior to running a full search, the search string's effectiveness was examined as recommended by [1] using a list of known publications [12, 21, 22]. The search, which was conducted on 1st of February 2018, was limited to
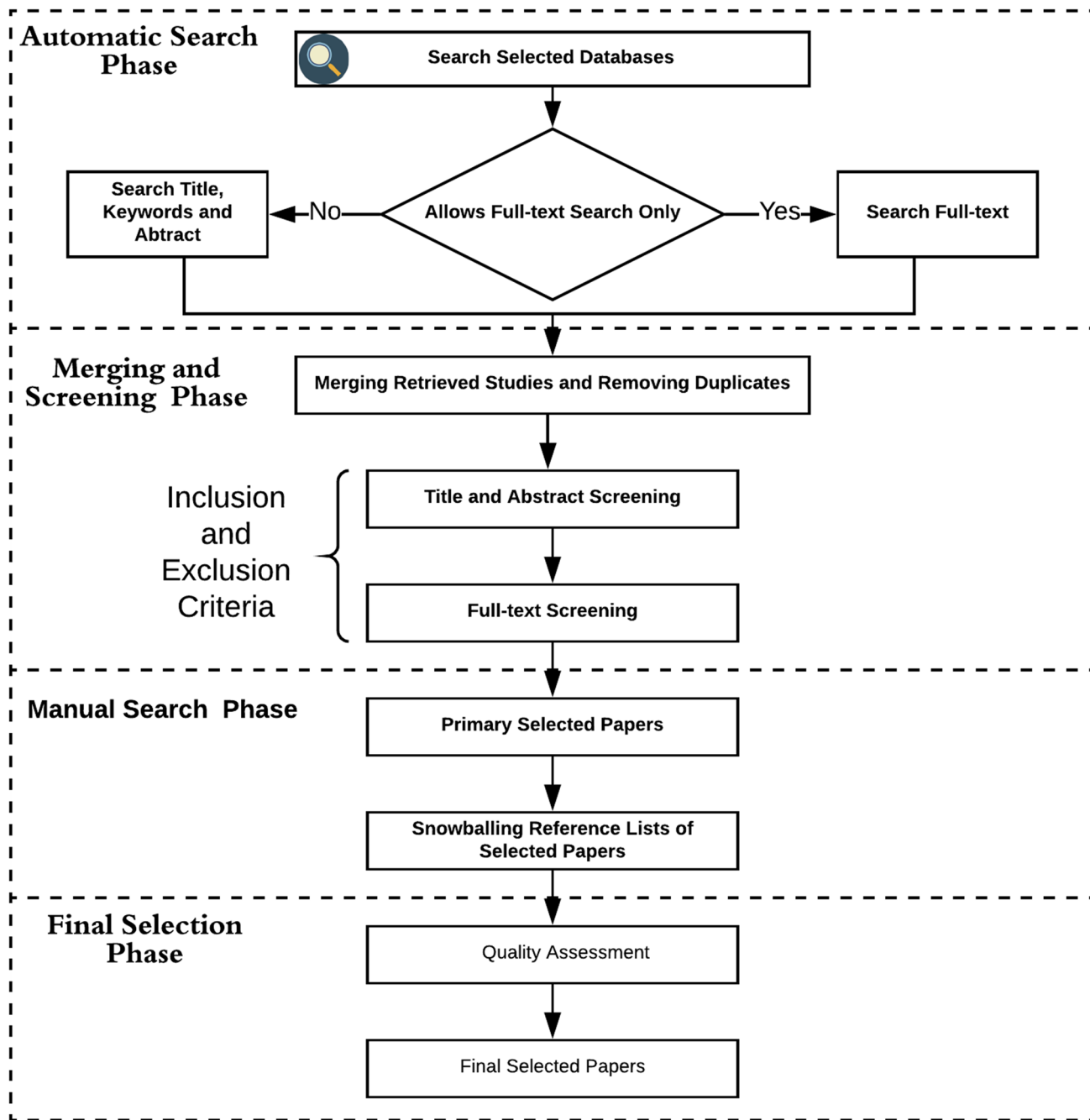
Fig. 2 Search and selection process

studies published between 2006 and January 2018. We also conducted a manual search as recommended by [1].

Study Selection Process

A four-phase search and selection process was used to search the digital libraries, filter the results, and collect and screen relevant papers as illustrated in Fig. 2. In the first phase, an automatic search of the digital libraries was performed using search strings. In the second phase, all retrieved studies were merged after duplicates were removed. Two screening stages were performed. Title and abstract screenings were applied when the inclusion and exclusion criteria (Table 4) were checked against the titles and abstracts of the retrieved

Table 4 Inclusion and exclusion criteria

Criteria		Rationale
Inclusion	I1: Studies in the form of academic peer-reviewed conference and journal articles	Peer-reviewed papers guarantee a level of quality and the well-trusted volume of content
	I2: Studies published between 2006 and January 2018	11 years has been selected as the SLR period. This decision was made based on the start time (2006) that UBRs closed and alternative web service discovery and recommendation solutions emerged
	I3: Studies with proposed solution and evaluation/validation of web service discovery and recommendation	We are looking for methods, algorithms, data sets and evaluation metrics using clustering and association rules in web service discovery and recommendation
Exclusion	E1: Studies in the form of abstracts, posters, short papers (less than six pages), technical reports, tutorial summary, or books	These studies do not usually provide the information needed for this review
	E2: Studies that are not in the English language	Although there are different languages, we excluded non-English studies due to language limitations
	E3: Studies were keywords (web service discovery or clustering or association rules or web service recommendations) were not addressed explicitly in the article	We are investigating studies which are related to the research questions. These keywords are needed for screening the primary selected studies
	E4: Studies that do not include their methods, data sets or validation procedures	Studies will be eliminated if they do not consider the full-text scanning phase. This is because they will not relate to the research questions

studies. The full-text screening was applied when the inclusion and exclusion criteria were checked against the full texts of the initially selected studies. In the third phase, a manual search was performed by snowballing from reference lists of the primary selected studies as recommended by [1]. In the final phase, we identified the quality of selected papers based on quality assessment criteria. The final selected papers were identified based on the quality/relevance score.

Quality Assessment of the Primary Selected Studies

We used a quality assessment method based on the questions designed to assess the relevance of the selected papers. This minimizes bias and maximizes validity [1, 19]. The quality assessment was a means for weighting the importance and relevance of studies during the synthesis stage of this SLR. It also supported the validity of the selected papers in this review. Based on the quality assessment questions, a numerical quantification (quality score) is assigned to rank the selected papers. This was done to help with the selection of the most related studies. It was not used as an absolute measure of a paper's quality, nor to compare the papers and the work of authors against each other. It actually indicates the relevance of the selected papers to our research questions. Based on the recommendations of [19, 20], we divided the quality questions to assess general questions (GQ) and specific questions (SQ) as outlined in Table 5. The GQs focused on the quality of reporting, including the papers' rationales, aims, and context. The SQs concentrated on the technical rigor and credibility of the papers.

To assess quality, each paper was evaluated against GQ and SQ quality assessment questions. The answers to the questions could be "Yes," "Partly," or "No." Numerical values were assigned to the answers (1 = "Yes," 0 = "No," and 0.5 = "Partly"). The final quality score for each primarily selected paper was calculated by adding up the scores of each question. The maximum score was 9; scores of 7.5–9 represented high-quality (high-relevance) papers, scores less than 7.5 and greater than or equal to 5.5 were acceptable with average quality (relevance), and scores less than 5.5 were low quality (relevance) and resulted in exclusion. Some of the primarily selected papers were recent publications. Therefore, they were not ranked based on the number of citations.

Data Extraction

This involved collecting data and information relevant to the RQs. We designed a data extraction form as outlined in Table 6. The test–retest process [1] was used to check the consistency and accuracy of the extracted data relative to the original sources. This was performed during the evaluation stage.

Table 5 Quality assessment questions

	Score		
	Yes	Partly	No
	1	0.5	0

General quality assessment questions (GQ)

GQ1   *(Motivation and Problem)* Do the authors explain the aims and objectives of the primary study? Is there a strong and clear rationale for why the study is undertaken?

GQ2   *(Research Environment)* Is the context in which the research was carried out adequately described?

GQ3   Are the contributions of the study well presented and in line with the result?

GQ4   Are the research methodology and design clearly stated?

Specific quality assessment questions (SQ)

SQ1   Is the study clearly focused on web service discovery or web service recommendation?

SQ2   Are the techniques, methods, and algorithms in the study clearly described and linked to their usefulness? (RQ1)

SQ3   Does it identify the data set information (name, location, etc.) for validation? (RQ2)

SQ4   Is there more than one evaluation metric used in the study for validation? (RQ2)

SQ5   Are the limitations and future directions clearly stated in the study? (RQ3)

Table 6 Data extraction form

| #	Data extraction category	Description	Purpose
Study general description			
1	Identifier	Identifier number (DOI)	
2	Date	Data extraction date	
Study description			
1	Title	Title of study	
2	Author name	Name of study authors	
3	Country	Country of study publication (first author)	
4	Publication year	Publication year	
5	Type	Conference proceedings, journal article	
6	Venue	Name of conference or journal	
7	Author affiliation(s)	Affiliation of author(s) (first author)	
Study content			
1	Objectives	Objectives of study	
2	Method	Method to support objectives	RQ1
3	Techniques	Clustering-discovery, clustering recommendation, association rule discovery, association rule recommendation, or combined	RQ1
4	Algorithm	Name of algorithm in study	RQ1
5	Evaluation metric	Name of evaluation metric	RQ1
6	Metric purpose	Evaluation metric purpose	RQ1
7	Data set	Name of data set for validation	RQ2
8	WS in data set	Number of WS in data set	RQ2
9	Sen vs. syn data set	Semantic or syntax data sets	RQ2
10	Data set purpose	Recommendation or discovery	RQ2
11	Future direction	Future direction	RQ3
12	Limitations	Restriction of the work	RQ3
13	Challenges	What challenges in this study are associated with web service discovery and recommendations?	

Fig. 3 Evaluation review protocol
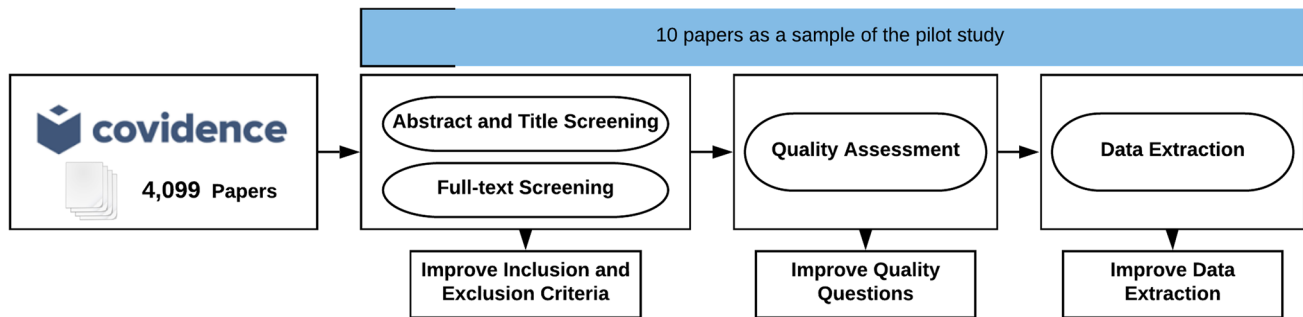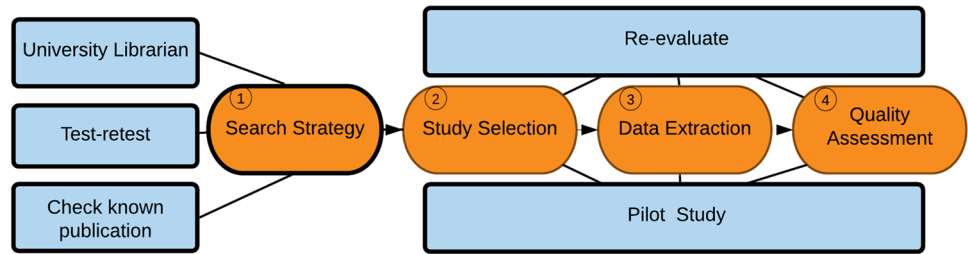
Fig. 4 Conducting the pilot study

Review Protocol Evaluation

Based on the recommendation of [1, 17], we confirmed that search strings were appropriately derived from the research questions. Furthermore, the search strings' effectiveness was preliminarily examined using a list of known publications [12, 21, 22] before using the full search. Search strings were constructed with the university librarian who had experience in working with the selected digital databases. We followed the test–retest approach to determine the final search strategy. To evaluate the study selection process, we ran and tested two stages of screening. This included title and abstract screening and full-text screening on a sample of retrieved papers with inclusion and exclusion criteria. As recommended by [1], we re-evaluated the screening stages to check the sample consistency. To evaluate quality assessment and data extraction, we performed a pilot study using a sample of the primarily selected papers to check the consistency and accuracy of the data. We confirmed that the extracted data addressed the research questions. Figure 3 illustrates the methods used in evaluating the review protocol. Furthermore, we externally evaluated the protocol before conducting the review. External experts were contacted for feedback and recommendations to refine the review protocol. Based on this feedback, we refined the review scope, improved the search strategy, and improved the inclusion and exclusion criteria.

The Conducting Phase

We conducted the search process to identify the final selected papers.

Identifying Relevant Research

Search strings were adapted and modified to fit the restrictions and requirements of the digital libraries. Some digital libraries supported searches for abstract, title, and keywords; other libraries supported full-text search. Appendix A indicates the search queries for each digital library. The total number of initially retrieved papers was 4581 where a total of 482 papers were excluded due to duplications. A selection process was performed on the remaining 4099 papers. The process proceeded with pilot selection and data extraction. All of the 4099 papers from the digital library search process were transferred into Covidence [18]. We used Covidence to facilitate the process of merging papers, removing duplicates, and conducting two-stage screening.

Pilot Selection and Extraction

Before running the selection process on 4099 papers, we ran a pilot study to test and evaluate selection, data extraction, and quality strategies. We followed the two stages of screening, including title and abstract screening and full-text screening with Covidence. The selection process stopped at

ten papers. The data extraction and quality assessment strategies were then applied as illustrated in Fig. 4. Based on this pilot study, we re-evaluated the selection, data extraction, and quality assessment strategies. We refined the review protocol based on the result of the pilot.

Selecting the Primary studies

The selection process consisted of two stages to screen the remaining 4099 papers.

- Abstract and title screening: Covidence was used to screen papers by looking at keywords, filtering keywords, and applying inclusion and exclusion criteria. Based on this stage of screening, 3,926 papers were excluded. We found that some of the retrieved papers did not address web service discovery and recommendation. For example, several papers related to biology, physics, and nature. Furthermore, some digital databases did not provide accurate results based on the chosen search strings. We excluded papers that did not explicitly address web service discovery and recommendation. Furthermore, we excluded abstracts, posters, and short papers (less than six pages), technical reports, tutorial summaries, and books.
- Full-text screening: 173 papers were scanned and 107 papers were excluded. Finally, 68 papers were selected based on the two screening stages.

Snowballing reference lists of the primarily selected papers led to the identification of nine additional primary papers. After applying inclusion and exclusion criteria, seven papers were excluded. Two additional papers were included. Figure 5 depicts a flowchart to illustrate the selection process of the primarily selected papers, including the manually selected papers before going to the quality assessment step.

The Reporting and Analysis Phase

We used various factors to analyze the selected papers, and we analyzed the content of the final papers to answer the research questions of this SLR.

Analyzing the Selected Papers

Quality of the Selected Studies The quality of the selected papers was identified using our quality assessment questions. Figure 6 indicates the primarily selected papers that successfully went through the data extraction and quality assessment process. Green bubbles provide the number of papers with high-quality scores (between 9 and 7.5). Blue bubbles provide the number of papers with average accept-

able-quality scores (less than 7.5 and greater than or equal to 5.5). Red bubbles provide the number of papers with low-quality scores, which lead to exclusion (less than 5.5). The results suggest that the selected papers are of relatively average acceptable quality. 15 papers are high quality and 42 papers are acceptable quality. Eleven papers are excluded as low quality. The list of the 57 final selected papers is provided in Appendix B.

Distribution of Papers over Years The selected papers that met our criteria were published in the last decade. The papers show a growing interest in this topic within the last 3 years. This study's search ranges from 2006 to January 2018 based on the search protocol. Figure 7, which illustrates the distribution of papers, shows that a significant number of papers were published in 2014 and 2015.

Distribution of Papers by Country Table 7 outlines the distribution of papers by country. It shows that while there is a general growth in the academic literature in the area of web service discovery and recommendation, specific research groups and authors play a major role in this growth. China has the largest number of contributions with 27 selected papers (47% of the overall selected papers). Distribution is based on the first author's country; it does not formulate a theory on the geographical allocation of teams working on web service discovery and recommendation at the time of the review. However, it shows a growing interest in the conducted study research area from different countries and teams.

The Result of Analyzing the Contents of the Final Selected Papers

In this section, we present the results of the analysis based on the contents of the final selected papers and present synthesized data to answer the research questions. We synthesized to demonstrate the classification of techniques used for web service discovery and recommendation according to the final selected papers. This was followed by a summarization of the methods. Then, synthesized data on the algorithms, similarity measures, and evaluation metrics were discussed. Finally, we discussed the classification of data sets used in web service discovery and recommendation approaches, as well as future trends, directions, and research gaps.

Classification of Web Service Discovery and Recommendation Techniques (RQ1)

After careful analysis of the final selected papers' contents, the papers were classified into five groups given their utilized techniques to support the process of web discovery and
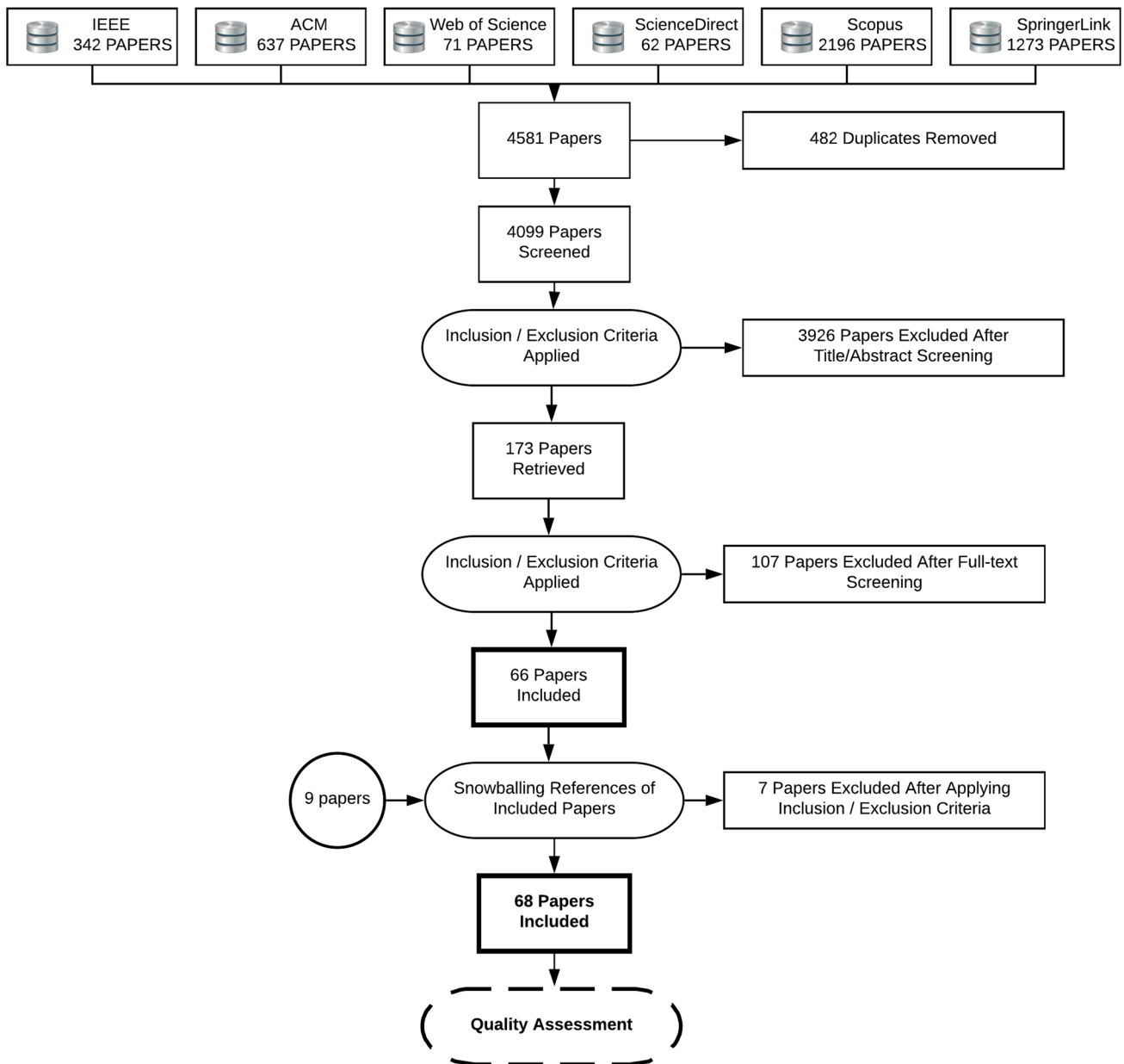
Fig. 5 Overall selection process

recommendation. Table 8 outlines the techniques used in the final selected papers, and Fig. 8 illustrates the classification of papers according to the techniques.

- **Clustering discovery:** This technique refers to papers in which web service discovery was indicated as a research problem and clustering techniques are used on web service data sets to facilitate the process of discovery.
- **Clustering recommendation:** This technique refers to papers in which web service recommendation was indicated as a research problem and clustering techniques

are used on specific web service data sets to facilitate the process of web service recommendation.

- **Association discovery:** This technique refers to papers in which web service discovery was indicated as a research problem and association rule techniques are used on specialized web service data sets to facilitate the process of web service discovery.
- **Association recommendation:** This technique refers to papers in which web service recommendation was identified as a research problem and association rule techniques were used to facilitate the process of web service recommendation.

The Quality of Selected Studies

● High Quality ● Acceptable Quality ● Low Quality

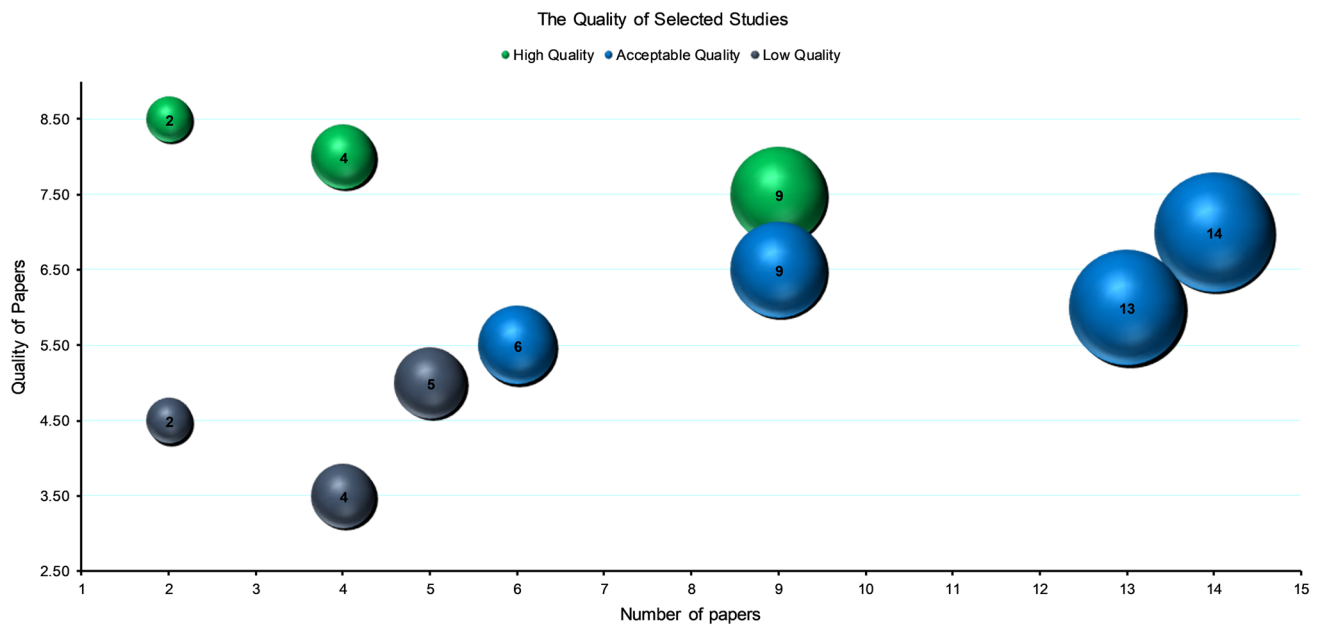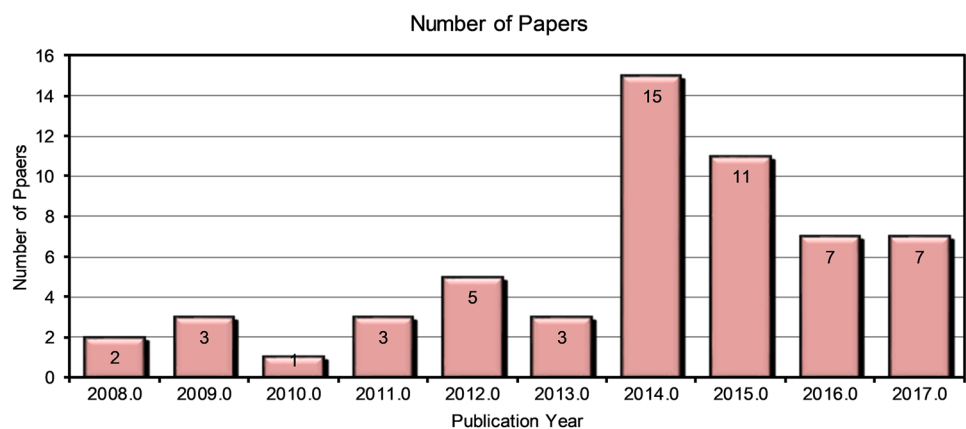**Fig. 6** Quality of the selected studies

Fig. 7 Distribution of papers over years

- **Combined technique:** This technique refers to papers in which both web service recommendation and discovery were identified as research problems and clustering or association rules (or both) were used to facilitate the process of web service discovery and recommendation.

Methods for Web Service Discovery and Recommendation (RQ1)

The methods proposed by researchers to facilitate the process of web service discovery and recommendation can be categorized as illustrated in our classification in Fig. 8. 53% of the selected papers used clustering to minimize the search spaces of web services by clustering similar web services using different similarity measures and clustering algorithms. 28% of the selected papers used clustering to support

the process of web service recommendation by clustering similar users or similar web services to facilitate the process of selection and recommendation. Two selected papers used association to support the process of web service discovery. Three papers used association to support the process of web service recommendation. Finally, 11% of the selected papers used clustering or association rule to support the combined process of web service discovery and recommendation. In the following sections, we synthesize the final selected papers based on our classification of the methods, providing more details about the used methods and their specifications. In each sub-section, we present a table which lists the clustering algorithms used, features and specifications of the proposed method, the type of web service description language used, and whether the proposed clustering is based on functional or non-functional properties of web services.
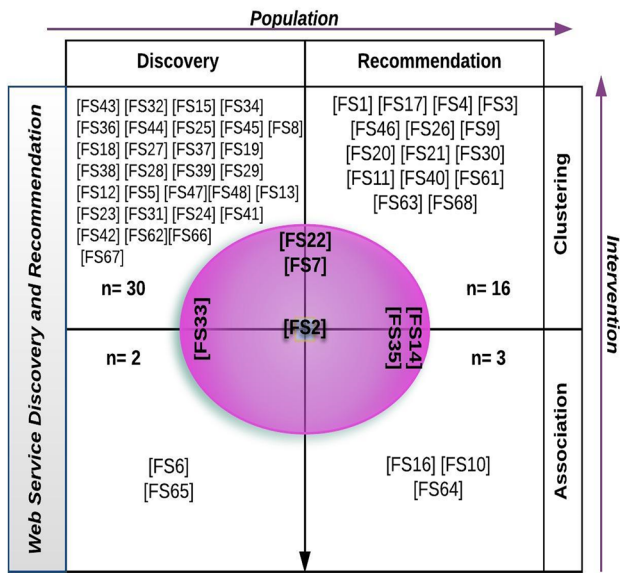
Fig. 8 Classification of papers according to the DM techniques employed

Table 7 Distribution of papers by country

Country	Number of papers	Percentage (%)
USA	5	9
UK	1	2
Tunisia	1	2
Sri Lanka	2	4
Spain	1	2
Romania	1	2
Japan	5	9
Italy	1	2
India	2	4
France	2	4
China	27	47
Canada	3	5
Brazil	1	2
Austria	1	2
Australia	3	5
Argentina	1	2
Total	57	100

Table 8 Classification of papers based on techniques

Number of papers	Techniques	(%)
30	Clustering discovery	53
16	Clustering recommendation	28
2	Association discovery	4
3	Association recommendation	5
6	Combined	11
57	Total	100

Methods Based on Clustering for Web Service Discovery

Table 9 shows the list of clustering discovery techniques. To enhance the performance of web service clustering and address the drawbacks of limited service data being available for clustering, [FS32] [FS12] used K-means clustering and NLP for functional-based web service clustering after incorporating auxiliary long texts from Wikipedia[2] with corresponding Tags to learn from the set of data and improve clustering. Tag-aided dual Author Topical Model (TD-ATM) on short text from PW[3] was used in [FS32] and [FS12] in a Dual Tag-aided Latent Dirichlet Allocation (DT-LDA) method based on transferring learning from auxiliary long-text data from Wikipedia to enrich the service data obtained from PW. For the same reason, [FS36] used Hierarchical Agglomerative Clustering (HAC) and WordNet in a crawler-based system to gather web service descriptions and functionally cluster them. NLP is used for WSDL feature extraction, and noun phrases are discovered from vectors for tagging services. Furthermore, [FS47] used k-means for functional clustering using Nonparametric Bayesian model-based Latent Functional Factors (BN-LFFs) for large-scale web service clustering, which can learn the number of LFFs in a service space. This includes representing web services in LFF space, identifying service LFFs and adding more information to services to improve clustering for discovery purposes.

To increase the performance of clustering using hidden semantic patterns and concepts extracted from the service description, [FS34] used HAC for an ontology-learning method. Three WSDL features are extracted, an ontology is generated for each feature, similarities are captured based on domain ontology learning (logic-based reasoning) and edge court based on WordNet measures, and features are integrated and clustered. [FS44] used k-means to cluster services functionally and WordNet and ConceptNet to extract concepts and semantic relations from service descriptions. It also extracts concepts from users' queries and guides users to formulate queries to return services associated with the concepts. To address the lack of semantics and scalability in web service discovery, [FS25] used modified k-means for functional clustering and the cosine angle for the similarity between documents. The method includes two main phases to decompose service collection and match services semantically. An extensive service collection is partitioned into a set of smaller clusters. The Singular Value Decomposition (SVD) technique is applied to the cluster, so that service matching against the query can be carried out at the concept level. SVD is used to extract the semantics between the user

Table 9 List of clustering discovery techniques

Techniques	REF#	Algorithm	Features/specifications	Population	Functional	Non-functional
Clustering-discovery	[FS43]	k-medoids	WordNet, Levenshtein Distance, Ontology	OWL-S	+	–
	[FS32]	k-means	NLP, Tags, TD-ATM, PW, Wikipedia	Short text	+	–
	[FS15]	HAC	NLP, WordNet, TFIDF, Ontology, CAS,	WSDL	+	–
	[FS34]	HAC	NLP, Ontology, WordNet	WSDL	+	–
	[FS36]	HAC	NLP, WordNet, Tags, crawler-based	WSDL	+	–
	[FS44]	k-means	NLP, WordNet, ConceptNet	WSDL/WADL	+	–
	[FS25]	k-means	Cosine, TFIDF, service collection, SVD	WSDL	+	–
	[FS45]	Quality threshold	NLP, NGD, context-aware, mobile environments	WSDL	+	–
	[FS24]	Quality threshold	NLP, NGD	WSDL	+	–
	[FS8]	HAC	NLP, Ontology, CTM, FCA, Apache Lucene	WSDL	+	–
	[FS18]	HAC	TFIDF, WordNet, Ontology	OWL-S	+	–
	[FS27]	HAC	Ontology learning, TFIDF, WordNet, Ontology	WSDL	+	–
	[FS37]	HAC	Ontology learning, TFIDF, WordNet, Ontology	WSDL	+	–
	[FS19]	k-means	NLP, TFIDF, co-clustering, bipartite graph	WSDL	+	–
	[FS38]	Self-Join RDB Clustering	Relational data-based self-join operation, ontology	OWL-S	+	–
	[FS28]	HAC	Euclidean distance, cosine, search engine	WSDL	+	–
	[FS39]	k-means	NLP, TFIDF, Apache lucene, CE-RSR	WADL	+	–
	[FS29]	k-medoids, HAC	Ontology, PAM, business process integration	OWL-S	+	–
	[FS12]	k-means	NLP, Tags, DT-LDA, PW, Wikipedia	Short text	+	–
	[FS5]	DBSCAN	Ontology, TFIDF, matching, ranking	OWL-S	+	–
	[FS47]	k-means	BN-LFF, add information in LFF space	WSDL	+	–
	[FS48]	HAC	NLP, TFIDF, cosine, WordNet	WSDL	+	–
	[FS13]	Neural network	WordNet, LSI, cosine, TFIDF, Mahalanobis distance	WSDL	+	–
	[FS23]	PSO	Ontology, DoM, optimization problem	SAWSDL	+	–
	[FS31]	Self-organizing based clustering	Ontology, taxonomic clustering, SGPS	OWL-S	+	–
	[FS41]	Tree-traversing ant (TTA) clustering	NLP, NGD, n° W, search engine	WSDL	+	–
	[FS42]	k-means	NLP, TFIDF, cosine, Euclidean distance, PLSA	WSDL	+	–
	[FS51] [FS52]	k-means HAC	NLP, mashup services, LDA topic model, Tags, topic clustering	Text	+	–
	[FS55]	HAC	NLP, ontology generation, specific terms	OWL-S	+	–
	[FS56]	k-means++	Word vectors, LDA, topic clustering	WSDL	+	–

Disocvery Published Benchmarks

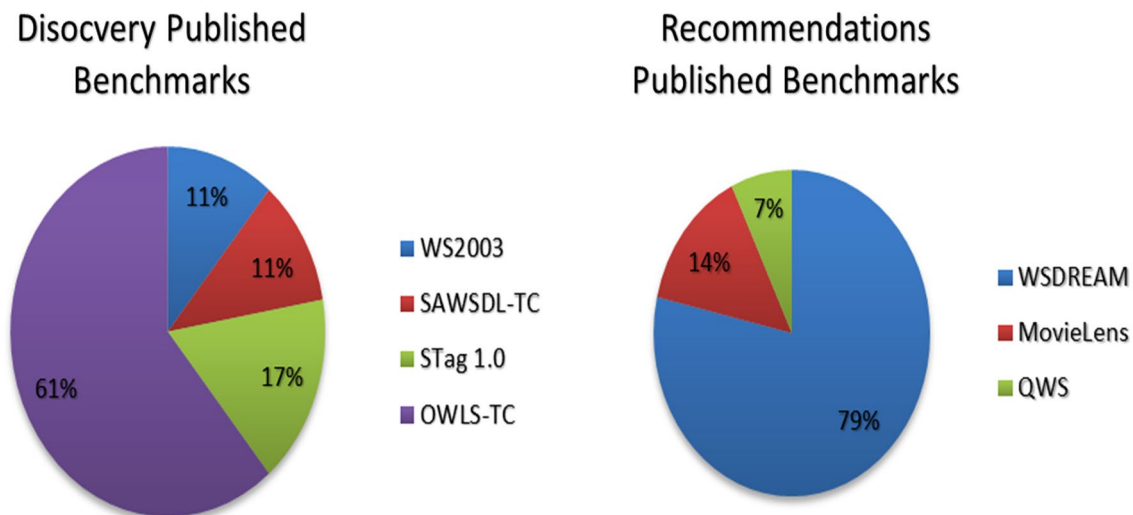## Recommendations Published Benchmarks

Fig. 9 Published benchmark usage

query and the service description. To overcome the time and computational complexity of logic-based semantic matching by extracting semantic, [FS8] used HAC and NLP for non-logic-based web service discovery including matching and ranking. Correlated Topic Model (CTM) is used to extract topics from semantic service descriptions and model the correlations between the extracted topics. Based on the topic correlation, service descriptions are grouped into hierarchical clusters, and Formal Concept Analysis (FCA) formalism is used to organize the constructed hierarchical clusters into concept lattices according to their topics. Using user queries to find web services based on keyword-based and semantic extraction, [FS42] used k-means clustering to eliminate and filter out irrelevant services based on user queries. Probabilistic Latent Semantic Analysis (PLSA) is used to capture semantics hidden behind the words in user queries and web service descriptions. NLP, term frequency–inverse document frequency (TFIDF), and cosine and Euclidean distance are used for similarity measurements.

To minimize the search space, in [FS43], a modified k-medoids clustering algorithm was used to functionally cluster similar web services by utilizing its OWL-S semantic representation. WordNet, Levenshtein Distance (LD), and domain ontology similarity measures are used. To include more information and minimize the search space, [FS45], [FS24] used a quality threshold clustering algorithm and normalized Google distance (NGD) and NLP to extract the features of WSDL documents based on five features and cluster web services functionally based on integrated similarities with weights. Providing a Discovery as a Service (DaaS), a cloud-based context-aware service discovery framework for mobile environments takes into consideration network characteristics, user preferences and context,

and device profiles. To overcome the limitations of service repositories, especially with the growing size of their services, [FS18] used HAC to cluster web services in the repository based on their functions using distance based on semantic subsumption relations to organize services registered in the repositories into clusters for the purpose of improve the response time, recall, and precision.

To improve semantic similarity using up-to-date knowledge and fine-grained information, [FS15] used the HAC algorithm and NLP for a context-aware similarity (CAS) method. CAS is based on a post-filtering method to increase the performance of web service clustering using support vector machine (SVM) and domain context data sets from search engines. SVM was trained to classify domains based on terms extracted from Google and Wikipedia. To improve web service clustering using different similarities, [FS27] [FS37] used HAC for functional clustering, TFIDF, WordNet, and Ontology for term-similarity measures in a multiphase web service clustering method including a feature extraction phase using NLP, ontology-learning phase, similarity calculation phase, feature-integration phase, and clustering phase. To overcome the limitations of semantic similarity distance measures and their thresholds, [FS31] used a technique based on a self-organizing clustering algorithm called taxonomic clustering for functionally organizing semantic web service advertisements. A Semantic Genome Propagation Scheme (SGPS)-based ontology was used to measure the similarity between semantic concepts to semantically position web services in the cluster space by searching the most specific parent (MSP) and the least specific children (LSC).

To reduce the drawbacks of only using features extracted from service descriptions in clustering and ignoring other

words in the document, [FS19] used k-means and TFIDF for co-clustering a WSDL document and its words after extracting WSDL features using NLP and dealing with the discovery problem as a bipartite graph partitioning problem. To improve the efficiency of web service clustering, [FS38] used Relational Data-based self-Join Operation (RDBJO) to cluster web services functionally using Ontology. Ontology technology was used to do the computation from the semantic level, and the self-join operation in RDB was used to do the calculation on the service interface and capability tables. To enhance search engine results with a list of similar services for each hit, [FS28] used HAC for functional clustering, Euclidean distance, and cosine in web-based search engines allowing web service consumers to easily find and relate specific services to a given query. The clusters are built for results of a search query only, which limits the number of elements to a reasonable size and improves the visibility of the results. To improve REST service discovery and explore the discovery of different types of web services, [FS39] used different types of k-means for functional clustering after using NLP for pre-processing steps on WADL files, TFIDF, and the Apache Lucene framework for indexing to build a Cluster Enhanced REST Service Registry (CE-RSR). Motivated by the fact that searching for the best cluster can be viewed as an optimization problem, since the number of services involved in a clustering process is large, [FS23] used Particle Swarm Optimization (PSO) to functionally cluster web services where they define the semantic similarity metrics between web services based on the degree of match (DoM) using ontology.

To minimize the discovery time and maximize the use of web services for business process integration within organizations, [FS29] used partitioning around medoids (PAM) and HAC to functionally cluster web services based on semantically similar domains using ontology. To minimize the discovery time among large service pools, [FS5] used a DBSCAN in the pre-clustering step using functional similarities based on ontology and process similarities going over both web service clustering and matching. To improve the efficiency and accuracy of semantic web service discovery, [FS48] used HAC for functional clustering, TFIDF, cosine, and WordNet for similarity measures. Four WSDL features are extracted using NLP pre-processing steps. To overcome the drawback of keyword-based discovery systems, and motivated by the fact that most web services are in non-semantic form, [FS13] used a modified Kernel Batch Self-Organizing Map (KBSOM) neural network for clustering using WordNet and latent semantics index (LSI) to represent a web service as a service feature vector. Cosine, TFIDF, and Mahalanobis distance were used as similarity measures in the service clustering and matching process. To gather, discover, extract and integrate features from WSDL files automatically, [FS41] used the Tree-Traversing Ant

(TTA) clustering algorithm to functionally cluster web services into homogenous service communities. NLP was used to extract features, and NGD and Wikipedia (n° W) were used as similarity measurements.

To accurately discover users desired mashup services, [FS51], [FS52] proposed a method for clustering mashup services based on exploring services document content and network. Data from PW are used where two-level topic model (LDA) is designed to mine the latent functional topics by incorporating relationships among mashup services and their content. The similarity between mashup services calculated using different similarity measures where a combination of k-means and HAC are used to cluster similar mashup services. To improve clustering web services based on their functional properties, [FS55] proposed a method to cluster web services based on ontology generation of the extracted terms by focusing on the specific terms than general terms where the specificity of terms are calculated before generating ontology. Ontology relationship and IR-based similarities are used to find the similarities between web service and cluster them based on HAC. This work used [FS34] approach and added their method of ontology generation based on specific terms. A topic-based clustering method proposed by [FS56] uses augmented LDA model to improve the clustering of web services based on functional properties. Motivated by the lacking of text information, they trains the latent topic information on web service descriptions using Word2vec where web services with the same topic are clustered together. The training process done based on clustering smiler words from web services documents into similar clusters.

Methods Based on Clustering for Web Service Recommendation

Table 10 shows the list of clustering recommendation techniques. To overcome the challenge of data sparsity in QoS predication using Collaborative filtering (CF), [FS17] used HAC to cluster services based on their physical environment, measured the similarity of users using Pearson correlation coefficient (PCC) based on the clusters, and then applied CF. [FS20] utilized customer profiles and historical usage experiences of service invocation using K-means to cluster customers. CF was used to study users' behaviors from a service usage history perspective and then find patterns among web services in the repository to predict a user's preferences on certain web services. PCC, cosine, and Euclidean distance are used to measure similarity. K-means was also used in [FS11] in a clustering collaborative filtering (CluCF) method by employing users and service clusters using time-aware similarity measures and using the location factor to update clusters. QoS performance and invocation time were used to improve the prediction accuracy. PCC was used to calculate the similarities between users and services.

Most existing approaches ignore the data credibility problem leading to the unreliable QoS data contributed by dishonest users. [FS1] used k-means clustering on historical QoS data to cluster users and services, predict QoS, and recommend a service. CF and clustering are used where the service reputation is calculated, similar users are identified, trustworthy users are identified, similar services are identified, and then, missing values can be predicted. PCC was used to measure the similarity. To overcome the challenge that historical web service QoS data are not updated in real-time leading to low QoS prediction accuracy, [FS40] proposed a landmark-based QoS prediction based on User-Based Clustering (UBC) and Web Service-Based Clustering (WSBC) based on HAC for web services. Real QoS data from a set of fixed landmarks from PlanetLab and PCC as the similarity measure between two users were used. To increase the performance of QoS prediction and to provide a more personalized recommendation, [FS4] used HAC to cluster users and services based on their locations and QoS data. The system works to predict web services' QoS values and recommend the best one for active users based on historical web service QoS records using QoS-aware collaborative filtering. PCC is used as a similarity measure between users based on the QoS values of web services they both invoked.

To handle the problem of limited tags attached with web services, [FS3] proposed a hybrid Web Service Tag Relevance Measurement mechanism (WS-TRM) to measure the relevance of a tag to a web service using semantic similarity and tag authority. WSDL features extracted with the tag are calculated using NGD and then integrated with Tag-authority-based Hyperlink-Induced Topic Search (HITS) similarity. WS-TRM is applied in tag recommendation for web services. Users invoke a few services leading to many missing QoS values, and the user-service matrix might be sparse. [FS9] [FS26] used k-means to cluster users and services based on their locations in a location-based hierarchical matrix factorization (HFM) method to predict missing QoS values. User-service global context and geographic information were used to build an HFM model-based CF method to predict missing values.

To prevent time-consuming and inaccurate recommendations for semantic web services for a composite service, [FS21] proposed a method of Clustering and Recommendation for OWL-S web services in Evolution (CRE) using HAC-based WordNet, TFIDF, and cosine similarity measurement and matrix factorization recommendation. Using the historical QoS experiences of similar users to predict a user's QoS on a known web service, [FS30] used k-means clustering to cluster users and services in a global structure using Relational Clustering-based Model (RCM), a collaborative filtering-based scheme, to evaluate the QoS of a priori unknown service providers. CF is used to predict the QoS of unknown web services. PCC, WordNet, and cosine are used

to measure similarities. Based on the use of heterogeneous features to improve recommendation, [FS46] used k-means to cluster services in a Web Service Heterogonous Feature Selection (WS-HFS) method on heterogeneous data sources of web services to perform web service mining. The framework has two components: feature selection and service mining. Features are selected and transformed into knowledge in the feature subspace, and a data source-based weight learning method is proposed.

To increase the performance of web service selection and recommendation, [FS49] proposed clustering web service based on their functionalities using HTS [13]-based semantic similarities and cluster them again based on their manually assigned QoS values using Spherical-Associated Keyword Space (SASKS) algorithms. By not only considering QoS metrics for web service recommendations, [FS50] proposed a method to include web service functional properties extracted from WSDL files in addition to QoS metric for web service recommendations. Fuzzy C-means clustering algorithm is used to cluster user and services where matrix factorization approach used that integrates the functional categories of the web service. [FS57] proposed an active learning approach for web service tag recommendation where Support Vector Machine (SVM) is trained on labeled web services to classify unlabeled web services based previously labeled small set of web services by a domain expert. By learning the correlations among tags by computing Jaccard and hierarchical clustering-based tag correlations, they minimize the efforts of the domain expert using an active learning-binary classification on a service for each possible label to determine if the tag should be recommended to the service.

Methods Based on Association Rules for Web Service Discovery

Table 11 shows the list of association discovery techniques. To infer patterns from service descriptions by providing a summarized and integrated representation of web service functionality, [FS6] used the Apriori algorithm to extract useful knowledge from a set of data sets based on WSDL files and service invocation. Web service discovery via intentional knowledge mining is a method for extracting a useful intentional representation of web service repository contents to help application developers learn new knowledge regarding the available service and make the correct choice. Using web service interfaces when its parameters contain meaningful synonyms, abbreviations and semantic can improve the discovery process, [FS54] propose an approach that mine the underline semantics and conduct semantic extension of web service interfaces. They mine the underling semantics to create index libraries using association rules and then clustering interaction interface names and fragment under the supervision of co-occurrence probability.

Methods Based on Association Rules for Web Service Recommendation

Table 12 shows the list of association recommendation techniques. To decide which service among the retrieved set of semantically equivalent web service candidates is the best, [FS16] used the Apriori algorithm and CF to study users' invocation behaviors from a web service composition perspective and then use association rules among web services in the repository to predict users' preferences on certain web services. User usage patterns are considered, and a profile is generated to provide a personalized ranking for the retrieved set of web services. For the same purpose, [FS10] used the Apriori algorithm to understand the correlation between users and construct group and individual profiling based on usage history. A group knowledge-based recommendation system using CF is used for ranking and recommendation. By exploring users' relations, their history of used services, and the quality of services, [FS53] proposed a web service relationships track (WSR-Track) approach where service ecosystems represented in heterogeneous multigraph. Nodes represent users and services, Edges represent trust/usage relationships between users and services. By incorporating history and QoS data in addition to users, service data association rules are generated and web service recommendation process provides adequate web service for users based on tracked relation.

Methods Combining Clustering/Association Rules for Web Service Discovery and Recommendation

Methods which used clustering for web service discovery and recommendation include [FS7],[FS22] where their techniques support both matching (discovery) and selection (recommendation):

Table 13 shows the list of combined techniques. Because clustering can boost the power of web services' search engines and generating tags can improve the search accuracy, [FS7] used k-means clustering and Carrot search clustering to cluster web services and NLP to extract service features in an automatic tagging for web services using machine learning. The method extracts relevant tags from a WSDL file and uses co-occurrence to generate tags from a collection of untagged service descriptions files. Cosine similarity is used to measure the similarity between the user query and the content vector of the web service document. Given the fact that the availability of QoS is important in web service recommendation and to predict the QoS of non-invoked web services, [FS22] used NLP to extract features and k-means to cluster similar web services using semantic LSI and non-semantic approaches. Content-based and Slop one CF were used to predict QoS values and recommend the

top-k web services to users. For similarity cosine, PCC and TFIDF are used.

Methods which combined clustering and association rules for web service recommendation include [FS35] and [FS14]

By organizing web services into similar clusters to reduce the search time, [FS35] used HAC and gain-based association rule classification (GARC) to recommend web services for the currently invoked service using the proposed cluster-based service recommendation approach. HTS as presented in [FS34], TFIDF, Ontology, and WordNet are used. To reduce the dimensionality of a sparse matrix and solve the cold-start problem in web service recommendation, [FS14] proposed the User Relationship and Preferences Clustering and Recommendation (URPC-Rec) algorithm to combines users' historical behaviors and their personal interests to perform personal service recommendation for new users. The method combined a defined clustering algorithm and a recommendation algorithm based on finding an association between users and services. User interest tags and social network relationships and information are used, and WordNet is used to measure the similarity in the recommendation.

Method combined clustering and association rules for discovery: To overcome the problem where web service discovery given non-explicit semantic based on service description to match a specific user request, [FS33] used HAC for clustering and Hyperclique patterns to find association in an integrated method for semantic web service discovery, which relies on semantic web service categorization and semantic web service selection. The method utilizes WSDL descriptions and adds semantics using the WordNet database and Suggested Upper Merged Ontology (SUMO) mapping. The selection process relies on the use of LSI and finds the association between permeants in a cluster.

Method combined both clustering and association rules for web service discovery and recommendation: [FS2] used the K-means clustering algorithm to cluster web services and association rules for tag recommendation. They used NLP to extract the features from a WSDL document and add tags as another feature. A web service tag recommendation strategy was proposed to overcome the limitations of noise and uneven distribution of tags where association rules are used to extract the association using training and testing data sets (Tables 10, 11, 12 and 13).

Algorithms for Web Service Discovery and Recommendation (RQ1)

The clustering and association rule algorithms used by the researcher to facilitate the process of web service discovery and recommendation are outlined in Table 14. K-means is the most-used clustering algorithm (42% of the final selected papers). This was followed by the HAC algorithm (36%).

Table 10 List of clustering recommendation techniques

Techniques	REF#	Algorithm	Features/specifications	Population	Functional	Non-functional
Clustering recommendation	[FS1]	k-means	QoS credibility, CF, PCC, user clustering, service clustering	QoS	–	+
	[FS17]	HAC	Data sparsity, clustering QoS prediction, PCC, user similarity	QoS	–	+
	[FS4]	HAC	Personalized recommendation, CF, users location, PCC	QoS	–	+
	[FS3]	k-means	NLP, tag recommendation, NGD, tag relevance	WSDL	+	–
	[FS9] [FS26]	k-means	QoS recommendation, matrix factorization, location-based	QoS	–	+
	[FS20]	k-means	PCC, cosine, Euclidean distance, CF, customers' clustering	WS	+	–
	[FS21]	HAC	WordNet, cosine, TFIDF, matrix factorization	OWL-S	+	–
	[FS30]	k-means	RCM, cosine, CF, PCC, WordNet, QoS prediction	QoS,WS	+	+
	[FS11]	k-means	CluCF, PCC, data sparsity, location user clustering	QoS	–	+
	[FS46]	k-means	Mining web services, heterogonous feature selection	WSDL, Tags	+	–
	[FS40]	HAC	Real-time QoS, PCC, user clustering, Qos predication	QoS	–	+
	[FS49]	HAC, SASKS	Hybrid Term Similarity, Euclidean distance,	WSDL, OWL-S	+	+
	[FS50]	Fuzzy C-Means	PCC, matrix factorization, NGD	QoS, WSDL	+	+
	[FS57]	HAC	Tag recommendation, SVM, Jaccard, active learning	Text	+	–

Table 11 List of association discovery techniques

Techniques	REF#	Algorithm	Features/specifications	Population	Functional	Non-functional
Association discovery	[FS6]	Apriori algorithm	Intensional knowledge mining, intensional querying system	WSDL	+	–
	[FS54]	Apriori algorithm	Web service model extraction, interface underling semantic mining, co-occurrence	WSDL/WADL	+	–

Table 12 List of association recommendation techniques

Techniques	REF#	Algorithm	Features/specifications	population	Functional	Non-functional
Association recommendation	[FS16]	Apriori algorithm	CF, PCC, service composition, ranking	Movielens	–	–
	[FS10]	Apriori algorithm	CF, PCC, usage history, user profiling	Movielens	–	–
	[FS53]	Apriori algorithm	Graph representation, relationship tracking,	WS-PW	–	+

Table 13 List of combined techniques

Techniques	REF#	Algorithm	Features	Population	Functional	Non-functional
Combined	[FS7]	k-means	NLP, Carrot search, Tags recommendation, cosine	WSDL	+	−
	[FS22]	k-means	NLP, TFIDF, cosine, PCC, CF, QoS predication	WSDL, QoS	+	+
	[FS35]	HAC, GARC	TFIDF, WordNet, ontology, association factor	WSDL, OWL-S	+	+
	[FS14]	Defined	Tags, social network relationships, WordNet, algorithm	QoS	−	+
	[FS33]	HAC, Hyperclique patterns	NLP, WordNet, SUMO, LSI, cosine	WSDL	+	−
	[FS2]	k-means, defined AR	NLP, NGD, Tags, TFIDF, jaccard coefficient	WSDL	+	−

Table 14 Clustering and association algorithms

Clustering algorithms		
k-medoids clustering	2	[FS43] [FS29]
k-means clustering	21	[FS32] [FS7] [FS44] [FS25] [FS1] [FS17] [FS3] [FS46] [FS26] [FS9] [FS19] [FS20] [FS39] [FS2] [FS30] [FS11] [FS12] [FS47] [FS22] [FS42] [FS51] [FS56]
Hierarchical agglomerative clustering algorithm (HAC)	20	[FS15] [FS33] [FS34] [FS35] [FS36] [FS8] [FS17] [FS4] [FS18] [FS27] [FS37] [FS28] [FS29] [FS21] [FS48] [FS40] [FS51] [FS54] [FS55] [FS57]
Density-based spatial clustering of applications with noise (DBSCAN)	1	[FS5]
Quality threshold (QT) clustering algorithm	2	[FS45] [FS24]
Neural network clustering	1	[FS13]
PSO clustering algorithm	1	[FS23]
Self-organizing-based clustering algorithm	1	[FS31]
Self-Join RDB Clustering	1	[FS38]
Tree-traversing ant algorithm	1	[FS41]
Fuzzy C-means	2	[FS50] [FS52]
Association rules algorithms		
Hyperclique patterns	1	[FS33]
Apriori algorithm	4	[FS16] [FS10] [FS6] [FS53]
GARC: gain-based association rule classification	1	[FS35]
Defined correlation	2	[FS2] [FS14]

K-means clustering was used with modifications in [FS32], [FS7], [FS44], [FS25], [FS1], [FS3], [FS46], [FS26], [FS9], [FS19], [FS20], [FS39], [FS2], [FS30], [FS11], [FS12], [FS47], [FS22], and [FS42]. Hierarchical agglomerative clustering was used with modifications and adjustments in [FS15], [FS33], [FS34], [FS35], [FS36], [FS8], [FS17], [FS4], [FS18], [FS27], [FS37], [FS28], [FS29], [FS21], [FS48], and [FS40]. Bio-inspired clustering algorithms were used to cluster web services in [FS13] using a neural network, [FS23] used particle swarm optimization, and [FS41]

used the tree-traversing ant algorithm. The quality threshold clustering algorithm is another partition-based clustering algorithm used in [FS45] and [FS24]. Density-based clustering (DBSCAN) was used in [FS5]. The self-organizing-based clustering algorithm was used in [FS31]. A relational database (RDB), which is a self-join clustering algorithm, was used in [FS38]. Among the final selected papers, clustering algorithms dominated and were more popular when comparing the association rule algorithms. For association rule algorithms, the well-known Apriori algorithm (43%)

Table 15 Similarity measures

Similarity measures	Type	#used	Papers
WordNet	Semantic	15	[FS43] [FS15] [FS34] [FS35] [FS36] [FS44] [FS27] [FS21] [FS30] [FS48] [FS14] [FS18] [FS55] [FS49]
Normalized Google distance (NGD)		7	[FS45] [FS3] [FS2] [FS24] [FS41] [FS50] [FS55]
Ontology		15	[FS43] [FS15] [FS34] [FS35] [FS27] [FS38] [FS29] [FS5] [FS23] [FS18] [FS31] [FS18] [FS8] [FS49] [FS55]
Wikipedia		1	[FS41]
Cosine similarity	Syntactic	10	[FS7] [FS33] [FS25] [FS20] [FS30] [FS48] [FS13] [FS42] [FS28] [FS21]
TFIDF		16	[FS43] [FS15] [FS35] [FS27] [FS37] [FS19] [FS39] [FS2] [FS21] [FS48] [FS22] [FS42] [FS13] [FS18] [FS25] [FS49] [FS50] [FS55]
Pearson correlation coefficient (PCC)	Distance	11	[FS16] [FS1] [FS17] [FS4] [FS20] [FS10] [FS30] [FS11] [FS22] [FS40] [FS50]
Levenshtein distance (LD)		1	[FS43]
Euclidean distance		4	[FS20] [FS28] [FS49] [FS56]
Jaccard coefficient		3	[FS2] [FS51] [FS56]
Mahalanobis distance		1	[FS13]

was used in [FS16], [FS10], and [FS6] to find the association between different variables to meet the research objectives.

Similarity Measures for Web Service Discovery and Recommendation (RQ1)

Similarity measures used in the final selected papers for the purpose of web service discovery and recommendation are listed in Table 15. Used similarity measures can be classified into:

- *Semantic-based similarity measurements:* These measurements include WordNet, NGD, Wikipedia, and Ontology. WordNet [63] is a lexical database that groups English words into sets of synonyms and records various semantic relations between these synonym sets. Eighteen percent of the final selected papers used WordNet where the similarity measure based on the structured knowledge base, and edge-based method mostly used. NGD [64] is computed by the Google web search engine to measure semantic similarity and relatedness between words and recently stated as search engine similarity. Wikipedia was used as a knowledge base semantic similarity measure in [FS41]. Ontology using logic-based reasoning was used in matchmaking and semantic similarity measurement for web services in [FS43], [FS15], [FS34], [FS35], [FS27], [FS38], [FS29], [FS18], [FS31], [FS5], and [FS23].
- *Syntactic-based similarity measurements:* These measurements include IR and NLP techniques for measuring similarities in web service discovery and recommendation. TFIDF is a text-mining method used mostly in feature extraction process. It measures the importance of words in a document. TFIDF was used in [FS43], [FS15], [FS35], [FS27], [FS37], [FS19], [FS39], [FS2], [FS21], [FS48], [FS13], [FS22], [FS25], and [FS42]. This is

approximately 20% of the final selected papers. Cosine similarity was used to measure the similarity between two documents in the vector space by calculating the cosine of their angle. Cosine similarity was used in [FS7], [FS33], [FS25], [FS20], [FS30], [FS48], [FS28], [FS13], [FS21], and [FS42].

- *Distance-based similarity measurements:* These measurements were used to measure the similarity between web services, including Levenshtein distance (LD), Euclidean distance, Mahalanobis distance, and Jaccard coefficient. LD is used to measure the similarity between two strings. The distance is the number of deletions, insertions, or substitutions required to transform one string into another [65], as in [FS43]. The Euclidean distance was used to measure the distance between two vectors in [FS20], [FS42], and [FS28]. Mahalanobis distance was used in [FS13]; the Jaccard coefficient was used in [FS2]. The PCC was used mostly to find the correlation and similarity for web service recommendation. PCC measures the linear correlations between two variables. It has been used in [FS16], [FS1], [FS17], [FS4], [FS20], [FS10], [FS30], [FS11], [FS22], and [FS40].
- *Hybrid-based similarity measurements:* These measurements are used by combining semantic similarity measurement, syntactic-based similarity measurements, or distance-based similarity measurements. For example, [FS43] combined WordNet, NGD, and ontology as semantic similarity measures, TFIDF as IR-based syntactic similarity measurements, and LD as distance-based similarity measures. Hybrid-based similarity measures are the most-used measurements. They provide a better solution by integrating different similarities.

Based on the algorithms and the similarity measures used in the final selected papers, Table 16 discusses the most-used

Table 16 Similarity measures *vs.* algorithms

Similarity measures	k-medoids	k-means	HAC	DBSCAN	QT	Neural network clustering	PSO clustering algorithm	self-organizing clustering	Self-Join RDB Clustering	Tree-traversing ant	Hyper-clique patterns	Apriori algorithm	GARC	Defined correlation	Fuzzy c-mean
WordNet	[FS43]	[FS44] [FS30]	[FS15] [FS34] [FS35] [FS36] [FS27] [FS21] [FS48] [FS18] [FS55]										[FS35]	[FS14]	
Cosine similarity		[FS7] [FS25] [FS20] [FS30] [FS42]	[FS33] [FS28] [FS48]			[FS13]					[FS33]				
PCC		[FS1] [FS17] [FS20] [FS30] [FS11] [FS22]	[FS42] [FS4] [FS40]									[FS16] [FS10]			[FS50]
NGD	[FS43]	[FS3] [FS2]	[FS55]		[FS24] [FS45]	[FS13]				[FS41]				[FS2]	[FS50]
TFIDF	[FS43]	[FS19] [FS39] [FS2] [FS22] [FS42]	[FS15] [FS27] [FS37] [FS21] [FS48] [FS55]										[FS35]	[FS2]	[FS50]
Ontology	[FS43] [FS29]		[FS15] [FS34] [FS35] [FS27] [FS29] [FS18] [FS55]	[FS5]			[FS23]	[FS31]	[FS38]				[FS35]		
LD	[FS43]														
Wikipedia										[FS41]					

Table 16 (continued)

| Algorithms | | | | | | | | | | | | | | | |
Similarity measures	k-medoids	k-means	HAC	DBSCAN	QT	Neural network clustering	PSO clustering algorithm	self-organizing clustering	Self-Join RDB Clustering	Tree-traversing ant	Hyper-clique patterns	Apriori algorithm	GARC	Defined correlation	Fuzzy c-mean
Euclidean distance		[FS20] [FS56]	[FS28]												
Jaccard coefficient	[FS2] [FS51]	[FS51]	[FS57]										[FS2]		
Mahalanobis distance						[FS13]									

algorithms, linking them to the similarity measures used. Furthermore, it shows gaps that the proposed methods did not investigate. For example, it indicates that a combination of using HAC with the semantic similarity measure Word-Net is one of the most-used techniques in the proposed methods. It also indicates that HAC is not used with the NGD semantic similarity measure.

Evaluation Metrics Used for Web Service Discovery and Recommendation (RQ1)

The evaluation metrics used in the final selected papers to validate and evaluate the proposed web service discovery and recommendation approaches are outlined in Table 17. Knowing the evaluation metrics used to evaluate the proposed methods helps researchers in comparing their results to those of other studies. Evaluation metrics had individual purposes to measure depending on the context of use. For example, the table shows that recall and precision are the most-used evaluation metrics to measure the performance of the proposed methods comparing to the other evaluation metrics.

Data Sets Used for Web Service Discovery and Recommendation (RQ2)

The final selected papers used various data sets to validate and evaluate proposed web service discovery or recommendation solutions. These data sets vary based on the method of collection. Some researchers gathered and captured their data sets from famous web service repositories. Other researchers tested their solution based on publicly published benchmarks. We categorized web service data sets based on the method of collection into self-gathered and published benchmarks. Self-gathered data sets refer to web service data sets collected, captured, and crawled from different web service repositories over the Internet for validation self-use. Some researchers share their collected web service data sets. For self-gathering of web service data sets, researchers used several websites to capture and crawl web services. These included Seekda,[4] PW,[5] WebserviceX,[6] WebserviceList,[7] xMethods,[8] Salcentral, and Titan Search Engine. However, some of the websites are no longer available, including Seekda, WebserviceList, Salcentral, and xMethods.

The existence of published benchmarks helps researchers to validate their solutions without the need to collect web

---

[4] http://www.seekda.com.

[5] https://www.programmableweb.com.

[6] http://www.webservicex.net.

[7] http://www.webservicelist.com.

[8] http://www.xmethods.com/.

Table 17 Evaluation metrics

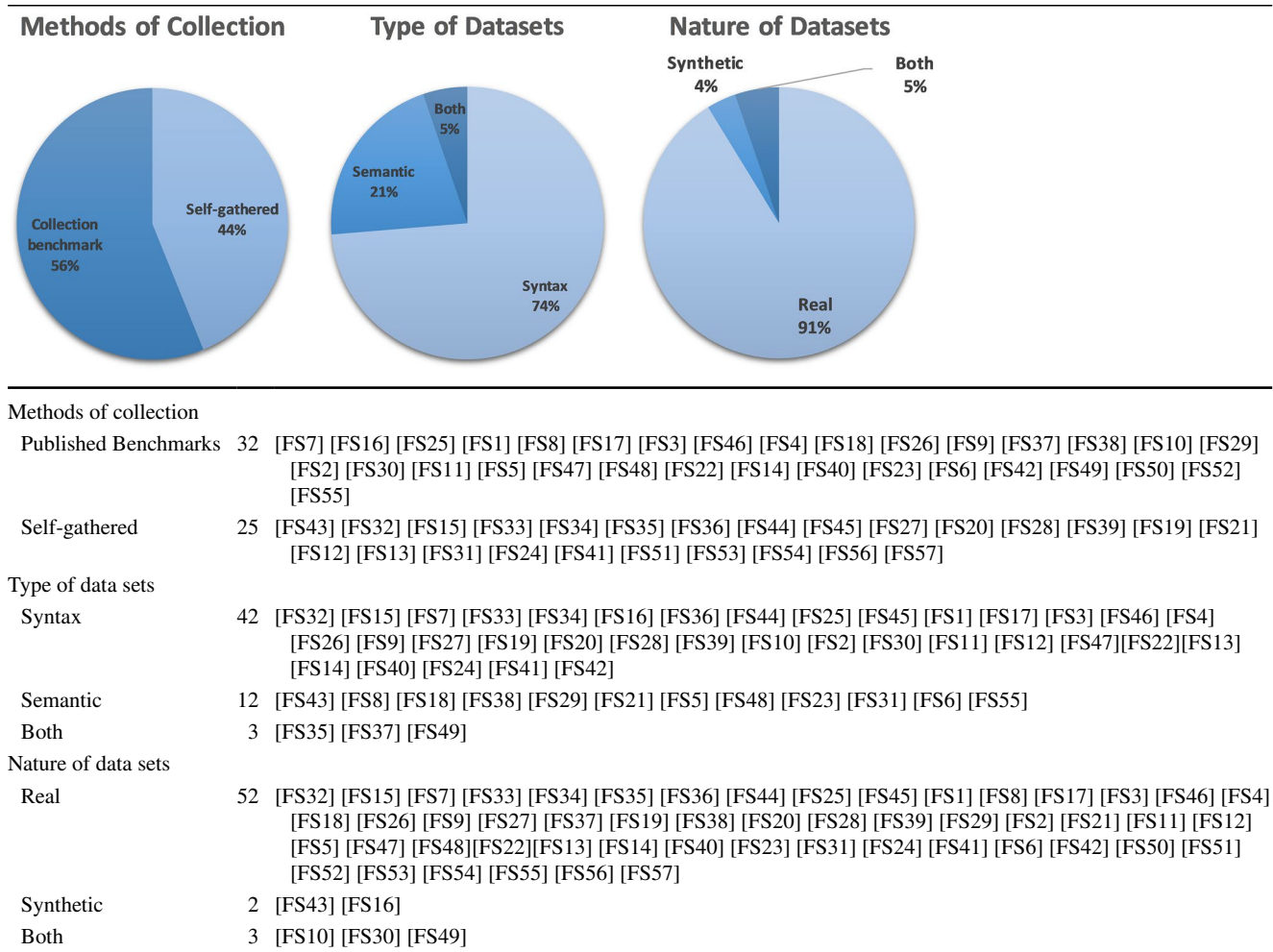| Evaluation metrics	#used	Papers
Recall	31	[FS43] [FS15] [FS7] [FS34] [FS36] [FS44] [FS25] [FS45] [FS3] [FS46] [FS18] [FS27] [FS37] [FS19] [FS38] [FS20] [FS39] [FS2] [FS5] [FS48] [FS22] [FS31] [FS24] [FS6] [FS42] [FS49] [FS51] [FS53] [FS54] [FS55] [FS56]
Precision	32	[FS15] [FS7] [FS34] [FS35] [FS36] [FS44] [FS25] [FS45] [FS3] [FS46] [FS18] [FS27] [FS37] [FS19] [FS20] [FS39] [FS10] [FS29] [FS2] [FS5] [FS48] [FS22] [FS31] [FS24] [FS6] [FS42] [FS49] [FS51] [FS53] [FS54] [FS55] [FS56]
Entropy	7	[FS32] [FS27] [FS21] [FS12] [FS51] [FS55] [FS56]
F-measure	16	[FS32] [FS15] [FS33] [FS34] [FS8] [FS27] [FS37] [FS20] [FS39] [FS29] [FS5] [FS22] [FS14] [FS49] [FS55] [FS57]
Precision@N	6	[FS16] [FS44] [FS8] [FS10] [FS2] [FS14]
Recall@N	1	[FS14]
Mean absolute error (MAE)	11	[FS1] [FS4] [FS26] [FS9] [FS21] [FS30] [FS11] [FS22] [FS40] [FS50] [FS52]
Normalized mean absolute error (NMAE)	5	[FS1] [FS17] [FS9] [FS11] [FS40]
Purity	7	[FS32] [FS46] [FS27] [FS21] [FS12] [FS51] [FS56]
Time	4	[FS43] [FS18] [FS38] [FS48]
Accuracy	6	[FS7] [FS25] [FS38] [FS47] [FS13] [FS42]
Silhouette	2	[FS29] [FS47]
Quantisation error (QE)	1	[FS13]
Dunn Index	1	[FS23]
Root-mean-square error (RMSE)	4	[FS17] [FS22] [FS50] [FS52]
S@K	1	[FS2]
Topographic error (TE)	1	[FS13]
Intra-cluster variance (ICV)	1	[FS23]
Normalized discounted cumulative gain (NDCGn)	1	[FS8]
Median relative error (MRE)	2	[FS17] [FS40]
Neuron utilization (NU)	1	[FS13]
Average item-cluster similarity (AICS)	1	[FS23]
Normalized mutual information (NMI)	1	[FS46]

Table 18 Published benchmark details

	Name	Collection data sets/benchmark	Type	Purpose	Papers
1	WS2003	http://www.andreas-hess.info/projects/annotator/ws2003.html	WSDL	Discovery	[FS25] [FS42]
2	WS-DREAM	http://wsdream.github.io/	QoS	Recommendations	[FS1][FS17] [FS4] [FS26] [FS9] [FS30] [FS11] [FS14] [FS40][FS50] [FS52]
3	SAWSDL-TC	http://projects.semwebcentral.org/projects/sawsdl-tc/	SAWSDL	Discovery	[FS8] [FS23]
4	STag 1.0	http://www.zjujason.com/data.html	WSDL,Tags	Discovery	[FS3] [FS46] [FS2]
5	OWLS-TC	http://projects.semwebcentral.org/projects/owls-tc/	OWLS	Discovery	[FS18] [FS37] [FS38] [FS29] [FS5] [FS47] [FS48] [FS31] [FS6] [FS49] [FS55]
6	MovieLens	https://grouplens.org/datasets/movielens/	Movies	Recommendations	[FS10] [FS14]
7	QWS	http://www.uoguelph.ca/~qmahmoud/qws/	WSDL QoS	Recommendations	[FS22]

Table 19 Used data sets in the final selected papers

Final selected papers

Methods of Collection	Type of Datasets	Nature of Datasets

Methods of collection

Published Benchmarks	32	[FS7] [FS16] [FS25] [FS1] [FS8] [FS17] [FS3] [FS46] [FS4] [FS18] [FS26] [FS9] [FS37] [FS38] [FS10] [FS29] [FS2] [FS30] [FS11] [FS5] [FS47] [FS48] [FS22] [FS14] [FS40] [FS23] [FS6] [FS42] [FS49] [FS50] [FS52] [FS55]
Self-gathered	25	[FS43] [FS32] [FS15] [FS33] [FS34] [FS35] [FS36] [FS44] [FS45] [FS27] [FS20] [FS28] [FS39] [FS19] [FS21] [FS12] [FS13] [FS31] [FS24] [FS41] [FS51] [FS53] [FS54] [FS56] [FS57]

Type of data sets

Syntax	42	[FS32] [FS15] [FS7] [FS33] [FS34] [FS16] [FS36] [FS44] [FS25] [FS45] [FS1] [FS17] [FS3] [FS46] [FS4] [FS26] [FS9] [FS27] [FS19] [FS20] [FS28] [FS39] [FS10] [FS2] [FS30] [FS11] [FS12] [FS47][FS22][FS13] [FS14] [FS40] [FS24] [FS41] [FS42]
Semantic	12	[FS43] [FS8] [FS18] [FS38] [FS29] [FS21] [FS5] [FS48] [FS23] [FS31] [FS6] [FS55]
Both	3	[FS35] [FS37] [FS49]

Nature of data sets

Real	52	[FS32] [FS15] [FS7] [FS33] [FS34] [FS35] [FS36] [FS44] [FS25] [FS45] [FS1] [FS8] [FS17] [FS3] [FS46] [FS4] [FS18] [FS26] [FS9] [FS27] [FS37] [FS19] [FS38] [FS20] [FS28] [FS39] [FS29] [FS2] [FS21] [FS11] [FS12] [FS5] [FS47] [FS48][FS22][FS13] [FS14] [FS40] [FS23] [FS31] [FS24] [FS41] [FS6] [FS42] [FS50] [FS51] [FS52] [FS53] [FS54] [FS55] [FS56] [FS57]
Synthetic	2	[FS43] [FS16]
Both	3	[FS10] [FS30] [FS49]

services. In the final selected papers, web services' published benchmarks have been used more than self-gathered web services'. Table 18 outlines and details eight web services' published benchmarks. The current locations of the used published benchmarks are also indicated in the table. [FS25] and [FS42] used a WSDL-based published benchmark named WS2003 for discovery purposes. WS2003 [66] used it for annotation of web services using machine learning with WSDL files gathered from Salcentral and XMethods. [FS8] and [FS23] used a semantic-based published benchmark named SAWSDL-TC for web service discovery. [FS3], [FS46], and [FS2] used a WSDL-based data set annotated with user tags named STag 1.0 for discovery purposes. [FS18], [FS37], [FS38], [FS29], [FS5], [FS47], [FS48], [FS31], and [FS6] used a semantic-based published benchmark named OWLS-TC for web service discovery. WS2003, SAWSDL-TC, STag 1.0, and OWLS-TC are published benchmarks and test collections that have been used in the final selected papers for discovery

purposes. For web service recommendation, [FS1], [FS17], [FS4], [FS26], [FS9], [FS30], [FS11], [FS14], and [FS40] used the web service QoS data set named WS-DREAM. [FS10] and [FS14] adopted the famous movies recommendation data set MovieLens and [FS22] used the quality of web service (QWS) data set for web service recommendation. Figure 9 illustrates the usage of published benchmarks in the final selected papers.

Table 19 outlines the methods for web service collection, types of data sets, and nature of data sets used in the final selected papers. Fifty-six percent of the final selected papers used known published benchmarks to validate their proposed solutions. Alternately, 44% of the final selected papers used self-gathering to collect web service data sets for validation purposes. The type of web service data sets can be syntax, semantic, or both. Based on the DL used, researchers tend to use syntax-based web services more than semantic-based ones to validate and test their solutions. However, some researchers used both syntax and semantics

Table 20 Future research vs. challenges addressed

	Suggestions for future research	Challenges addressed
1	Comprehensive and neutral approaches Using NLP, IR, text mining, tagging, data mining, information extraction	Semantics extraction and extension Minimize search space (clustering) Add extra data/words to web services
2	Social network data and network analysis techniques	Additional information about service users to improve clustering and recommendation Detect correlation and similarities between service users Detect user behavior patterns Enrich user context
3	Big data analytics tools and techniques	Increase clustering speed Online service clustering and processing Online web service discovery including crawling and processing to delete duplicates of web service files Online QoS perdition

to validate their work. The nature of web service data sets can be from real-world web services, synthetic data sets for validation purposes only, or a combination of both real and synthetic. Real-world web service data sets are used more by researchers compared to other types.

Web Service Discovery and Recommendation Trends and Future Directions (RQ3)

The last decade witnessed the growth of web services as providers developed and published an enormous number of Web-based services. This made it a challenge to discover services in a broad and diverse web service space. The discovery problem is inherently difficult because of the large-scale SOA systems, including web services, mobile services, cloud services, and cloud computing. These services are dynamic, changing, and uncertain in nature. The scope of services in SOA is changing rapidly as new services are added and old ones are deleted or modified. New standards emerge as old ones are abandoned by developers. Web service discovery and recommendation methods and techniques proposed by researchers have been applied separately rather than working in parallel. Web service discovery (matching) and web service recommendation (selection and ranking) need to be integrated to better understand user requirements in locating the right web services. Previous approaches separated web service discovery from web service recommendation. We argue that the cause of this is the need to find the right web service data sets to apply both discovery and recommendation to. However, matching and selecting the right web services should be considered together and in parallel in this process. Another gap is the use of one type of similarity measure to find similar web services in the process of web service clustering or discovery. The limitations of syntactic-based similarity measure can be overcome or minimized if combined with another similarity measure, such as distance, semantic, or context similarity.

Clustering techniques can be used to reduce the search space when looking for the right web services. However, it can be time-consuming to cluster web services based on different similarities. New parallel and distributed data processing tools and techniques (e.g., Apache Hadoop) can be used to group web services and increase or enhance clustering speed. One example is its use in the k-means parallel clustering algorithm to address web service discovery and prediction problems with big data circumstances. Another gap is found in achieving a domain-specific web service clustering while considering users and service context. The performance of web service clustering can be improved with additional features and web service information. As an example, web service tag information can be utilized to improve the performance of web service clustering. Most of the proposed approaches investigate offline web service clustering. Real-time or online web service clustering provides a better solution and current web service information. QoS parameters are an important aspect of the process of web service ranking, selection, and recommendation. This is not considered in major web service discovery works. Detecting malicious and outlier users with inaccurate QoS information and noise QoS improves prediction. Finding a complete solution for web service discovery and recommendation starting with user query to matching and retrieving the requested web service and finally selecting and ranking web service is rare to find in such an environment where most proposed solutions have enhanced just one part of the overall process. Finding a complete web service discovery and recommendation system in more detail is a gap. Semantic-based web service approaches need to provide service requestors with step-by-step semantically guided methods to locate the target web service components to minimize the complexity.

We have identified three areas of trends and future directions for web service discovery and recommendation. Table 20 outlines the three areas and the challenges to be addressed:

- Comprehensive and Neutral web Service Discovery and Recommendation: The discovery process of web services focuses on WSDL-based or semantic-based web services, neglecting the fact that most web services are described in plain text. Web service discovery approaches should extend their methods to be comprehensive and neutral to work with any web service description. Most web services (web APIs) are described in plain text. Therefore, NLP, text and DM, IR, collaborative tagging, and service information extraction techniques can be used to identify useful patterns, clusters, and group-related web services and extract useful semantics without focusing on particular form of web services description language.

- Social Network web Service Discovery and Recommendation: web service discovery and recommendation can be enhanced using social network data to detect hidden relationships between services and users to generate recommendations. Information and interactions can improve the process of web service discovery and recommendation by utilizing social network data through user activities. For example, user experiences, ratings, questions, and feedback posted on the social network regarding web services or any other topics can provide a better understanding of user requirements. Social network and network analysis tools and techniques can boost and improve the discovery and recommendation of web services.

- Big Data Analytics-driven web Service Discovery and Recommendation: There are limitations to the proposed web service discovery and recommendation approaches. These can be overcome using big data analytics tools and techniques. Clustering web service approaches can benefit from running the parallel clustering algorithm to speed up the process of clustering. Furthermore, online web service discovery and recommendation provide a new research direction to take advantage of the development of algorithms and models for processing data online in big data ecosystems. It provides a way to develop more scalable and adaptable discovery and recommendation solutions in a large-scale and changing environment.

Discussion

Web service discovery and recommendation methods have been proposed in different studies using various techniques to facilitate the process of web service selection and invocation for users. By analyzing the final selected papers, we found that most approaches focus exclusively on either functional or non-functional clustering of web services, and that there is a dearth of research that combines these two properties. A possible reason could be due to the fact that combining both functional and non-functional properties in the clustering process may increase complexity. One of the challenges in building web service discovery and recommendation solutions is finding the right web services data sets that meet the requirements of the proposed method. Some proposed methods use real web services data set to test their solution; others had to add synthetic data to meet the required set of input for their proposed solution. The existence of new web services data sets and collection benchmarks can help the community produce web service discovery and recommendation solutions, and test and compare their results with other solutions. In general, syntactic-based approaches, where syntactic similarity measures are used, provide a straightforward solution with low performance, contrary to semantic-based approaches, where semantic measures are used, which provide a more complex solution with a better performance. Hybrid-based approaches, which combine both syntactic and semantic similarity measures, provide a better solution with high performance in finding web services. However, complexity and processing time will be increased.

In terms of implications for research, we provide a synthesis on the current proposed methods for web service discovery and recommendation based on systematic procedures. We contribute to the theory of web service discovery and recommendation by providing a classification of the approaches employed based on the selected paper. Furthermore, we provide a list of algorithms for clustering and association rules, similarity measures, and evaluation metrics. This provides future researchers with insights on algorithms, similarity measures, and evaluation metrics used, and the gaps for future research. Moreover, we identified the most-used data sets for web service discovery and recommendation with their locations and classified web service data sets based on their types, methods of collection, and their nature. Finally, we identified the trends and future directions for web services discovery and recommendations.

In terms of implication for practice, finding a complete solution for web service discovery and recommendation, which starts with users sending a query to match, retrieve, and finally select and rank the list of matched web services, is rare to find in the proposed solutions. However, a complete solution that provides a combination of web service discovery and recommendations methods can be exist in practice in a real working environment. This solution should be able to understand the user's requirements from both functional and non-functional perspectives, and incorporate the requirements into a discovery and recommendation solution. The solution should be able to analyze the user requirements and match against the list of services. Based on our SLR, to overcome the limitation of short description of web service (with too little words), and to improve clustering performance, we suggest the use of semantic extensions and extraction techniques, in addition to adding extra words in

the web services data sets during the pre-processing step. Furthermore, the use of users' data from social networks will add more information about service users, which will reflect in finding the correlation between service users, their patterns, and context. The use of big data analytics tools and techniques such as Hadoop and Spark can improve the performance of clustering and speed up the processing time.

Threats to Validity

This SLR provides a classification of clustering and association rule techniques for web service discovery and recommendation. It identifies methods, algorithms, similarity measures, and evaluation metrics. The review was carried out to provide the current state-of-the-art information in the area of clustering and association rules for web service discovery and recommendation based on systematic procedures. The focus was finding answers to predefined research questions from the final selected papers. We identified 57 final selected papers with specific quality requirements. We extracted the data to answer the research questions, and we believe that significant work remains to improve the current state of research in the area of clustering and association rules for web service discovery and recommendation. However, validity is a primary concern in such empirical studies. We discuss the limitations to construct, internal, and external validities [67].

Construct validity looks at whether the implementation of SLR matches the initial objectives. We identify the search and selection processes as possible limitations. Search terms and keywords were derived from the population and intervention used in forming the research questions. They were tested against a well-known list of research studies. However, the completeness and thoroughness of the terms are not assured. To mitigate the risk, we snowballed the references of the list of selected studies. The search identified two studies in the Chinese language, which were excluded. In addition, ten studies were excluded, because we were unable to access the full text. This may present a threat to construct validity. Although well-known digital libraries were used to search for the selected studies, other digital libraries may contain relevant studies that have not been taken into consideration.

Internal validity is the extent to which the design and conduct of the study can prevent systematic error and work as a prerequisite for external validity [1]. Some concerns about data extraction are worth highlighting here. In the process of data extraction from the final selected studies, we rely on our interpretation and analysis when necessary data are not clearly stated. Some of the required data in the process of extraction were missing in the final selected studies. This may potentially affect the internal validity of the study.

External validity is the extent to which the effects observed in the study are applicable outside [1]. It is about the generalizability and applicability of the study outcomes. The research questions and quality assessment reduced the risk of generalizability of the outcomes. Furthermore, the SLR is focused on being controlled by a focus problem related to clustering and association rule techniques for web service discovery and recommendation. This was at a predefined time (2006–2017), and the problem was not previously investigated. When considering this information, we consider the outcome of the SLR to be generalizable and applicable. It is worth noting that all steps for this SLR including the discussions and decisions about the SLR search and selection protocols were done collaboratively among the authors. None of the final selected papers were co-authored by us or by anyone related to our affiliations. The research questions were designed, while multiple authors provided their feedback to improve the final research questions. The search query was resolved collaboratively, while each author verified the results. Furthermore, the development of the inclusion and exclusion criteria was conducted collaboratively, and conflicts were resolved by discussion.

Conclusions

The reported SLR provides a classification of clustering and association techniques for web service discovery and recommendation, and it identifies methods, algorithms, similarity measures, data sets, and evaluation metrics. The review was carried out to provide the current state-of-the-art in the area of clustering and association rules for web service discovery and recommendation based on systematic procedures. The focus was finding answers to the predefined research questions from the final selected papers. We identified 57 final selected papers with specific quality requirements and extracted data to answer the research questions. Our SLR reveals that clustering techniques are the most in the literature and have been mostly used for web service discovery. As such, there is a dearth of academic literature where association rule techniques have been used for web service discovery and recommendation. Furthermore, K-means and HAC are the most common clustering algorithms, and most studies have used these techniques in combination with semantic or syntactic similarity measures. We believe that much work remains to improve the current state of research in the area of clustering and association rules for web service discovery and recommendation. Future studies should investigate online live web services which broaden the basis of discovery and recommendation by including various type of web service descriptions including plain text description that are currently used in web APIs. There is also an opportunity to improve web service discovery and recommendation by

utilizing modern techniques based on big data analytics and social network analysis. Finally, future research could also benefit from harvesting current and generating new web service description data sets for benchmarking and evaluating web service discovery and recommendation solutions.

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

Appendix A: The Search Queries for Each Selected Digital Library

ACM Digital Library: Does not support abstract, title, and keywords search.

"query": {("Web Service*" "Web Service* Discovery" "Web Service* Recommend*" "Clustering" "Association Rule*")}

"filter": {"publicationYear":{"gte":2006}},

{owners.owner = HOSTED}

IEEE Digital Library: Does not support abstract, title, and keywords search.

"Web Service*" OR "Web Service* Discovery" OR "Web Service* Recommend*") AND ("Clustering" OR "Association Rule*") and refined by Year: 2006-2016

SpringerLink: Does not support abstract, title, and keywords search.

'("Web Service*" OR "Web Service* Discovery" OR "Web Service* Recommend*") AND ("Clustering" OR "Association Rule*")'

within English Computer Science Article 2006 2016

ISI Web of Science: support abstract, title, and keywords search.

TOPIC: (("Web Service*" OR "Web Service* Discovery" OR "Web Service* Recommend*") AND ("Clustering" OR "Association Rule*"))

Timespan = 2006–2016

ScienceDirect: support abstract, title, and keywords search.

pub-date > 2006 and TITLE-ABS-KEY ("Web Service?" OR "Web Service? Discovery" OR "Web Service? Recommend*") AND TITLE-ABS-KEY ("Clustering" OR "Association Rule?")

Scopus: support abstract, title, and keywords search.

TITLE-ABS-KEY (("Web Service*" OR "Web Service* Discovery" OR "Web Service* Recommend*") AND ("Clustering" OR "Association Rule*")) AND PUBYEAR > 2006 AND (LIMIT-TO (LANGUAGE, "English")) AND (EXCLUDE (PUBYEAR, 2017))

Appendix B: List of Final Selected Papers

ID	Title	Venues	Year	Citation	Quality Score
[FS1]	TAP: A personalized trust-aware QoS prediction approach for web service recommendation [50]	KNOSYS (J)	2016	0	8.5
[FS2]	Clustering web services to facilitate service discovery [22]	KAIS (J)	2014	21	8.5
[FS3]	Modeling and exploiting tag relevance for web service mining [51]	KAIS (J)	2014	9	8
[FS4]	Web Service Recommendation via Exploiting Location and QoS Information [15]	TPDS (J)	2014	26	8
[FS5]	Web service discovery among large service pools utilizing semantic similarity and clustering [44]	EIS (J)	2015	1	8
[FS6]	Characterization and search of web services through intensional knowledge [56]	JIIS (J)	2016	1	8
[FS7]	Automatic Tagging Web Services Using Machine Learning Techniques [59]	WI-IAT (C)	2014	4	7.5
[FS8]	Leveraging Formal Concept Analysis with Topic Correlation for Service Clustering and Discovery [30]	ICWS(C)	2014	12	7.5
[FS9]	Location-based Hierarchical Matrix Factorization for Web Service Recommendation [52]	ICWS(C)	2014	11	7.5
[FS10]	Collaborative personal profiling for web service ranking and recommendation [58]	ISF(J)	2015	9	7.5
[FS11]	CluCF: a clustering CF algorithm to address data sparsity problem [49]	SOCA (J)	2016	1	7.5
[FS12]	Transferring auxiliary knowledge to enhance heterogeneous web service clustering [24]	IJHPCN (J)	2016	0	7.5

ID	Title	Venues	Year	Citation	Quality Score
[FS13]	Clustering facilitated web services discovery model based on supervised term weighting and adaptive metric learning [46]	IJWET (J)	2013	5	7.5
[FS14]	Recommending web service based on user relationships and preferences [61]	ICWS (C)	2013	7	7.5
[FS15]	Context-Aware Post-filtering for Web Service Clustering [34]	SCC (C)	2014	1	7
[FS16]	Personalized Web Service Ranking via User Group Combining Association Rule [57]	ICWS (C)	2009	49	7
[FS17]	User-QoS-based Web Service Clustering for QoS Prediction [47]	ICWS (C)	2015	0	7
[FS18]	Service discovery acceleration with hierarchical clustering [21]	ISF(J)	2015	5	7
[FS19]	Co-clustering WSDL Documents to Bootstrap Service Discovery [38]	SOCA (C)	2014	2	7
[FS20]	Service Recommendation Using Customer Similarity and Service Usage Pattern [48]	ICWS (C)	2015	1	7
[FS21]	Time-aware Semantic Web Service Recommendation [53]	SCC (C)	2015	1	7
[FS22]	Collaborative filtering-based hybrid approach for web service recommendations [60]	RJASET (J)	2014	0	7
[FS23]	Particle swarm optimization for clustering semantic web services [42]	ISPDC (C)	2011	5	7
[FS24]	Clustering WSDL documents to bootstrap the discovery of web services [12]	ICWS (C)	2010	193	7
[FS25]	Web Services Discovery Based on Latent Semantic Approach [29]	ICWS (C)	2009	12	6.5
[FS26]	A Hierarchical Matrix Factorization Approach for Location-based Web Service QoS Prediction [14]	SOSE (C)	2014	0	6.5
[FS27]	Web Service Clustering using a Hybrid Term-Similarity Measure with Ontology Learning [35]	IJWSR (J)	2014	5	6.5
[FS28]	Web Service Clustering using Multidimensional Angles as Proximity Measures [40]	TOIT (J)	2009	85	6.5
[FS29]	Semantic-Based Clustering of Web Services [43]	JWE (J)	2015	0	6.5
[FS30]	QoS-aware service selection via collaborative QoS evaluation [54]	WWW (J)	2014	6	6.5
[FS31]	Taxonomic clustering and query matching for efficient service discovery [37]	ICWS (C)	2011	19	6.5
[FS32]	Leveraging Auxiliary Knowledge for Web Service Clustering [23]	CIE (J)	2016	0	6
[FS33]	Semantic-Based Automated Service Discovery [62]	TSC (J)	2012	42	6
[FS34]	Ontology-learning method for web services clustering [27]	ICCCT (C)	2012	0	6
[FS35]	Cluster-Based Web Service Recommendation [13]	SCC (C)	2016	0	6
[FS36]	Similarity analysis of service descriptions for efficient web service discovery [25]	DSAA (C)	2014	1	6
[FS37]	Ontology Learning with Complex Data Type for Web Service Clustering [36]	CIDM (C)	2014	0	6
[FS38]	Web Service Clustering Using Relational Database Approach [39]	IJSEKE (J)	2015	1	6
[FS39]	Improving REST Service Discovery with Unsupervised Learning Techniques [41]	CISIS (C)	2015	1	6

ID	Title	Venues	Year	Citation	Quality Score
[FS40]	A clustering-based QoS prediction approach for web service recommendation [16]	ISORCW (C)	2012	7	6
[FS41]	Web service clustering using text-mining techniques [11]	IJAOSE (J)	2009	93	6
[FS42]	Efficiently finding web services using a clustering semantic approach [31]	CSSSIA (C)	2008	77	6
[FS43]	Research on web service discovery with semantics and clustering [32]	ITAIC (C)	2011	20	5.5
[FS44]	A concept analysis approach for guiding users in service discovery [28]	SOCA (C)	2012	4	5.5
[FS45]	DaaS: Cloud-based mobile web service discovery [33]	PMC (J)	2013	22	5.5
[FS46]	WS-HFS: A Heterogeneous Feature Selection Framework for Web Services Mining [55]	ICWS (C)	2015	1	5.5
[FS47]	Learning Sparse Functional Factors for Large-Scale Service Clustering [26]	ICWS (C)	2015	3	5.5
[FS48]	Hierarchical clustering-based web service discovery [45]	ICISO (C)	2014	1	5.5
[FS49]	QoS-Aware Service Clustering to Bootstrap the Web Service Selection [68]	SCC (C)	2017	0	6
[FS50]	A new QoS-aware web service recommendation system based on contextual feature recognition at server-side [69]	TNSM (J)	2017	0	7
[FS51]	Mashup service clustering based on an integration of service content and network via exploiting a two-level topic model [70]	ICWS (C)	2016	6	7
[FS52]	Domain-aware Mashup service clustering based on LDA topic model from multiple data sources [71]	IST(J)	2017	0	6.5
[FS53]	Leveraging Track Relationships for Web Service Recommendation [72]	ICEBE (C)	2016	1	6
[FS54]	A Web Service Discovery Approach Based on Mining Underlying Interface Semantics [73]	TKDE (J)	2017	2	7
[FS55]	Improving Web Service Clustering through a Novel Ontology Generation Method by Domain Specificity [74]	ICWS (C)	2017	0	7
[FS56]	WE-LDA: A Word Embeddings Augmented LDA Model for Web Services Clustering [75]	ICWS (C)	2017	0	7.5
[FS57]	Correlation-Aware Multi-Label Active Learning for Web Service Tag Recommendation [76]	ICWS (C)	2017	0	6.5

References

1. Kitchenham B, Charters S, Guidelines for performing Systematic Literature reviews in Software Engineering, 2007.
2. Obidallah WJ, Ruhi U, Raahemi B, Current landscape of web service discovery: a typology based on five characteristics, In: 2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI), 2016. pp. 678–683.
3. Crasso M, Zunino A, Campo M. Easy web service discovery: a query-by-example approach. Sci Comput Program. 2008;71(2):144–64.
4. Ngane LD, Goh A, Tru CH, A survey of web service discovery systems. In: Alkhatib GI, Rine DC, editors. IGI Global; 2008.
5. M. Klusch, Service discovery, Encyclopedia of Social Networks Min. R. Alhajj J. Rokne, editors. New York:Springer; 2014.
6. Garofalakis A, Panagis J, Sakkopoulos Y, Tsakalidis E, Web service discovery mechanisms: looking for a needle in a haystack? In: International workshop on web engineering; 2004, vol. 5, no. 3. pp. 265–290.
7. Zhang M, Liu X, Zhang R, Sun H A web service recommendation approach based on QoS prediction using fuzzy clustering. In: 2012 IEEE ninth international conference on services computing; 2012, pp. 138–145.
8. Le DN, Soong Goh AE. A survey of web service discovery systems. Int J Inf Technol Web Eng. 2007;2(2):65–80.
9. Newcomer E, Understanding Web Services: XML, Wsdl, Soap, and UDDI. Addison-Wesley Professional; 2002.
10. Obidallah WJ, Raahemi B, A survey on web service discovery approaches. In: Proceedings of the second international conference on internet of things and cloud computing—ICC '17; 2017, pp. 1–8.
11. Liu W, Wong W. Web service clustering using text mining techniques. Int. J. Agent-Oriented Softw. Eng. 2009;3(1):6.
12. Elgazzar K, Hassan AE, Martin P, Clustering WSDL documents to bootstrap the discovery of Web services. In: ICWS 2010–2010 IEEE 8th international conference on web services; 2010, pp. 147–154.
13. Kumara BTGS, Paik I, Siriweera THAS, Koswatte KRC, Cluster-based web service recommendation. In: 2016 IEEE international conference on services computing (SCC); 2016, pp. 348–355.
14. He P, Zhu J, Xu J, Lyu MR, A hierarchical matrix factorization approach for location-based web service QoS prediction. In: 2014 IEEE 8th international symposium on service oriented system engineering; 2014, pp. 290–295.
15. Chen X, Zheng Z, Yu Q, Lyu MR. Web service recommendation via exploiting location and QoS information. IEEE Trans Parallel Distrib Syst. 2014;25(7):1913–24.
16. Zhu J, Kang Y, Zheng Z, Lyu MR, A clustering-based QoS prediction approach for web service recommendation. In: 2012 IEEE 15th international symposium on object/component/service-oriented real-time distributed computing workshops; 2012, pp. 93–98.
17. Kitchenham B, Procedures for performing systematic reviews, Keele: Keele University; 2004, vol. 33, no. TR/SE-0401, p. 28.
18. Mavergames C, Covidence (Systematic Review Software); 2013. https://www.covidence.org/. Accessed 07 Dec 2016.
19. Kitchenham B, Pearl Brereton O, Budgen D, Turner M, Bailey J, Linkman S. Systematic literature reviews in software engineering—a systematic literature review. Inf Softw Technol. 2009;51(1):7–15.
20. Dybå T, Dingsøyr T. Empirical studies of agile software development: a systematic review. Inf Softw Technol. 2008;50(9–10):833–59.
21. Cong Z, Fernandez A, Billhardt H, Lujak M. Service discovery acceleration with hierarchical clustering. Inf Syst Front. 2015;17(4):799–808.
22. Wu J, Chen L, Zheng Z, Lyu MRMR, Wu Z. Clustering Web services to facilitate service discovery. Knowl Inf Syst. 2014;38(1):207–29.
23. Tian G, et al. Leveraging auxiliary knowledge for web service clustering. Chin J Electron. 2016;25(5):858–65.
24. Tian G, Sun C, He K-Q, Ji X-M. Transferring auxiliary knowledge to enhance heterogeneous web service clustering. Int J High Perform Comput Netw. 2016;9(1/2):160.
25. Sowmya Kamath S, Ananthanarayana VS, Similarity analysis of service descriptions for efficient Web service discovery. In: 2014 International Conference on Data Science and Advanced Analytics (DSAA), 2014, pp. 142–148.
26. Yu Q, Wang H, Chen L, Learning sparse functional factors for large-scale service clustering. In: 2015 IEEE international conference on web services; 2015, pp. 201–208.
27. Kumara BTGS, Paik I, Lee G, Ontology learning method for Web services clustering. In: Proceedings—2012 7th international conference on computing and convergence technology (ICCIT, ICEI and ICACT), ICCCT 2012; 2012, pp. 705–710.
28. Upadhyaya B, Khomh F, Zou Y, Lau A, Ng J, A concept analysis approach for guiding users in service discovery. In: 2012 Fifth IEEE international conference on service-oriented computing and applications (SOCA); 2012, pp. 1–8.
29. Ma J, Zhang Y, He J, Web services discovery based on latent semantic approach. In: 2008 IEEE international conference on web services; 2008, pp. 740–747.
30. Aznag M, Quafafou M, Jarir Z, Leveraging formal concept analysis with topic correlation for service clustering and discovery. In: 2014 IEEE international conference on web services; 2014, pp. 153–160.
31. Ma J, Zhang Y, He J, Efficiently finding web services using a clustering semantic approach. In: Proceedings of the 2008 international workshop on Context enabled source and service selection, integration and adaptation organized with the 17th International World Wide Web Conference (WWW 2008)—CSSSIA '08; 2008, vol. 292, pp. 1–8.
32. Wen T, Sheng G, Li Y, Guo Q, Research on web service discovery with semantics and clustering. In: 2011 6th IEEE joint international information technology and artificial intelligence conference; 2011, vol. 1, pp. 62–67.
33. Elgazzar K, Hassanein HS, Martin P. DaaS: cloud-based mobile web service discovery. Pervasive Mob Comput. 2014;13:67–84.
34. Kumara BTGS, Paik I, Ohashi H, Chen W, Koswatte KRC, Context aware post-filtering for web service clustering. In: 2014 IEEE international conference on services computing; 2014, pp. 440–447.
35. Kumara BTGSBTGS, Paik II, Chen WW, Ryu KHKH. Web service clustering using a hybrid term-similarity measure with ontology learning. Int J Web Serv Res. 2014;11(2):24–45.
36. Kumara BTGS, Paik I, Koswatte KRC, Chen W, Ontology learning with complex data type for Web service clustering. In: IEEE SSCI 2014–2014 IEEE symposium series on computational intelligence—CIDM 2014: 2014 IEEE symposium on computational intelligence and data mining, proceedings; 2014, pp. 129–136.
37. Dasgupta S, Bhat S, Lee Y, Taxonomic clustering and query matching for efficient service discovery. In: 2011 IEEE international conference on web services; 2011, pp. 363–370.
38. Liang T, Chen L, Ying H, Wu J, Co-clustering WSDL documents to bootstrap service discovery. In: 2014 IEEE 7th international conference on service-oriented computing and applications; 2014, pp. 215–222.

39. Liu J, Liu F, Li X, He K, Ma Y, Wang J. Web service clustering using relational database approach. Int J Softw Eng Knowl Eng. 2015;25(08):1365–93.

40. Platzer C, Rosenberg F, Dustdar S. Web service clustering using multidimensional angles as proximity measures. ACM Trans Internet Technol. 2009;9(3):1–26.

41. Rodriguez JM, Zunino A, Mateos C, Segura FO, Rodriguez E, Improving REST service discovery with unsupervised learning techniques. In: 2015 ninth international conference on complex, intelligent, and software intensive systems; 2015, pp. 97–104.

42. Nagy A, Oprisa C, Salomie I, Pop CB, Chifu VR, Dinsoreanu M, Particle swarm optimization for clustering semantic web services. In: 2011 10th international symposium on parallel and distributed computing; 2011, pp. 170–177.

43. De Jesus Silva L, Claro DB, Lopes DCP. Semantic-based clustering of web services. J Web Eng. 2015;14(3–4):325–45.

44. Chen F, Li M, Wu H, Xie L. Web service discovery among large service pools utilising semantic similarity and clustering. Enterp Inf Syst. 2017;11(3):452–69.

45. Gao H, Wang S, Sun L, Nian F, Hierarchical clustering based web service discovery. In: IFIP advances in information and communication technology, vol. 426; 2014, pp. 281–291.

46. Chen LL, Yang GG, Zhu WW, Zhang YY, Yang ZZ. Clustering facilitated web services discovery model based on supervised term weighting and adaptive metric learning. Int J Web Eng Technol. 2013;8(1):58.

47. Chen F, Yuan S, Mu B, User-QoS-Based web service clustering for QoS prediction. In: 2015 IEEE international conference on web services; 2015, pp. 583–590.

48. Liu R, Xu X, Wang Z, Service recommendation using customer similarity and service usage pattern. In: 2015 IEEE international conference on web services; 2015, pp. 408–415.

49. Yu C, Huang L. CluCF: a clustering CF algorithm to address data sparsity problem. Serv Oriented Comput Appl. 2017;11(1):33–45.

50. Su K, Xiao B, Liu B, Zhang H, Zhang Z. TAP: a personalized trust-aware QoS prediction approach for web service recommendation. Knowl-Based Syst. 2016;115:55–65.

51. Chen L, Wu J, Zheng Z, Lyu MRMR, Wu Z. Modeling and exploiting tag relevance for Web service mining. Knowl Inf Syst. 2014;39(1):153–73.

52. He P, Zhu J, Zheng Z, Xu J, Lyu MR, Location-based hierarchical matrix factorization for web service recommendation. In: 2014 IEEE international conference on web services; 2014, pp. 297–304.

53. Lei Y, Jiantao Z, Junxing Z, Fengqi W, Juan W, Time-aware semantic web service recommendation. In: 2015 IEEE international conference on services computing; 2015, no. PG-664-671, pp. 664–671.

54. Yu Q. QoS-aware service selection via collaborative QoS evaluation. World Wide Web. 2014;17(1):33–57.

55. Chen L, Yu Q, Yu PS, Wu J, WS-HFS: a heterogeneous feature selection framework for web services mining. In: 2015 IEEE international conference on web services; 2015, pp. 193–200.

56. Bianchini D, Garza P, Quintarelli E. Characterization and search of web services through intensional knowledge. J Intell Inf Syst. 2016;47(3):375–401.

57. Rong W, Liu K, Liang L, Rong W, Liu K, Liang L, Personalized web service ranking via user group combining association rule. In: 2009 IEEE international conference on web services; 2009, pp. 445–452.

58. Rong W, Peng B, Ouyang Y, Liu K, Xiong Z. Collaborative personal profiling for web service ranking and recommendation. Inf Syst Front. 2015;17(6):1265–82.

59. Lin M, Cheung DW, Automatic tagging web services using machine learning techniques. In: 2014 IEEE/WIC/ACM international joint conferences on web intelligence (WI) and intelligent agent technologies (IAT); 2014, vol. 2, pp. 258–265.

60. Vadivelou G, Ilavarasan E. Collaborative filtering based hybrid approach for web service recommendations. Res J Appl Sci Eng Technol. 2014;8(5):615–22.

61. Gong M, Xu Z, Xu L, Li Y, Chen L, Recommending Web service based on user relationships and preferences. In: 2013 IEEE 20th international conference on web services; 2013, pp. 380–386.

62. Paliwal AV, Shafiq B, Vaidya J, Xiong H, Adam N. Semantics-based automated service discovery. IEEE Trans Serv Comput. 2012;5(2):260–75.

63. Miller GA. WordNet: a lexical database for English. Commun ACM. 1995;38(11):39–41.

64. Cilibrasi RL, Vitanyi PMB. The google similarity distance. IEEE Trans Knowl Data Eng. 2007;19(3):370–83.

65. Levenshtein VI. Binary codes capable of correcting deletions, insertions, and reversals. Sov Phys Dokl. 1966;10(8):707–10.

66. Heß A, Kushmerick N, Machine learning for annotating semantic web services. In: AAAI spring symposium on semantic web services; 2004.

67. Wohlin C, Runeson P, Höst M, Ohlsson MC, Regnell B, Wesslén A. Experimentation in software engineering, vol. 15. Berlin Heidelberg: Springer; 2012.

68. Kumara BTGS, Paik I, Siriweera THAS, Koswatte KRC, QoS aware service clustering to bootstrap the web service selection. In: 2017 IEEE international conference on services computing (SCC); 2017, pp. 233–240.

69. Li S, Wen J, Luo F, Gao M, Zeng J, Dong ZY. A new QoS-aware web service recommendation system based on contextual feature recognition at server-side. IEEE Trans Netw Serv Manag. 2017;14(2):332–42.

70. Cao B et al., Mashup service clustering based on an integration of service content and network via exploiting a two-level topic model. In: Proceedings—2016 IEEE international conference on web services, ICWS 2016; 2016, pp. 212–219.

71. Cao B, Liu X, Liu J, Tang M. Domain-aware Mashup service clustering based on LDA topic model from multiple data sources. Inf Softw Technol. 2017;90:40–54.

72. Slaimi F, Sellami S, Boucelma O, Ben Hassine A, Leveraging track relationships for web service recommendation. In: 2016 IEEE 13th international conference on e-business engineering (ICEBE); 2016, pp. 220–225.

73. Cheng B, Zhao S, Li C, Chen J. A web services discovery approach based on mining underlying interface semantics. IEEE Trans Knowl Data Eng. 2017;29(5):950–62.

74. Rupasingha RAHM, Paik I, Kumara BTGS, Improving web service clustering through a novel ontology generation method by domain specificity. In: 2017 IEEE international conference on web services (ICWS); 2017. pp. 744–751.

75. Shi M, Liu J, Zhou D, Tang M, Cao B, WE-LDA: a word embeddings augmented LDA model for web services clustering. In: 2017 IEEE international conference on web services; 2017. pp. 9–16.

76. Shi W, Liu X, Yu Q, Correlation-aware multi-label active learning for web service tag recommendation. In: 2017 IEEE International Conference on Web Services (ICWS); 2017. pp. 229–236.