



Self-adaptive mobile web service discovery framework for Dynamic Mobile Environment[☆]

Salisu Garba ^{*}, Radziah Mohamad, Nor Azizah Saadon

School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, 81310 UTM, Skudai, Johor, Malaysia



ARTICLE INFO

Article history:

Received 12 November 2020

Received in revised form 18 August 2021

Accepted 10 October 2021

Available online 19 October 2021

Keywords:

Self-adaptive

Mobile web service

Service discovery

Negative Selection Algorithm

Dynamic Mobile Environment

ABSTRACT

This paper proposes a self-adaptive mobile web service (MWS) discovery framework for a dynamic mobile environment (DME) to deal with MWS proliferation, dynamic context, and irrelevant MWS discovery challenges. The main contribution of this research includes an improvement of the match-making algorithm, enhanced MWS categorization approach, and extensible meta-context ontology that represents the context information in DME. This was achieved by enabling the self-adaptive matchmaker to learn MWS relevance using a Modified-Negative Selection Algorithm (M-NSA) and retrieve the most relevant MWS based on the current context of the discovery. To assess the proposed framework, series of experiments was carried out using publicly-available datasets. The performance of the framework is evaluated against the state-of-the-art frameworks. It was found that the proposed framework is more effective and attained better binary and graded relevance when subjected to context variations which are prevalent in DME. This is useful for service-based application designers and other MWS clients.

© 2021 Elsevier Inc. All rights reserved.

1. Introduction

Mobile Web Service (MWS) is a lightweight subset of conventional web services. It requires adaptable protocols, RESTful description, smaller message formats suitable for deployment in resource-constrained devices in a dynamic mobile environment (DME) (Elgazzar et al., 2014; Scioscia et al., 2014; Ruta et al., 2015). The MWS discovery framework consists of three distinct parts: MWS provisioning, MWS request, and MWS matchmaking (Cheniki et al., 2016). The MWS request describes the user needs to obtain the most relevant MWS for a particular task. The MWS provisioning part is responsible for advertising the availability of MWS in a local service directory by individual MWS providers. The matchmaking part is responsible for determining the degree of match between the advertised MWS and the requested MWS to find the most relevant web service for a particular task per the quality and context needs of the service request (Elgazzar et al., 2014; Ruta et al., 2015).

Different components of the MWS discovery framework contribute to the challenges in discovering MWS suitable for a particular user needs. Three problems plague the accuracy and effectiveness of the MWS discovery framework. Firstly, coarse-grained MWS categorization and bootstrapping approaches that failed to

deal with the proliferation of MWS, cold-start, reduce the discovery space (Cao et al., 2017, 2013; Jiang et al., 2017). Secondly, context models used in organizing large, heterogeneous context knowledge are constricted by insufficient expressiveness, inadequate extensibility, and slight fragmentation, which confound the difficulty in describing the DME, MWS, and the user's MWS needs (Cabrera et al., 2017; Cheniki et al., 2016). Thirdly, matchmaking requires manual adjustment and disregard context information that triggers self-adaptation, leading to ineffectiveness and inaccuracy in discovering relevant MWS (Win et al., 2019). The machine learning techniques are based on the binary classification that demands an undesirable compromise between partial scores, leading to ineffectiveness and inaccuracy in discovery results (Tian et al., 2019b,a; Xie et al., 2019). Fig. 1 summarizes the concept of MWS discovery in DME.

Recent studies have shown that some components of the MWS discovery framework can be targeted to improve the overall accuracy and effectiveness of the framework. For instance, the most important functional features of MWS (Goals and Tags) can be extracted and categorized to alleviate the MWS proliferation problem (Cao et al., 2017). Similarly, the most crucial context can be identified based on the 5Ws (who, when, what, where, and why) and consolidated into three meta-contexts (user, service, and environment) to help in creating an ontology that organizes context information and support decision making in self-adaptive MWS discovery in DME (Aguilar et al., 2018; Lu et al., 2017). Besides, machine learning can be used to train a model that learns MWS relevance for a given query to effectively alleviate

[☆] Editor: P. Lago.

* Corresponding author.

E-mail addresses: salisu.garba@graduate.utm.my (S. Garba), radziahm@utm.my (R. Mohamad), azizahsaadon@utm.my (N.A. Saadon).

the ineffectiveness and inaccuracy in discovering relevant MWS (Cao et al., 2019; Xie et al., 2019).

The provisioning of MWS is continuous as services can be offered by small providers, including individuals, leading to proliferation and a high influx of similar MWS. This results in a huge search space that affects the effectiveness of the discovery (Zhang et al., 2018). MWS categorization approaches such as (Cao et al., 2017, 2013; Jiang et al., 2017) are proposed to deal with these complications. However, these approaches are affected by a lack of quality feature extraction technique, nevertheless, it outweighs the performance of the MWS categorization algorithm in real-world applications since it portrays the fundamentals of the dataset. The Context Manager is one of the components of the MWS matchmaking that is responsible for monitoring, collecting, organizing, and composing context information for matchmaking and ranking MWS. Context-awareness is a key distinguishable element that provides additional information to the MWS discovery (Bouguettaya et al., 2017). Recently, some context models have been introduced to exploit ontology for improving the MWS discovery framework (Aguilar et al., 2018; Cheniki et al., 2016; Lu et al., 2017). However, the choice of context parameters for MWS discovery is assessed only to a very limited extent, and the use of limited contextual information to make a generalized decision may hinder the MWS discovery in DME.

The discovery of MWS usually takes place in a Dynamic Mobile Environment (DME); an environment that consists of heterogeneous mobile devices operated by numerous classes of users in highly mutable settings (Barakat et al., 2018; Bobek and Nalepa, 2017; Mezni and Sellami, 2016; Verma and Srivastava, 2018). This makes it almost impossible to have an exhaustive model that is applicable in all MWS discovery scenarios. In this environment, the machine learning-based discovery techniques such as neural collaborative filtering (He et al., 2017), deep hybrid collaborative filtering (Xiong et al., 2018), heterogeneous information network (HIN), and Latent Dirichlet Allocation (LDA) (Tian et al., 2019b,a; Xie et al., 2019) are the most promising discovery techniques in the self-adaptive framework. However, the LDA-based approach is less accurate since it can only obtain limited semantically coherent topics in the absence of word embedding and the Relational Topic Model and Factorization Machines (MTR-FM) interaction matrix hinder correct MWS classification.

This paper proposes a self-adaptive mobile web service discovery framework for a DME with improved accuracy and effectiveness in view of the above inspirations. The framework comprises three different components: (1) Mobile web service categorization, (2) Meta-context model, and (3) Self-adaptive matchmaker. The first component consists of Natural Language Processing (NLP) based feature selection and extraction, followed by MWS categorization inspired by a Negative Selection Algorithm (NSA) and K-means to deal with the proliferation of functionally similar MWS. The second component involves the use of a lightweight unified process for ontology building (UPON-Lite) in collaboration with the Feature-Oriented Domain Analysis (FODA) to design an ontology-based context model. The third component consists of Service Relevance Learning (SRL) and the MWS matchmaking inspired by a Modified Negative Selection Algorithm (M-NSA) to deal with the irrelevant discovery of MWS and improve the effectiveness of the self-adaptive MWS discovery framework. Contrary to other frameworks, this proposed framework utilizes explicitly goals and tags for MWS categorization, consideration for the entire information space (users, service, and environment), hybridization of an approximate algorithm, and logic-based matchmaking technique.

The proposed framework was evaluated in terms of both binary relevance and graded relevance using ProgrammableWeb

datasets. The framework was then compared with existing frameworks, and various experiments show better results of the proposed Self-Adaptive Matchmaker (SAMM) framework compared to the baselines. The main contributions of this research to the current state of practice can be summarized as follows:

- An enhanced fine-grained MWS categorization and bootstrapping approach to pave the way for efficient and relevant web service discovery in a dynamic mobile environment
- An enhanced meta context ontology model (MeCOMo) that represents the heterogeneous context information to facilitate self-adaptive service discovery in a dynamic mobile environment
- An improved self-adaptive MWS matchmaking algorithm (SAMM) and an effective self-adaptive service discovery framework for a dynamic mobile environment.
- Series of experiments were carried out using publicly-available datasets to evaluate the accuracy and effectiveness of the proposed framework against state-of-the-art approaches.

The remainder of this paper is arranged as follows: we present the related works in Section 2. Section 3 provides an overview of the proposed framework. The MWS categorization approach, the meta-context model, and the self-adaptive matchmaker, are presented in Sections 4, 5, and 6 respectively. Sections 5 and 6 presents a series of experiments and the conclusion of the paper.

2. Related work

In this section, different studies related to this research can principally be categorized into three directions, namely; MWS categorization/clustering, context model/ontology, and self-adaptive web service discovery framework.

2.1. Goal-based MWS categorization

As a phase that precedes the self-adaptive MWS discovery, MWS clustering/categorization deals with the extracting of critical features from a collection of MWS description and organizing them into clusters so that the MWS in the same cluster have close affinity than MWS in other clusters (Kotekar and Kamath, 2018). For the past years, several approaches have been introduced to investigate MWS clustering/categorization. These include categorization based on functional features, non-functional features, and composite MWS (Cao et al., 2017). Shi et al. (2018) mined the important features of web services described in WSDL or RESTful to generate a keyword vector representing the service's functionalities before clustering the MWS. The functional-based approach proposed by Jiang et al. (2017), in which Natural Language Processing (NLP) is used to acquire the MWS functions, coupled with a semantic expansion of the MWS functions, before clustering using k-means.

The work in Liang et al. (2014) and Wu et al. (2014) co-clustered MWS based on normalized google distance (NGD) and Jaccard Similarity Coefficient (JSC) to kickstart MWS discovery. Tag is a critical attribute of MWS. It offers supplementary qualitative and linguistic insight about each MWS and has been used to enhance the accuracy of service categorization by numerous works (Cao et al., 2017; Chen et al., 2011; Shi et al., 2018; Wu et al., 2014). While these approaches have accomplished many milestones, one of the drawbacks of these approaches is that they neglect the semantic knowledge of words that have proven to be quite significant in the NLP. MWS is now described in natural language. The traditional categorization algorithms used in these approaches such as k-means (Shi et al., 2018), agglomerative

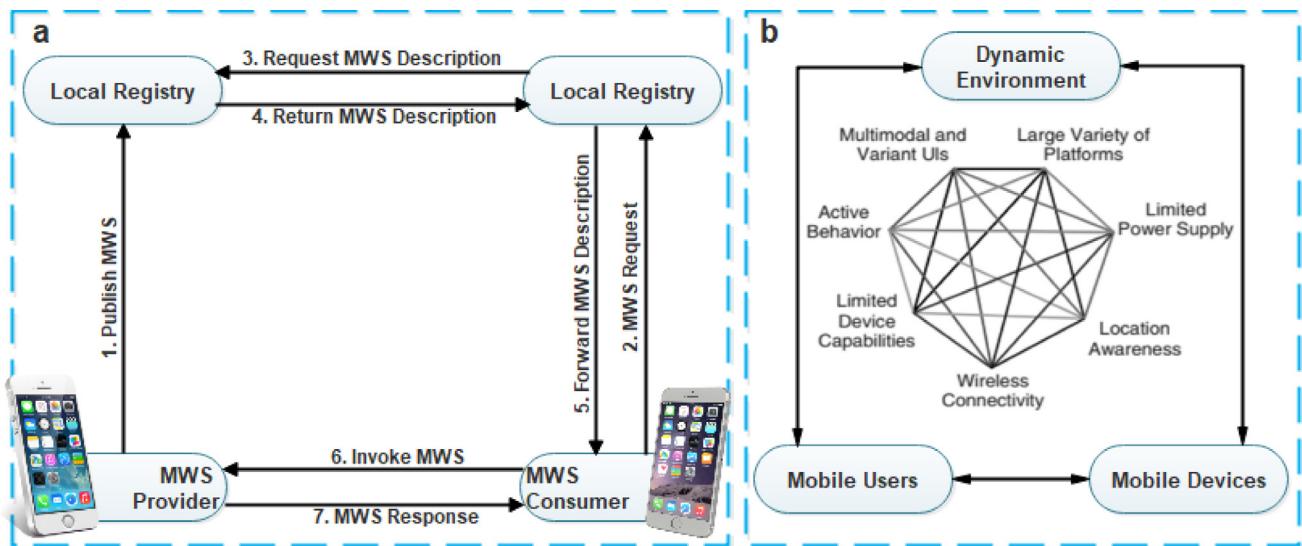


Fig. 1. (a) The concept of MWS discovery (b)The concept of DME.

(Rupasingha et al., 2017), density-based spatial clustering of applications with noise (DBSCAN) (Chen et al., 2017) left more to be desired in terms of speed, simplicity, and accuracy.

With the recent achievements of meta-heuristics clustering algorithms, several attempts have been made to hybridize traditional and meta-heuristics clustering algorithms (Cao et al., 2016; Garba et al., 2020a; Kotekar and Kamath, 2018, 2016). For example, Kotekar and Kamath (2018) hybridized Cat Swarm Optimization (CSO) and k-means to determine the categories and placed the services in the appropriate cluster. Moreover, other approaches (Cao et al., 2017, 2016) typically consider the LDA model in computing the degree of similarity between the MWS based on the topic and agglomerative nesting for hierarchical clustering (Agnes) to categorize and positioned each MWS in the suitable cluster. One major drawback of these hybridized approaches is that they are linear models as such they require relatively larger datasets for effective training, while the meta-heuristics clustering algorithms are sometimes volatile in terms of categorization outcomes.

Contrary to the above approaches, the proposed MWS categorization approach in this paper introduces a qualitative feature extraction technique based on NLP for capturing both goals tags and specifically utilize NSA for MWS categorization to improve the accuracy of the categorization.

2.2. Context model for MWS discovery

The Context Manager is one of the components of the MWS matchmaking responsible for organizing and composing context information for matchmaking and ranking of MWS. A considerable number of literature has been published on the importance of context and its impact on the accuracy of MWS discovery provided that it is properly modeled (Cheniki et al., 2016). Most of the early context models for MWS discovery specify non-functional properties (NFPs) using attribute-value specifications (Saadon and Mohamad, 2011). In this case, the context parameters of the MWS are identified, and values are assigned, e.g. cost = free. However, predefined contexts allocated with arbitrary values are not compatible when context values are gauge dynamically. Moreover, the segregation of NFPs such as QoS measurement from other context entities can constrain the expressiveness of the entire model. Similarly, in Pahlevan et al. (2015) skyline and aggregate were adopted to model equally important context attributes. This was achieved by ensuring that

one context attributes do not dominate other context attributes as skyline is context-agonistic and strive for balance unlike Pareto used in García et al. (2010) and aggregation used in Chouiref et al. (2016) that reduce the attributes to a single score, which consequently, leads to less accuracy in multi-criteria decision-making such as self-adaptive service discovery.

Recently, ontology-based context models are used to support a formal representation of various context knowledge and enable the sharing of information between contexts through logical reasoning, which enhances service discovery. In the development of ontology-based context models, CONON (CONtext ONtology), CoDAMoS, and Standard Ontology for Ubiquitous and Pervasive Applications (SOUPA) are the most widely adopted generic ontologies (Cabrera et al., 2017). This is due to the modularity outlook presented by these ontologies in which several entities of ubiquitous applications such as user, location, etc., form the high-level ontology that can be extended to domain-specific concepts hierarchically. In Cheniki et al. (2016), WSMO-Context, MobiSO, WSMO-QoS, and WSMO-M (Mobile) were also discussed. These are extensions of the two notable languages used to describe web service semantically; OWL-S (Web Ontology Language for Web Services) and WSMO/WSMO-Lite (Web Service Modeling Ontology) that are proposed to support contextual web service discovery despite processing complexity. These models exemplified a context model that considers service alone while giving insufficient consideration to other critical contextual entities such as user, devices, location, etc. The use of limited contextual information to make a generalized decision may hinder the MWS discovery in DME (Lu et al., 2017).

Different from these studies, this paper extends previous context models inspired by MobiSo (Cheniki et al., 2016), MCOnT (Lu et al., 2017), CAMeOnto (Aguilar et al., 2018) and introduced meta-context ontology to organize and represent the heterogeneous context information which is critical in differentiating relevant and irrelevant MWS, thus improving the accuracy and effectiveness of MWS discovery framework.

2.3. Self-adaptive web service discovery

Web Service discovery is the act of matchmaking between the advertised service and the service request to find the most relevant web service for a particular task. Previously, Syntactic (Cheng et al., 2018), Semantics-aware (Saadon and Mohamad, 2015), logic-based (Stavropoulos et al., 2016) are the dominant discovery

techniques in the literature. For example, web service operations discovery (OpD) proposed in [Cheng et al. \(2018\)](#) mine the service interface to create index libraries and use co-occurrence probability to discover services with solitary or multiple operations. The TOMACO proposed in [Stavropoulos et al. \(2016\)](#) integrates formal reasoning with syntactical technique to carry out the matching in an open-ended, dynamic registry. Higher priority is given to the logic-based matching while the syntactic matching compensates for the absence of exact matching in the logic-based.

The introduction of self-adaptive service discovery by [Ke and Huang \(2012\)](#) and [Klusch et al. \(2016\)](#) is fitted with a classification model that switches from one pattern to another to identify viable alternatives whenever a MWS is estimated to be irrelevant according to its functional specifications, and/or the MWS does not match the context needed. This paved the way for the recent success of self-adaptive service discovery. For instance, [Win et al. \(2019\)](#) propose Self-adaptive QoS-aware web service discovery (SQoSD) in which concept, attribute, and constraint similarity between service request and offered services are matched. The self-adaptive matchmaker iteratively restructures requirement ontology trees to match request and offer until the similarity is above a certain threshold, or the user requirement is satisfied. However, the continuous iteration can lead to an infinite loop because only one level of matchmaking is considered to satisfy the request and the cost-effectiveness of such discovery cannot be guaranteed especially in DME.

In a new development, neural collaborative filtering ([He et al., 2017](#); [Xiong et al., 2018](#)), Relational Topic Model (RTM), and Factorization Machines (FMs) ([Cao et al., 2019](#)) constitute the contemporary self-adaptive MWS discovery techniques. Essentially, collaborative filtering (CF), either a memory-based or model-based identify MWS based on the correlation between the user request and the MWS description ([He et al., 2017](#)). The model-based CF is more effective and popular in the MWS discovery domain as it uses data mining and machine learning techniques in the context of predictive models to find complex patterns in MWS description ([Xiong et al., 2018](#)). However, these MWS discovery techniques generally experience some significant problems, such as data sparsity and cold-start problems.

With the recent success of machine learning in Natural Language Processing (NLP), several machine learning approaches for self-adaptive MWS discovery have been proposed ([Zhang et al., 2018](#)). Most of these approaches are based on the Probabilistic Latent Semantic Analysis (pLSA), Latent Dirichlet Allocation (LDA) models ([Tian et al., 2019b,a](#); [Xie et al., 2019](#)). Although LDA and pLSA based approaches are good in identifying the relationship between topics of MWS, one major issue of the LDA and pLSA based approaches is that they typically rely on only pre-trained word embedding such as Word2vec to capture both syntactic and semantic information of words which makes them less accurate as word embeddings alone cannot efficiently capture all syntactic information of some MWS terms. Moreover, current approaches are often constrained by a design-time adaptation technique that involves reconstruction instead of reinforced learning and additional training of model after feedback acquisition over a long period which is a very costly process ([Klusch and Kapahnke, 2012](#)).

Despite the above contributions, the self-adaptive MWS discovery framework has not realized its full potential since it has fallen short in resolving challenges that include a high influx of similar MWS, unpredictable changes in DME, the discovery of irrelevant MWS which hinder its accuracy and effectiveness. Contrary to the related approaches, the proposed framework in this paper introduces a Modified Negative Selection algorithm (M-NSA) based model and a matchmaker to improve the accuracy and effectiveness of the self-adaptive MWS discovery framework, given that it is adopted to solve numerous web service discovery and composition problems ([Zhao et al., 2019, 2014](#)).

3. Framework overview

To improve the effectiveness and efficiency of the self-adaptive MWS discovery framework for a dynamic mobile environment, several components of the MWS discovery framework are targeted for improvement. Fig. 2 illustrates the overall self-adaptive MWS discovery framework, which comprises three distinct parts, specifically, MWS provisioning, MWS request, and MWS matchmaking, where each part is composed of a set of disparate components. The role of each component is discussed in this section; however, special attention is accorded to the components (MWS categorization component in Section 4, context component in Section 5, matchmaking component in Section 6) that are improved in this paper.

In the MWS categorization component, a web crawler tool is employed to obtain MWS. The most important functional features (Goals and Tags) are extracted, which is used for MWS organization to reduce the search space. Consequently, k-means is subsuming in the modified negative selection algorithm (M-NSA) to put the MWS in a suitable category. In the context component, the 5Ws (who, when, what, where, and why) are used to identify the context for MWS discovery in DME as it is crucial to identify who conveys the MWS request and why, what MWS was requested, where, and when was the request made. To this end, a diverse number of classes are consolidated into three meta-contexts (user, service, and environment), the Feature-Oriented Domain Analysis (FODA) and the unified process for ontology building (UPON Lite) is adopted to create an ontology that organizes context information and support decision making in self-adaptive MWS discovery in DME. In the matchmaking component, the two stages of the M-NSA are transformed into service relevance learning and self-adaptive matchmaking (SAMM) to improve the MWS discovery so that more relevant MWS are returned for a given request by leveraging the reduced search space and the context model.

Mobile Web Service Request: The MWS Request describes the user's needs to obtain the most relevant MWS for a particular task. The MWS request is typically submitted in plain text rather than formal description language (e.g., WSDL) to describe the user needs due to the constraints of mobile devices. The MWS request is passed to the request handler together with contextual information for further analysis before the matchmaking.

Request Handler: The Request Handler is one of the components of the self-adaptive MWS discovery framework that carry out advanced analytical operations on the users' MWS request. It is responsible for parsing user requests to obtain keywords, acquire fresh context information from a user request, format the MWS request to distribute each extracted parameter to the suitable component in the matchmaking process.

Context Manager: The Context Manager is one of the components of the self-adaptive MWS discovery framework that is responsible for monitoring, collecting, organizing, and composing context information for matchmaking and ranking MWS. In a user-centric MWS discovery, context information in DME consists of the user context (e.g. user profile, activities, devices, and preferences), the service context (e.g. services' profile, capabilities, QoS), and the environment context (e.g. users' location, environmental condition, and the time), which are all fundamental in the discovery of the most relevant MWS in DME. Details about the proposed context model are provided in Section 5.

Mobile Web Service Categorization: The Mobile Web Service Categorization is one of the components of the self-adaptive MWS discovery framework responsible for placing functionally similar MWS into the same category as functionality is the primary

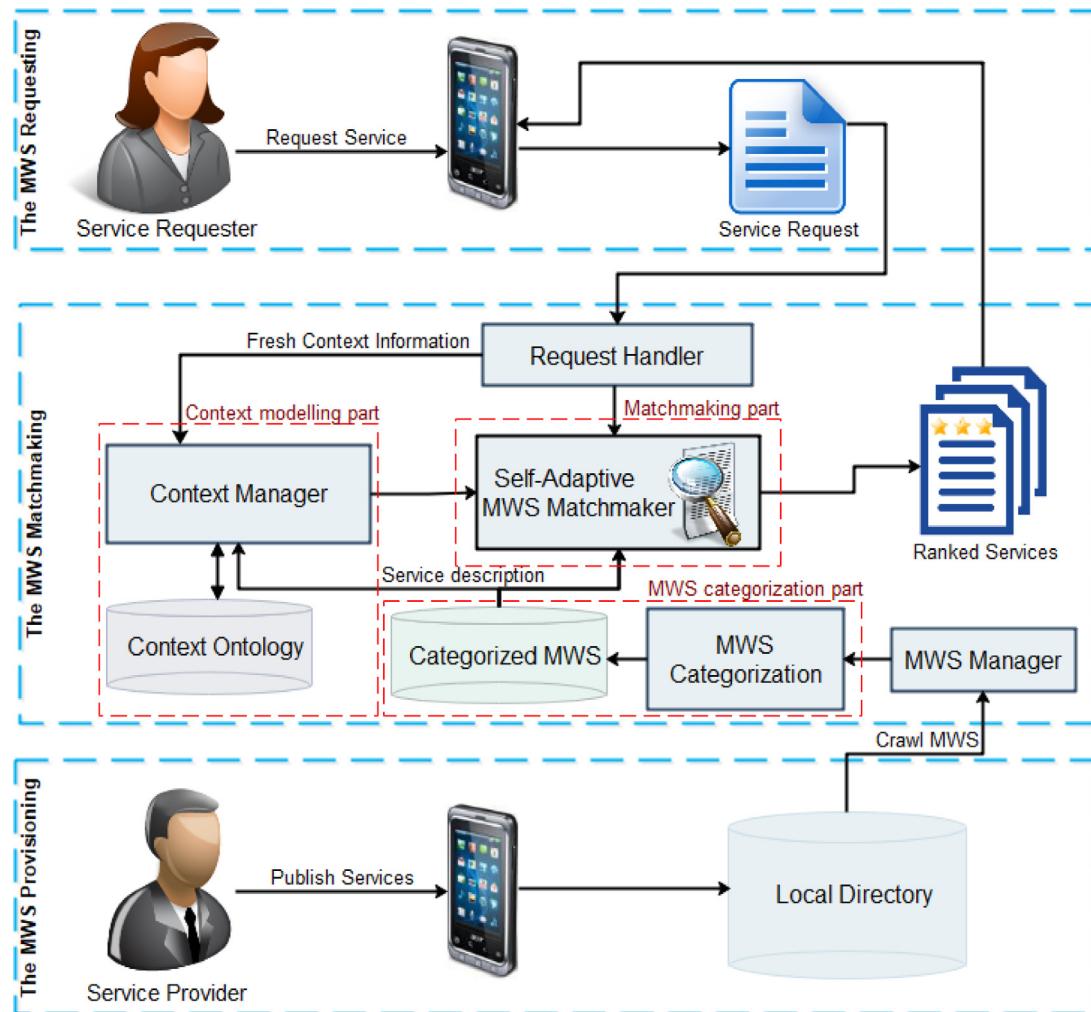


Fig. 2. An overview of the self-adaptive MWS discovery framework.

target of every MWS request. This ensures that matching is only performed on the MWS in the target category instead of the entire MWS to reduce the search space and improve the efficiency and overall accuracy of self-adaptive web service discovery. Details about the proposed MWS categorization approach are provided in Section 4.

Self-adaptive Matchmaking: Matchmaking is the crucial component of the self-adaptive MWS discovery framework responsible for determining the degree of match between the advertised MWS and the requested MWS to find the most relevant web service for a particular task. In the self-adaptive MWS discovery, the matchmaker initially learns the MWS relevance over a specified training set of positive and negative samples. This is followed by matchmaking, where the request is matched against the offer to obtain a ranked result based on the learned relevance. Details about the proposed self-adaptive MWS matchmaker are provided in Section 6.

MWS Manager: The MWS Manager is one of the components of the self-adaptive MWS discovery framework that performs advanced analytical operations on the offered MWS. It is responsible for extracting useful phrases or keywords that describe the MWS. The functional and non-functional parameters provided in the MWS offer are used for categorization and subsequently

matchmaking to identify the most relevant MWS based on the user request.

MWS provisioning: The MWS provisioning part of the self-adaptive MWS discovery framework is responsible for offering and advertising the availability of MWS in a local service directory by individual MWS providers to ensure total control of the MWS offer. The MWS is normally described with textual descriptions rather than formal description language (e.g., OWL-S, WSDL, etc.). The MWS provided is passed to the MWS manager for further analysis before the categorization and matchmaking.

The overall flow of the entire approach is shown in Fig. 3. The primary input of the approach is MWS Data, algorithm 1 uses the goals and tags from the MWS data to formulate the categories. The output of algorithm 1 is the initial detector set, which is used in algorithm 2 to categorize the MWS, the categories of functionally similar MWS represent the output of the algorithm. Algorithm 3 takes the training set as input; the training set is part of the MWS Data that consists of the MWS description, a set of offered MWS, and MWS requests. The context ontology is used to describe offered MWS and MWS requests context used in algorithm 3. The output of algorithm 3 is a set of detectors which serve as one of the inputs of algorithm 4. The top number of MWS that satisfy a given request in the output of algorithm 4.

After the results are obtained, the next step is result interpretation and optimization which involves making sense of the

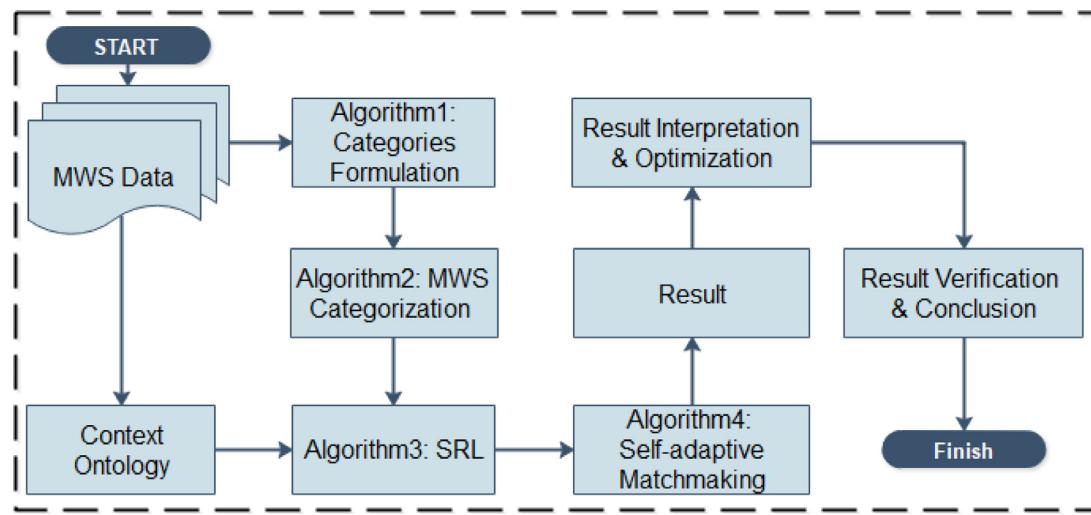


Fig. 3. The overall flow of the entire approach.

results and seeing whether the findings confirm, improve, contradict, or offer novel insights in comparison with the related studies. The final step is result verification and conclusion, where the results are checked to ensure that the framework fulfills its intended purpose which is the discovery of relevant MWS followed by a concluding remark on the research.

4. Mobile web service categorization

In this section, the MWS categorization approach is discussed. As illustrated in Fig. 4, the proposed MWS categorization approach comprises three stages (feature selection and extraction, formulating the categories, and goal-based MWS categorization). This approach was inspired by a negative selection algorithm (NSA) and K-means to deal with the proliferation of functionally similar MWS by minimizing the search space, minimizing the computation time, reducing the workload of the matchmaking algorithm for more accurate MWS discovery. The three stages are explained in Sections 4.1, and 4.2.

4.1. Feature extraction and categories formulation

The description of MWS is provided in the form of natural language to avoid the complex standards (WS*) that are unsuited for DME (Fielding, 2000). Given that the functionality (Goals) of the MWS is stable compared to non-functional properties and is always the primary target of a request (Sellami et al., 2013). Goals are considered the most important features of MWS. The Tags provide additional information, especially the categorical meta-data of the MWS. Therefore, the description of the MWS is assessed to mine the goals and tags. Algorithm 1 provided in Fig. 5 guides the feature selection and extraction process.

Steps 4 through 8 describe how the Goals are extracted from the MWS description, combine with Tags parameters, and converted into tokens. In the natural language processing toolkit (NLTK), tokenization is used to break down the textual description of MWS into smaller units termed tokens. This is followed by removing stop words from the MWS corpus, after which NLTK lemmatization is applied. Lemmatization is used to analyze the MWS goals and tags morphologically to remove inflectional endings after considering the context. This is because lemmatization is more semantic-aware than the NLTK stemmer to ensure the efficient creation of the MWS vector. As shown in steps 6 through 7, the TF-IDF weight is computed. Given the large dimension created, a multidimensional scaling (MDS) is used to place each

MWS data point in 2-dimensional space to preserve the distance between MWS and tackle outliers. In step 8, the Elbow method is used to identify the Elbow Point, which corresponds to the appropriate number of categories.

The profile of the categories is formulated to enable discrimination between what belongs to a particular category and what does not. This process is called detectors generation. In NSA, the detector (artificial lymphocytes) are generated to recognize non-self. Variable-sized detectors are used in this case because there is no need to set the number of detectors in advance, the large area of non-self space can be covered by a limited number of variable-sized detectors; likewise, smaller detectors can cover the gaps, thus reducing the categorization error, unlike constant-size detectors. Steps 10 through 19 of algorithm 1 describe how the profile of the categories is formulated.

The centers C in each category K are chosen at random to initialize the process. Since the description of MWS is provided in the form of natural language, cosine similarity is used to assign MWS to the Category based on its proximity to center C as shown in step 12. The distance between two MWS is shown in Step 12 in which S_1 and S_2 are two MWS vectors under consideration; S_{1i} and S_{2i} are components of vector S_1 and S_2 , respectively; and n is the number of components available. Thus, the result ranges between -1 and 1 , where -1 is perfectly dissimilar, and 1 is perfectly similar. As shown in Step 14, the centers C of each category Z are recomputed after every new MWS until no change is observed. The k-means is extended to perform a comparison between a new MWS_{xi} and a threshold value τ (the boundary and outside the category), if the MWS is less than the threshold value, then the data will be discarded as it does not satisfy the condition of being a detector; otherwise, it will be added into the initial set of detectors (DS) as shown in Steps 16 through 19.

4.2. Goal-based MWS categorization

The proposed MWS categorization algorithm for self-adaptive MWS discovery in DME is designed to deal with the proliferation of functionally similar MWS by minimizing the search space, minimizing the computation time, reducing the workload of the matchmaking algorithm, and paving the way for efficient and more accurate MWS discovery. As mentioned in the previous section, the entire MWS categorization is based on hybridizing k-means and M-NSA; therefore, the categorization consists of two phases (censoring and monitoring). Algorithm 2 provided in Fig. 6 guides the MWS categorization process.

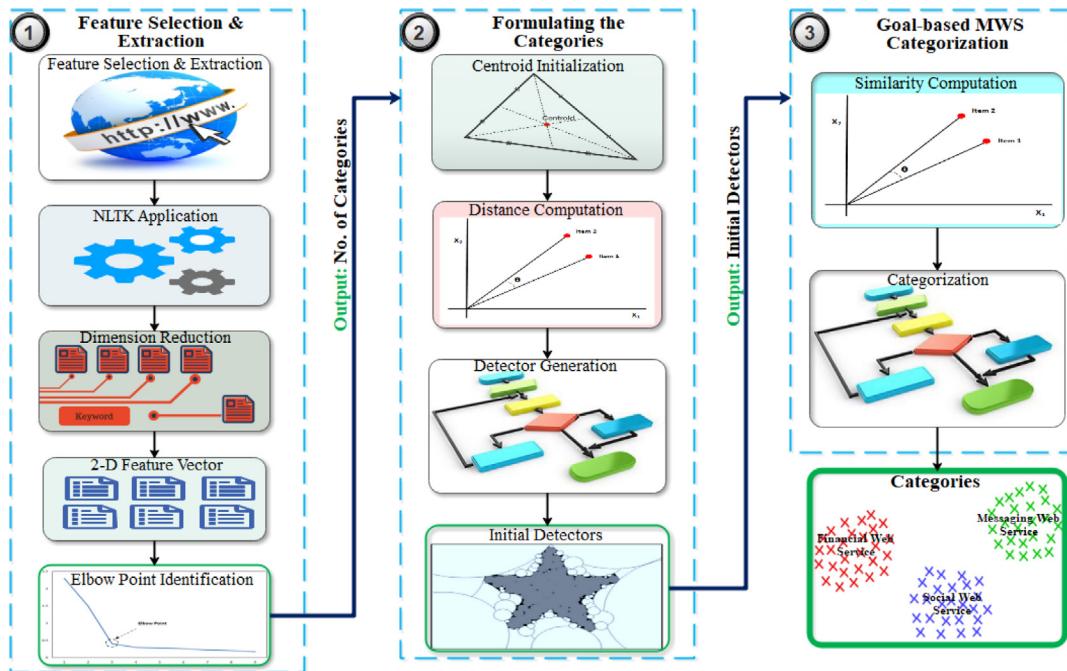


Fig. 4. The proposed MWS categorization approach.

Algorithm 1: Categories formulation algorithm

```

1. Input: Crawled MWS Data
Output: Initial Detector Set DS'
2. //Compute the features of the training sample
3. TS'  $\leftarrow \emptyset, K \geq 1$ 
4. for all Data Points in TS'
    5.     Pre-process using NLTK
    6.     Compute TF-IDF using  $tf - idf_{x,S,TS} = tf_{x,S} \times idf_{x,TS}$ 
    7.     Reduce dimension using  $M = [S_{ij}]$ , and  $S_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ 
    8.     Compute the optimal number of K using  $W_k = \sum_{r=1}^k \frac{1}{n_r} D_r$ 
9. end for
10. Initialize K random Centroids  $C, 1 \leq i \leq k$ 
11. for all instances S in TS do
    12.     Compute the nearest Center using  $Sim(S_1, S_2) = \frac{\sum_{i=1}^n S_{1i} \times S_{2i}}{\sqrt{\sum_{i=1}^n S_{1i}^2} \times \sqrt{\sum_{i=1}^n S_{2i}^2}}$ 
    13.     Assign S to K
    14.     Recompute Centroids C
    15.     if Centroids C changes then Repeat 12 and 13
    16.     else Check the Distance between S and τ in K
    17.     if  $S < \tau$  then Remove S
    18.     else Add S to DS
    19. end for
20. end

```

Fig. 5. Feature extraction and category formulation algorithm.

Algorithm 2: MWS Categorization algorithm

1. **Input:** MWS to be classified and Input Variable $X = (x_1, x_2, \dots, x_n)$ where $\forall x \in S$
2. Initial set of detectors, N_{\max} and the threshold τ
3. **Output:** Set of detectors $D = (d_1, d_2, \dots, d_n)$ that satisfy the categorization condition
4. Categories of functionally similar WS $Z = (z_1, z_2, \dots, z_n)$
5. **repeat**
6. Start Phase1: Censoring phase
7. Use the x_i generated by Algorithm 1 as initial candidate detectors d ;
8. Select web service data x , where $x \in S$;
9. Calculate the similarity of x with every detector d in D , $\forall d \in D$
using $Dist(x, d) = \sqrt{\sum_{i=1}^n (x_i - d_i)^2}$
10. Start Phase2: Monitoring phase
11. Check the Distance $f_{aff}(d, x)$;
12. **if** $f_{aff}(d, x) < \tau$ **then** Add the new x to the Category Z ;
13. **else** Add x to D ;
14. **end if**
15. **until** $|D| = N_{\max}$ or D cover all MWS

Fig. 6. MWS categorization algorithm.

Steps 1 through 4 show the input and the output of the MWS categorization algorithm, the output of algorithm 1 (initial detectors) serves as the input to algorithm 2; other inputs are the MWS to be categorized, the maximum number of detectors N_{\max} , and the threshold value τ set at 0.5. The output D represents the set of detectors that satisfy the categorization condition, and Z represents the categories of functionally similar MWS.

Steps 6 through 9 show the censoring process, which involves the random generation of detector candidates d to add and update the initial detectors in the entire feature space (S). The detectors are created based on how many MWS that needs to be categorized. During this process, MWS vectors mapped into the 2-dimensional hyperspace ($X = 16$) are compared with the detector candidates to evaluate the similarity values. In Steps 10 through 15, the monitoring phase begins by comparing the affinity value obtained in the censoring phase $f_{aff}(d, x)$ with a threshold value τ . If the distance is less than the threshold value, then the new MWS is added to the category; otherwise, it is added to the set of detectors (D) for that category, as shown in Fig. 6. The Censoring phase and Monitoring phase are iterated until all MWS are categorized.

For the proposed approach to self-adapt, the profile of each category is modified by adaptively fine-tuning the self radius as a result of the addition or withdrawal of MWS into or from the category. The detection radius also changes as the numbers of detectors increase to cover partial self space, thereby decreasing the categorization error.

5. Meta-context model

In this section, the Meta-context model is discussed. The goal of this context modeling is to resolve the problem of insufficient expressiveness and insufficient extensibility to enable new context elements that are constantly changing, hard to quantify, and

influenced by environmental factors during the MWS discovery in DME. This was inspired by MobiSo (Cheniki et al., 2016), MCOnT (Lu et al., 2017), and CAMeOnto (Aguilar et al., 2018), moreover, the unified process for ontology building (UPON Lite) methodology (De Nicola and Missikoff, 2016) was used to create an ontology that organizes context information and support self-adaptive MWS discovery in DME.

5.1. Model

The importance of context and its impact on the accuracy of MWS discovery, provided that it is properly modeled had been established (Cheniki et al., 2016). Given the different drawbacks of contextual information exhibited by the models discussed in Section 2.2, an extension that fits the contextual needs of self-adaptive MWS discovery is developed. In the meta-context ontology model (MeCOMo) illustrated in Fig. 7, the user, service, and environment are differentiated by defining three adequate classes based on the 5Ws context interaction schema (Aguilar et al., 2018; Lu et al., 2017). This is because a self-adaptive MWS discovery framework must identify who conveys the request and why (user), what was requested (service), where, and when the request was made (environment).

User meta-context defines who and why; it consists of information about the users' profile, activities, devices, and preferences. The "UActivity" represents the action of the user (e.g., running, walking, cycling, driving, etc.). Services meta-context defines what; it consists of information about the services' profile, capabilities, QoS, etc. Environment meta-context defines where and when; it consists of information about the users' environmental location, users' environmental condition, and the time. The "Condition" represents the prevailing state of the weather in the environment (e.g., temperature, rain, dust, etc.). The meta-context captures the general concepts that are common and

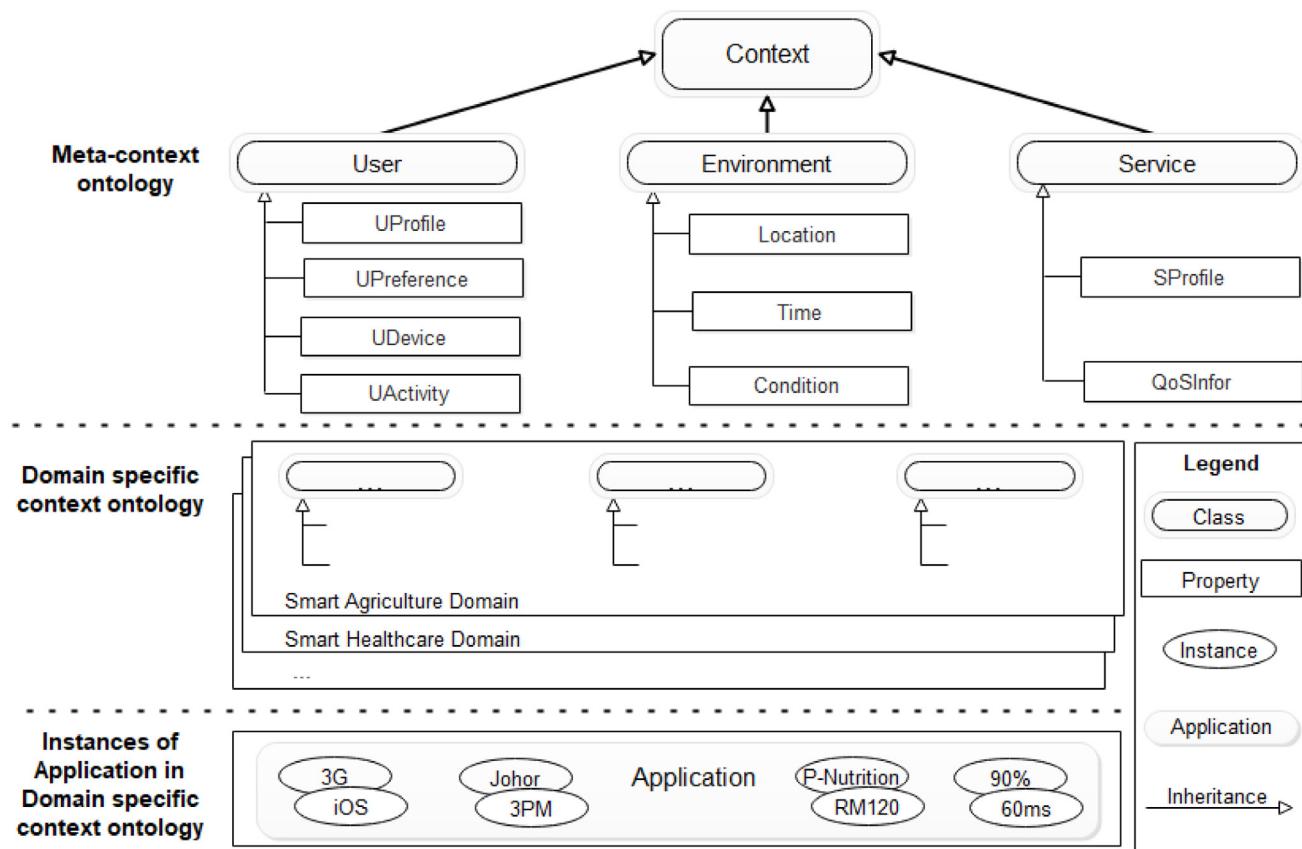


Fig. 7. Hierarchical representation of MeCOMo.

applicable regardless of the domain to promote extensibility and provide the needed expressiveness. The domain-specific contexts capture the concepts that apply to a particular domain. The instances of application-specific contexts capture the properties of concepts in a particular application at a particular domain.

5.2. Ontology

Ontology supports the formal representation of various context knowledge and enables sharing information through logical reasoning, detecting inconsistencies, and simplifying functional complexities (De Nicola and Missikoff, 2016). This is due to the modularity outlook presented by these ontologies in which several entities of ubiquitous applications such as user, location, etc., form the high-level ontology that can be extended to domain-specific concepts hierarchically. Though, the effort by CoDAMoS, MobiSO, mIO! (Poveda-Villalón et al., 2010) provided an exhaustive number of entities that complicate the description of context especially in DME. The MeCOMo ontology is scoped to cover the knowledge sharing between users, environments, and services to better represent the heterogeneous context in DME and promote expandability depending on domain and application.

In the UPON-Lite, the methodology of ontology construction begins with domain analysis, where fundamental concepts are identified. This is followed by predication, taxonomy, and part-hood. During the construction process, an appropriate ontological language is used to formalize ontology, which can support the adjustment and extension of the ontology-based on the domain of discourse. The fundamental concepts of MeCOMo are identified from the MobiSo, MCOnT, and CAMEOnto based on the 5Ws (who, when, what, where, and why). To extract the domain terminologies regarding self-adaptive MWS discovery, two domains that

need MWS are considered. Smart agriculture and healthcare are regarded as the most important and arguably the biggest service domains that affect the world population. These two domains affect the majority of the united nations' sustainable development goals (SDGs) directly or indirectly (Taylor, 2018). The Feature-Oriented Domain Analysis (FODA) is adapted (Fantechi et al., 2019; Hamdan and Jawawi, 2012; Zaki et al., 2014) to capture concepts that represent context properties by identifying the commonality and variability in smart agriculture and the smart healthcare domain.

Taxonomy organizes concepts into a hierarchy of fundamental classes and sub-classes. This enables better comprehension by humans, integration, and reuse with other ontologies. A taxonomy represents the backbone of an ontology but is difficult to define. Fig. 8 presents part of the taxonomy of MeCOMo, which shows the grouping of the main classes with their sub-classes.

Predication and parthood stage focus on the properties and attributes that define the relationship between relevant context entities in the MWS Discovery domain. The relation can be functional, inverse functional, transitive, symmetric, asymmetric, reflexive, irreflexive. e.g., isMetricOf relation is the inverse of hasMetric relation. Object properties are used to relate concepts, while data properties relate concepts with data values. e.g., the “QoSInfo” class has the object property “hasCharacteristics” which is linked to Cost, Rating, etc. The “User” class has an object property “hasActivity” which is linked to the “Role” class. Fig. 9 shows an overview of the classes, object properties, data properties, and individuals in Protégé (Musen, 2015).

To instantiate and demonstrate the usability of the MeCOMo for self-adaptive MWS discovery framework, Fig. 10 illustrates the MeCOMo with a description of CareMessage with context information in Protégé. One of the properties of the CareMessage MWS is the context-based MWS description which is adopted

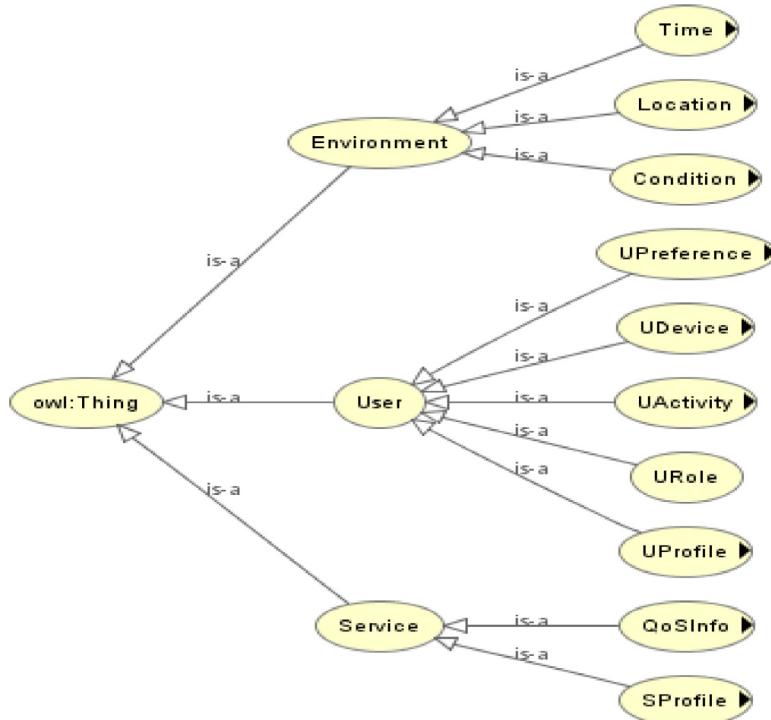


Fig. 8. The taxonomy of the MeCOMo.

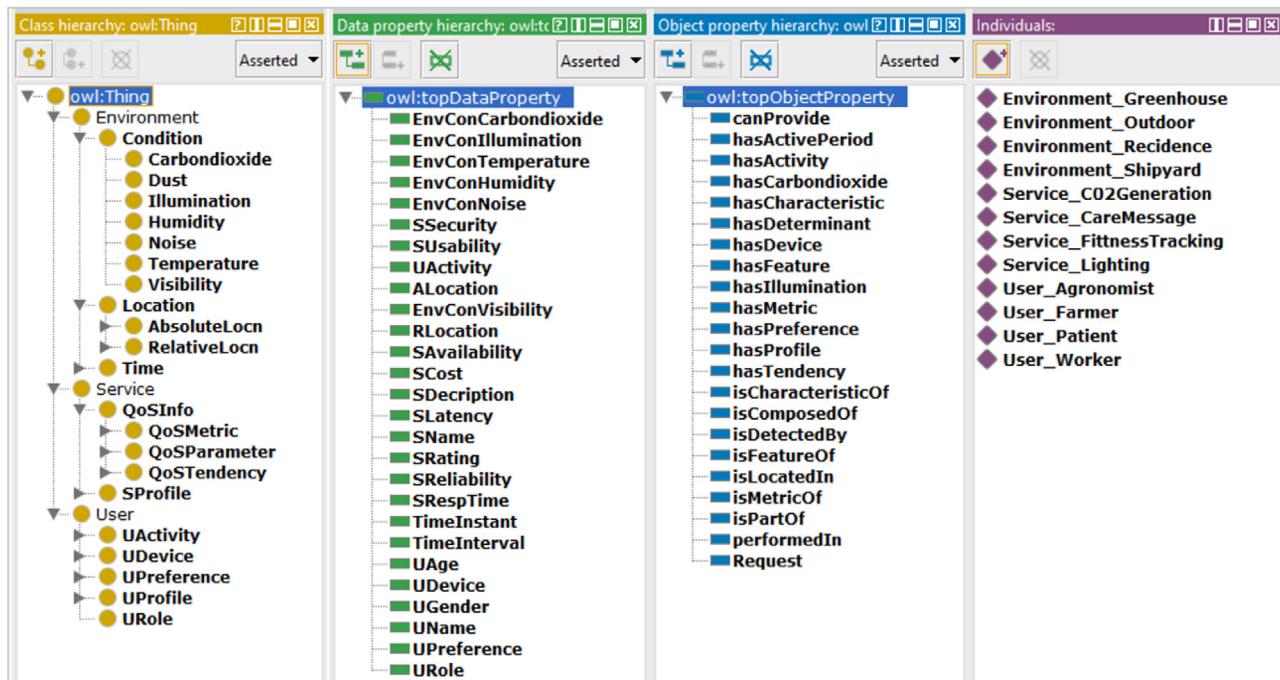


Fig. 9. The relationship between relevant context entities in MeCOMo.

as it allows the development of a self-adaptive MWS discovery algorithm where MWS provisioning is not limited to technical providers only. The CareMessage MWS discloses context information such as SMS mode of communication in the description (line 1543), the quality-of-service information such as security (line 1545), and its usability (line 1546). The information specified during the provisioning and requesting is also used in the discovery of a service.

6. Self-adaptive matchmaker

In this section, the self-adaptive matchmaker (SAMM) is discussed. As illustrated in Fig. 11, the proposed self-adaptive matchmaker (SAMM) comprises two stages (service relevance learning and MWS matchmaking). This matchmaker is inspired by a modified negative selection algorithm (M-NSA) to deal with the irrelevant discovery of MWS and improve the effectiveness of

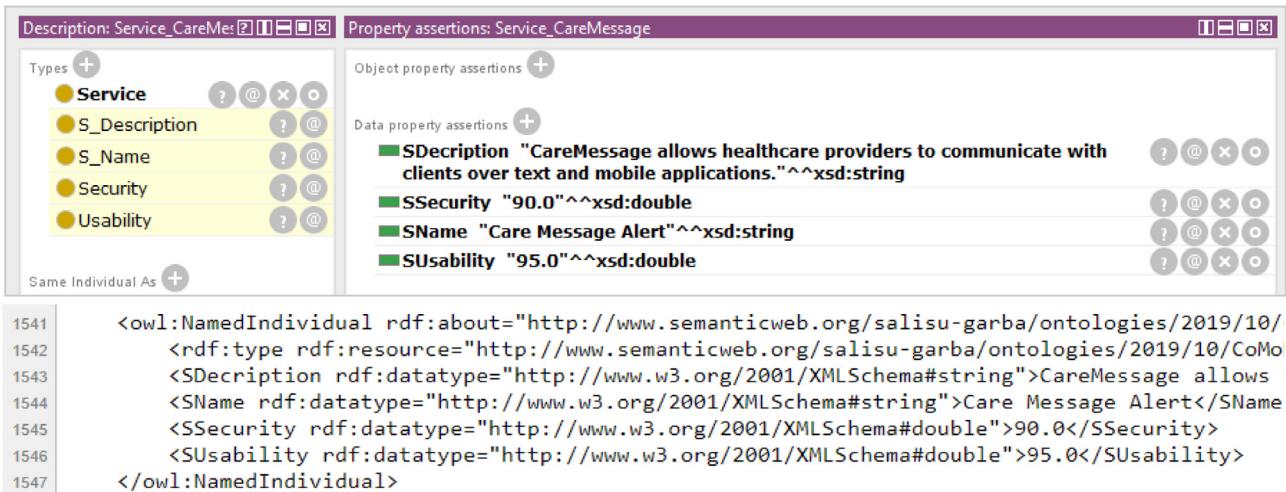


Fig. 10. The CareMessage MWS described with MeCOMo in Protégé.

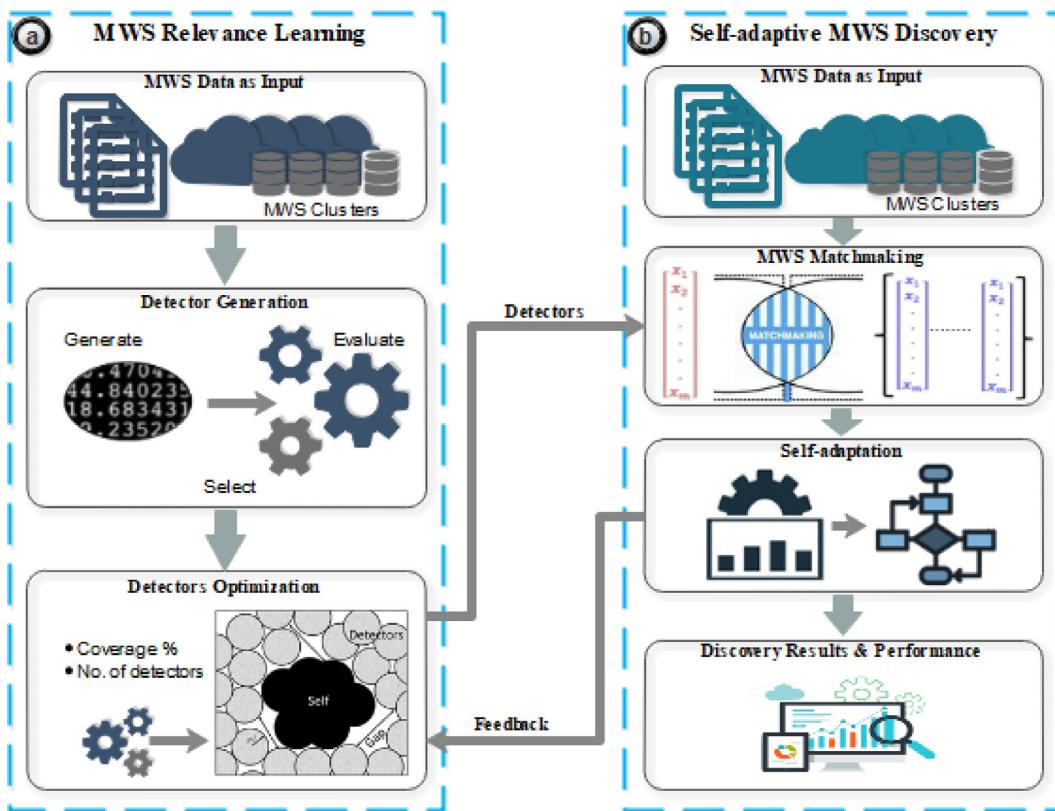


Fig. 11. Overview of the proposed self-adaptive approach.

the self-adaptive MWS discovery framework. The two stages are explained in Sections 6.1 and 6.2, respectively.

6.1. Service relevance learning algorithm

Service Relevance Learning (SRL) is a process that involves matching queries and MWS descriptions to determine the degree of relevance (numerical score) of each MWS to a given query. A learning algorithm subsequently uses this to produce a model that computes the relevance of MWS for actual queries. The SRL starts by assigning values to some control parameters. The control parameters are initialized parameters that affect the behavior of the MNS-A. The three control parameters, in this case, are

self radius r_s , the maximum number of detectors D_{max} , and the estimated coverage C_0 .

The self radius r_s , is the length of the line from the self space center to any point on its edge. The maximum number of detectors D_{max} is the number of detectors (constant-size or variable-size) that are randomly generated to cover the entire self space which contains all the MWS. The estimated coverage E_c is used to determine the level of coverage C_0 based on the percentage of covered and uncovered regions. The logical steps of the SRL algorithm are shown in Fig. 12. Step 6 explains the two stopping criteria for service relevance learning. Since the fundamental goal is to accomplish the generation of the detector

Algorithm 3: Service relevance learning algorithm

```

1. Input: Training Set TS'
2.           $r_s$ : Self-radius
3.           $E_c$ : Estimated coverage
4.           $D_{max}$ : Maximum number of detectors
5. Output: Initial Detector Set DS'
6. while ( $C < E_c$ ) || ( $D < D_{max}$ )
7.     Generate a random Detector candidate  $d$  for  $C_1, C_2, C_n$ 
8.     Calculate minimum distance to any self points using:
9.      $Dist(s, d) = \sqrt{\sum_{i=1}^n (s_i - d_i)^2}$ 
10.    if ( $dist\_min < r_s$ )
11.        Return to top
12.    else
13.        if ( $D = 1$ )
14.            Store detector as  $d_i$  and  $dist\_min = r_{di}$ 
15.            Increment i + 1
16.        else
17.            Calculate minimum distance for each previous detector  $dist\_min2$ 
18.            if ( $dist\_min2 < r_d$ )
19.                 $C = C + 1$ ,
20.            else
21.                Store detector as  $d_i$  and  $dist\_min2 = r_{di}$ ,
22.                Increment C + 1
23.                 $C = 0$ ,
24.            endif
25.        endif
26.    end while

```

Fig. 12. Service relevance learning (initial detector generation).

at the exact or nearby desired coverage, the most suitable condition is, therefore, to infer based on the estimated coverage. The substitute option is to limit the maximum number of detectors to deal with circumstances where the predicted coverage is not feasible.

Given that it is important to create, train and optimize sets of detectors that cover the entire self-space, step 7 generates candidate detectors per the number of categories of the MWS out of which optimized detectors are obtained to be used at the matchmaking stage. In step 8, the Euclidean distance is used to calculate the minimum distance $dist_min$ between self-points and the detectors. The $dist_min$ is compared with the self-radius and if the $dist_min$ is less than the self radius r_s , then that detector is discarded otherwise the detector is added. The detectors are counted before the introduction of the new detector to ensure that the required number is not exceeded.

In step 16, the Euclidean distance is used to calculate the minimum distance $dist_min2$ between the previous detector and the detector-radius r_d . The minimum distance is compared with the detector-radius r_d and if the $dist_min2$ is less than the detector-radius r_d , then the coverage is increased by 1. Given that too many detectors would result in redundant detectors, and too few detectors would result in poor detection rates, the choice to introduce additional detectors or otherwise is difficult as the coverage is constantly evolving. As such, a constant data set is used to estimate coverage and to assess the required number of detectors, which mostly maintains the overhead costs of service relevance learning under check.

Table 1
The five degree of match.

Degree of Match	Logical Representation	Weighted Representation
Exact Match	(RWS = OWS)	1.00
Plug-In Match	(RWS ⊃ OWS)	0.75
Subsumption Match	(RWS ⊂ OWS)	0.50
Intersection Match	(RWS ∩ OWS)	0.25
Fail/Non-Match	(RWS ≠ OWS)	0.00

6.2. Self-adaptive MWS matchmaking algorithm

Matchmaking is the process of matching advertised MWS and requested MWS, to identify the most relevant MWS. The matchmaking technique is created based on the requested capabilities and the provided capabilities of MWS to please users. Thus a logic-based matchmaking technique is incorporated into the M-NSA. The strategy includes a set of implemented operations to ensure that the most appropriate MWS is returned per the five degrees of match shown in [Table 1](#) and steps 8 through 19 of [Fig. 13](#).

An exact match occurs when the offered web service (OWS) and the requested web service (RWS) are equivalent concepts, a plug-in match occurs when the RWS is a super-concept of the OWS, subsumption match occurs when the RWS is a sub-concept of the OWS, an intersection match occurs when the intersection between the OWS and the RWS is acceptable, and a fail/non-match occurs when the OWS and the RWS are not equivalent concepts.

Algorithm 4: The self-adaptive matchmaking algorithm

```

1.   Input: The set of Offered MWS ( $O$ );
2.        $R_s$ : Requested MWS ( $R$ );
3.        $D_d$ : Initial set of detectors and the threshold  $\tau$ 
4.   Output: The top-N  $S$  that best fit the request
5.    $S \leftarrow \emptyset$ 
6.   for each service offer  $s \in O$  do
7.       Calculate the similarity of  $s$  with every detector  $d$  in  $D$ ,  $\forall d \in D$  using
8.        $Dist(s, d) = \sqrt{\sum_{i=1}^n (s_i - d_i)^2}$ 
9.       Check the Distance  $Dist(s, d)$ ;
10.      if  $R_s \equiv O_s \vee R_s = O_s$  // Exact Match
11.           $weight \leftarrow 1$  // Assign weight +=1
12.          add the new  $s$  to Top-N  $S$ ;
13.      else if  $R_s \supset O_s$  // Plug-In Match
14.           $weight \leftarrow 0.75$  // Assign weight +=0.75
15.          add the new  $s$  to Top-N  $S$ ;
16.      else if  $R_s \subset O_s$  // Subsumption Match
17.           $weight \leftarrow 0.5$  // Assign weight +=0.5
18.          add the new  $s$  to Top-N  $S$ ;
19.      else if  $R_s \cap O_s$  // Intersection Match
20.           $weight \leftarrow 0.25$  // Assign weight +=0.25
21.          add the new  $s$  to Top-N  $S$ ;
22.      else if  $R_s \neq O_s$  // Undesirable/Fail Match
23.           $weight \leftarrow 0$  // Assign weight +=0
24.      end if
25.   end do
26.   for each service request  $R_s$  do
27.       if  $Dist(R_s, D) \geq \tau$  // Updating the detector set
28.           Return Top-N  $S$ 
29.       else Update  $D$ 
30.           Generate new  $d$  using Use new  $R_s$ 
31.           Add new  $d$  to  $D$ 
32.           Go to 7
33.   end if
34. end do
35. end for
36. end for

```

Fig. 13. The self-adaptive matchmaking algorithm.

Matching rules are what control the self-adapting MWS Matching algorithm. Matching rules are based on a distance measure, which plays a central role in the M-NSA. Field of study and dataset representations (binary or real-valued) are used as parameters for deciding the most suitable matching rules. As such, the Euclidean distance ($Dist(s, d) = \sqrt{\sum_{i=1}^n (s_i - d_i)^2}$) is adopted as a matching rule for the M-NSA as it is more efficient on real-valued data compared to the Minkowski distance ($Dist(X, Y) = (\sum_{i=1}^n |x_i - y_i|^p)^{1/p}$) and the Manhattan distance ($Dist(X, Y) = \sum_{i=1}^n |x_i - y_i|$) (Abid et al., 2017).

The affinity binding between detectors generated (d) in the service relevance learning stage and the new service request R_s is obtained, and the Top-N MWS are returned depending on the DOM. The descriptive steps of the self-adaptive MWS matchmaking algorithm are seen in Fig. 13. The matchmaker adapts the model produced during the SRL based on the MWS matchmaking result. Steps 22 to 27 in Fig. 13 explain the state that induces self-adaptation to occur.

The similarity between the detectors produced at the SRL stage and the MWS request R_s is matched to the threshold τ , if the similarity is higher than or equivalent to the threshold, then the MWS request R_s is eligible to be a candidate detector, otherwise, the positioning of the detectors will be updated only since some detectors of similar value already exist. This is because, when so many FNs exist, the detectors have to be optimized to ensure that the correct MWS is discovered. The two main causes of FN are the self-space gaps that cannot be filled by the created detector and the huge self-radius that covers non-self-space. In addition, as the collection of self-elements (MWS) changes as a result of the influx of new or unavailable MWS, it becomes necessary to optimize the detectors to deal with the changes.

7. Empirical evaluation

This section covers evaluation measures used for the evaluation of self-adaptive MWS discovery in DME. The rationale behind

experimental decisions and choices regarding test collections, parameter settings, evaluation metrics, baseline frameworks are also discussed. The experiments and comparisons are carried out using Anaconda Navigator v1.9.7. Protégé Desktop v5.5.0 and Pellet reasoner for ontologies modeling and reasoning, respectively. All the experiments are conducted on a computer equipped with 64-bits Windows 10, 2.10 GHz Intel® Core™ i5, and 4 GB memory.

7.1. Test collection

The identification of the dataset that will be used for the evaluation of the research is another critical component in this research. hRESTS-TC3 dataset (Saadon and Mohamad, 2015) and ProgrammableWeb datasets (Cao et al., 2017; Shi et al., 2018) are identified. The hRESTS-TC3 test collections provide 20 to 50 pre-defined queries together with a set of relevant services for each query which is normally used for matchmaking algorithm evaluation. The ProgrammableWeb dataset is one of the contemporary, consistent, and largest MWS collections from different domains publicly accessible for evaluation; besides, its daily evolution set it apart from other test collections within the Semantic Service Matchmaker Evaluation Environment (SME2). Therefore, these standard and publicly available datasets was used to evaluate the accuracy and performance of the self-adaptive MWS discovery framework in DME.

7.2. Performance & effectiveness criteria

Two performance criteria determine the success of the proposed self-adaptive MWS discovery framework in DME. This section provides an overview of different evaluation metrics used to assess the effectiveness and accuracy of the proposed self-adaptive MWS discovery framework. The primary goal of the self-adaptive MWS discovery framework in DME is to achieve better accuracy and effectiveness. Thus, for evaluating the accuracy and effectiveness of the self-adaptive MWS discovery framework in DME, various evaluation metrics (precision, recall, f-measure, etc.) shown in Table 2 have been used (Xie et al., 2019; Zhang et al., 2018). These performance evaluation metrics can be broadly categorized into two, namely, binary relevance and graded relevance (Blake et al., 2012).

7.3. Competing approaches for comparison

To better assess the accuracy and effectiveness of the proposed framework, the best performing settings of the SRL model were first identified, then a further comparison has been made with state-of-the-art baselines frameworks was made. The selected baseline frameworks focused on improving MWS discovery, used similar datasets, and similar evaluation metrics to measure the results as applicable in the proposed framework. It is also important to note that, per the standard practice in research, the analysis explicitly uses the findings of the selected state-of-the-art framework as stated in their original papers and were directly compared with the proposed framework. The following are the state-of-the-art baselines frameworks which are summarized in Table 3.

- **RTM-FM** (Cao et al., 2019): This is a machine learning-based framework that uses the relational topic model to discover the relationship amongst MWS concepts and then use factorization machines to determine MWS for a given query.
- **GSD & KWSD** (Zhang et al., 2018): This framework returns MWS by comparing its capabilities with those found in an elaborated query through NLP and topic-based models.

- **DHCF** (Xiong et al., 2018): This is the deep learning-based framework that uses the point-wise regression model to convert the MWS matchmaking to a classification challenge.
- **GLDA** (Xie et al., 2019): This is a machine learning-based framework that uses pLSA and LDA to grasp and match the underlying concepts of MWS and queries.
- **NCF** (He et al., 2017): This is a machine learning-based framework that utilizes a neural network model to learn an explicit capability from MWS and obtain a nonlinear relationship amongst MWS and then produce matchmaking results per a given request.

7.4. Evaluation results & analysis

One of the goals of the proposed framework is to attain an improved performance in comparison to state-of-the-art MWS discovery frameworks. To evaluate the performance of the proposed framework, several experiments were conducted. These experiments investigate various aspects including, quality of service relevance learning, the quality of the discovery in terms of the binary and graded relevance, as well as the comparative evaluation to demonstrate the attained improvement.

7.4.1. Accuracy & effectiveness of service relevance learning

The success of the proposed matchmaking algorithms is determined by the ability of the SRL algorithm to have a high detection rate (DR) and a low false alarm rate (FAR). The DR is similar to precision while the FAR is the MWS incorrectly label as irrelevant (false positives) divided by the total irrelevant MWS (false positives plus true negatives).

To better investigate the effectiveness of the SRL model, three different runs were first designed based on the various settings of three important parameters and then identify the best performing of the SRL run to be used in the MWS matchmaking. The experiment was carried out on all the possible combinations to determine the best and the worst combination. Fig. 14 shows the cross-section design of different combinations of the three parameters. It should be noted that all hyperparameters were chosen based on the search space which contains eight different categories of MWS, 1700 candidate detectors, and 24 requests. These versions are described as follows:

- **MNSA-1:** This run was designed to study the impact of the self-radius r_s . In this run, the self-radius r_s was initialized at 0.01 and was tested with 100 maximum number of detectors D_{max} , and 99% estimated coverage E_c .
- **MNSA-2:** This run was designed to study the impact of the maximum number of detectors D_{max} . In this run, the maximum number of detectors D_{max} was initialized at 200 and was tested with 0.03 self-radius r_s , and 90% estimated coverage E_c .
- **MNSA-3:** This run was designed to study the impact of the estimated coverage E_c . In this run, the estimated coverage E_c was initialized at 80% and was tested with 0.05 self-radius r_s , and 300 maximum number of detectors D_{max} .

Table 4 shows the summary of experimental results of a different combination of the three parameters. Each value of the self-radius r_s (0.01, 0.03, and 0.05) was tested with the three different maximum number of detectors D_{max} (100, 200, and 300) and different estimated coverage E_c (80%, 90%, and 99%). The values of the self-radius r_s is increased by 0.02 for every run. This is because no changes were observed in the detection rate (DR) of the false alarm rate (FAR) when the values of the self-radius r_s was increased by 0.01. The maximum number of detectors D_{max} is also increased by 100 as anything below 100 does not result in noticeable changes in terms of DR and FAR during the runs.

Table 2
The evaluation metrics.

Metric	Formula	Description
Recall	$Rec = \frac{TP}{TP + FN}$	Recall refers to the percentage of ground-truth MWS correctly discovered as relevant
Precision	$Prec = \frac{TP}{TP + FP}$	Precision refers to the percentage of correctly discovered MWS as relevant out of all MWS
F1 score	$F1 = 2 \times \frac{Prec \times Rec}{Prec + Rec}$	F1-score refers to the harmonic mean of the precision and the recall
MAP	$MAP = \frac{\sum_{q=1}^Q AveP(q)}{Q}$	Mean Average Precision refers to the average of AP, which give more weight to errors that happen high up in the list of discovered MWS
NDCG	$NDCG = \frac{1}{Z} \sum_{k=1}^N \frac{2^{rel_k}}{\log_2(k+1)}$	Normalized Discounted Cumulative Gain normalizes the DCG, which discounts each discovered MWS based on its position in the results to a number between 0.0 and 1.0, given that some MWS are more relevant than others.

Table 3
Summary of the baseline frameworks.

Framework	Assumptions and Limitations	Adaptation Actions	Adaptation Period	Class	Accuracy
RTM-FM	High sparsity of the interaction matrix and difficult to comprehend algorithm structure	Process restructuring	Design	Decision Boundary	High
GSD & KWSD	Query extension complexities and disregard of semantics	Service withdrawal	Design	Probabilistic	Low
DHCF	Knowledge engineering effort to bootstrap, partial score issues, and high complexity	Process restructuring	Design	Nonlinear functional approximation	High
GLDA	Can only obtain limited semantically coherent topics without word embedding	Service withdrawal	Design	Probabilistic	Low
NCF	Cold start, data sparsity, gray ship problems, and prone exploding and gradient vanishing	Process restructuring	Design	Nonlinear functional approximation	High
SAMM	Low-high complexity and the best solutions majorly depend on a fitness function	Process restructuring	Runtime	Nonlinear functional approximation	High

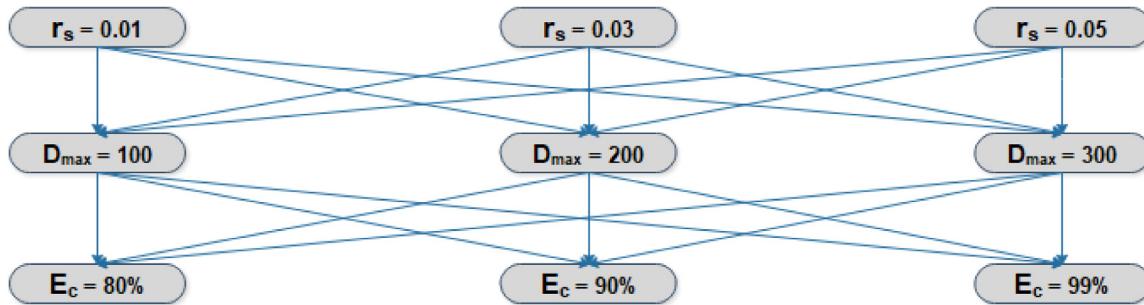


Fig. 14. The cross-section design of different combination of the three parameters.

The estimated coverage E_c was started at 80% as anything below 80 may result in huge FAR, it was also capped at 99% as 100% is mostly impractical.

Fig. 15 demonstrates the different versions of the SRL model. It can be observed from the results that the MNSA-1 version outperforms all the other versions of the model, namely, MNSA-2, and MNSA-3 in terms of detection rate (DR) as well as false alarm rate (FAR). For example, one can see that compared to the MNSA-2 version which utilized bigger self-radius r_s , and more number of D_{max} , the MNSA-1 performs better with significant improvements of 0.8% in terms of DR, and a comparable FAR of 0.26%. Moreover, the comparison with MNSA-3 which utilized bigger estimated coverage E_c , and a greater number of D_{max} , the MNSA-1 performs better with significant improvements of 1.1% in terms of DR, and a comparable FAR of 0.22%. This directly translates the advantage of exploiting the self-radius r_s , estimated coverage E_c , and the maximum number of detectors D_{max} for more effective and accurate MWS relevance learning. It is worth noting that, despite

that both the MNSA-2, and MNSA-3 rely on more detectors and coverage, however, the MNSA-1 outperformed the MNSA-2, and MNSA-3 given that the self-radius has more influence than other parameters.

7.4.2. Accuracy & effectiveness of matchmaking

The success of the proposed matchmaking algorithms is determined by the ability of the matchmaking algorithm to differentiate between MWS that are relevant to a given request and those that are irrelevant, usually known as the binary test. This is followed by the ability of the algorithm to return with the correct estimation of the degree of relevance, usually known as a graded-relevance test. To evaluate the quality of the matchmaking, the value of two important parameters (τ and N) need to be ascertained to ensure the most relevant services are discovered. The threshold τ is used to determine the need for generating new detectors in a situation where a given query does not yield a relevant result. While the number of relevant services to be returned for a given request is determined by $Top - N$.

Table 4
The Performance of the different runs of MNSA.

Algorithm	Self-radius $r_s = 0.01$		Self-radius $r_s = 0.03$		Self-radius $r_s = 0.05$	
	DR %	FAR %	DR %	FAR %	DR %	FAR %
MNSA-1	0.92	0.26	0.88	0.25	0.79	0.10
MNSA-2	0.84	0.25	0.81	0.22	0.72	0.07
MNSA-3	0.81	0.22	0.77	0.18	0.69	0.03
Algorithm	No. of D. $D_{max} = 100$		No. of D. $D_{max} = 200$		No. of D. $D_{max} = 300$	
	DR %	FAR %	DR %	FAR %	DR %	FAR %
MNSA-1	0.92	0.26	0.93	0.52	0.93	0.53
MNSA-2	0.84	0.25	0.85	0.43	0.85	0.46
MNSA-3	0.81	0.22	0.82	0.40	0.82	0.43
Algorithm	Estimated coverage $E_c = 80\%$		Estimated coverage $E_c = 90\%$		Estimated coverage $E_c = 99\%$	
	DR %	FAR %	DR %	FAR %	DR %	FAR %
MNSA-1	0.78	0.43	0.86	0.41	0.92	0.26
MNSA-2	0.72	0.40	0.75	0.39	0.84	0.26
MNSA-3	0.69	0.40	0.72	0.38	0.81	0.22

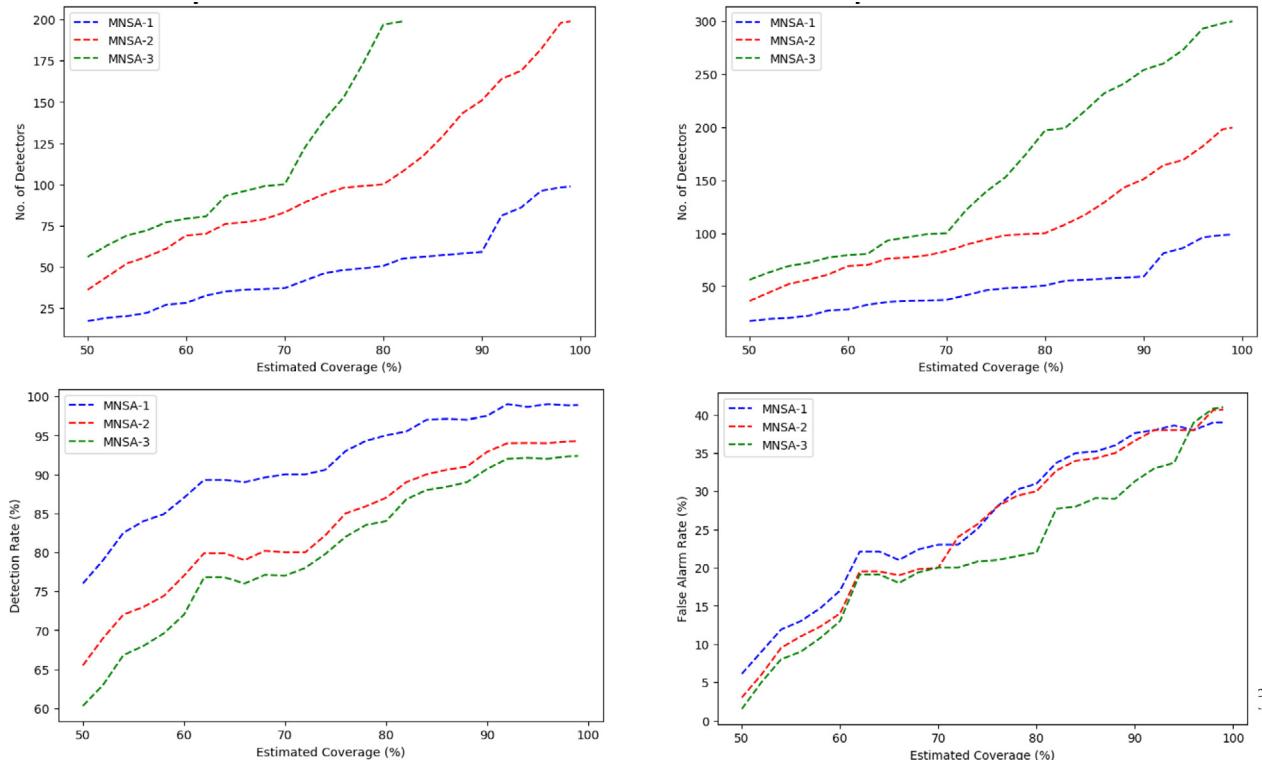


Fig. 15. The Comparison results of different versions of MNSA.

The threshold value was set between 0.1 to 0.5 as anything above 0.5 results in the inclusion of irrelevant discovery of MWS and lower the precision. The $Top - N$ is set at 10, 15, 20, 25, 30, 35, 40, 45, and 50. This is because $Top - N$ set at 10 to 50 is quite adequate to cover the most relevant MWS as well as disregard less relevant MWS. The experiments have been performed for the MWS discovery problem, which contains 24 requests all over the 8 defined domains. Each test was replicated dozens of times for every one of the 24 MWS requests and the mean value was taken.

The precision, recall, F1-score, MAP, and NDCG values for the proposed SAMM framework are given in Table 5 and plotted in Fig. 16. The highest average precision, recall, F1-score, MAP, and NDCG of the SAMM framework are 0.9167, 0.8430, 0.7237, 0.5028, and 0.8613 respectively. As can be seen, there is a trade-off between precision and recall which is acceptable given that the most relevant MWS is the target of the framework. The bar graph in Fig. 16 shows the plot of the average precision, recall, F1-score, MAP, and NDCG for different numbers of returned MWS.

Table 5
The evaluation results of the proposed SAMM framework.

Metrics	N = 5	N = 10	N = 15	N = 20	N = 40	N = 50
Precision	0.9167	0.8413	0.7745	0.7155	0.5888	0.5450
Recall	0.4991	0.6106	0.6614	0.7320	0.8327	0.8430
F-Measure	0.6463	0.7076	0.7135	0.7237	0.6898	0.6620
MAP	0.4597	0.4779	0.4849	0.4884	0.5028	0.4942
NDCG	0.8559	0.8613	0.8575	0.8439	0.7787	0.7066

The X-axis of the graph shows the 10 top-N while the Y-axis shows the score of each parameter.

After a thoughtful analysis, it is safe to say that SAMM's best accuracy is mainly because of the service relevance learned using MNSA, which contributes to the discrimination of irrelevant MWS from a set of relevant MWS. The accuracy of SAMM is owing to the additional consideration of variable-sized detector and the elimination of other unnecessary control parameters such as

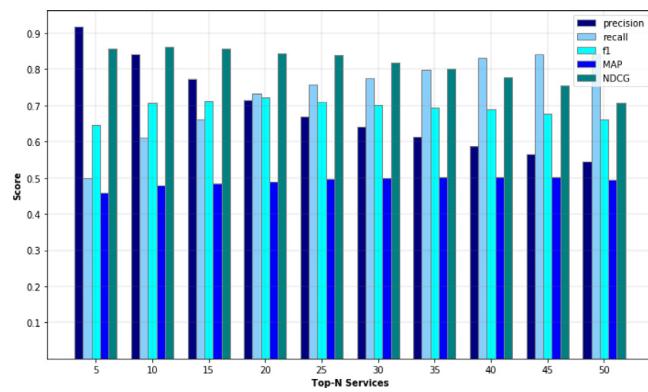


Fig. 16. The evaluation results of the proposed SAMM framework.

Table 6
The SAMM framework comparison with related work.

Frameworks	Prec.	Rec.	F1	MAP	NDCG
GDS	0.8498	0.8308	0.7088	0.4327	0.8632
DHCP	0.1719	0.8010	0.2326	0.3794	0.5062
GLDA	0.8002	0.6787	0.6204	0.4910	0.6917
RTM-FM	0.6896	0.6455	0.4891	0.3546	0.3146
KWSD	0.1023	0.7145	0.1571	0.4077	0.6897
NCF	0.1641	0.6695	0.2302	0.2920	0.4042
SAMM	0.9167	0.8430	0.7237	0.5028	0.8613

maximum age the detector (t) in the service relevance learning process contributes greatly to the higher accuracy of SAMM as well as the way it leverages more comprehensive information that covers the entire discovery space (service requester, service provider, and the environment) to achieve better accuracy. The comparative evaluation further discussed the accuracy of matchmaking.

7.4.3. Comparative evaluation results & discussion

To better assess the effectiveness of our proposed approach, a comparison was made with the existing methods. To this end, the best-performing setting of the proposed SRL (MNSA-1) model was selected and integrated into the matchmaking. The section first reports and discusses the comparison of the proposed framework with the state-of-the-art baseline frameworks on MWS datasets in terms of Recall, Precision, F1-score, MAP, and NDCG. Table 6 shows the comparison of the proposed SAMM framework with state-of-the-art frameworks and it can be observed that the proposed framework outperformed the existing framework across the evaluation metrics.

Compared to the RTM-FM based framework that uses the relational topic model and GLDA based framework that use pLSA + LDA, the proposed framework performed better with an improvement of 11.65%, 16.43%, 10.33%, 1.18%, and 16.96% or more on the ProgrammableWeb datasets in terms of the precision, recall, F1-score, MAP and NDCG respectively. This shows the advantage of the MNSA-based matchmaking compared to the RTM-FM-based and GLDA based framework. Such is due to the high sparsity of the MTR-FM interaction matrix that stifles the accuracy of the classifier of MWS. Moreover, the LDA hardly obtains semantically coherent topics in the absence of word embedding or other strategies.

Also, it can be shown from Table 6 and Fig. 17, the proposed framework performed better than NCF and DHCF with slight gains of 0.75%, 0.17%, 0.49%, 0.21%, and 0.46%, or more on the ProgrammableWeb datasets in terms of the precision, recall, F1-score, MAP and NDCG respectively. This can be attributed to the

fact that the NCF and DHCF based framework is a variant of the RNN and consequently have partial score issues, unlike the proposed MNSA-based framework.

The result of comparison with GSD & KWSD based framework indicates that the MWS goals can provide significant support for MWS discovery if extracted appropriately, which also validates the decision to integrate MWS goals at the base of SAMM. Yet, the proposed framework performed better than GSD & KWSD based framework with slight gains of 0.75%, 0.17%, 0.49%, 0.21%, and 0.46%, or more on the ProgrammableWeb datasets in terms of the precision, recall, F1-score, MAP and NDCG respectively. This is because the matchmaking in these frameworks lacks a clear conceptual connection between MWS goals within the request and needs highly-complex queries that require extensive refinement.

More specifically, it is reasonable to conclude that the best results obtained by the SAMM framework are primarily due to the importance of the SRL, which leads to the classification of irrelevant MWS and relevant MWS. The effectiveness of the SAMM framework is due to the additional consideration of the variable-size detector as well as the removal of other unwanted design variables such as the age limit of the detector (t) in the SRL process, which substantially leads to the increased accuracy and effectiveness of SAMM framework. Moreover, the search space reduction through MWS categorization and the use of more detailed knowledge covering the whole discovery space (service requester, service provider, and the environment) contributes to the performance of the SAMM framework.

Contrary to the Goal-based MWS categorization approaches discussed in Section 2 that neglected the semantic knowledge of goals and tags, less accuracy, high influx of similar MWS, and many other issues, the proposed MWS categorization approach in this study introduces a qualitative feature extraction technique based on NLP for capturing both goals, tags, recognize the semantic knowledge, and specifically utilizes NSA for MWS categorization to improve the accuracy of the categorization. Contrary to the Context model for MWS discovery discussed in Section 2 that are constraints by predefined contexts allocated with arbitrary values, the use of limited contextual information to make a generalized decision, the proposed meta-context ontology organizes and representation the heterogeneous context information which is critical in differentiating relevant and irrelevant MWS, thus improve the accuracy and effectiveness of MWS discovery framework. Contrary to the self-adaptive MWS discovery discussed in Section 2 that are constraints by infinite loop, data sparsity, design-time adaptation, cold-start problems, unpredictable changes in DME, irrelevant MWS discovery, the proposed self-adaptive MWS discovery approaches in this study introduces a service relevance learning model and matchmaker based on M-NSA to deal with these issues and improve the accuracy and effectiveness of the self-adaptive MWS discovery framework.

8. Limitations and threats to validity

8.1. Limitations

There are some limitations to the proposed SAMM framework. As for the MWS categorization approaches, the crucial part is feature selection and extraction, after which the semantic similarities are calculated. The feature selection and extraction were implemented using NLP. However, the application of stop-words removal, lemmatization, and non-linear dimensionality reduction technique may lead to loss of semantics. Consequently, not all MWS features were extracted and categorized. As for the meta-context model, the complexity of knowledge acquisition is not

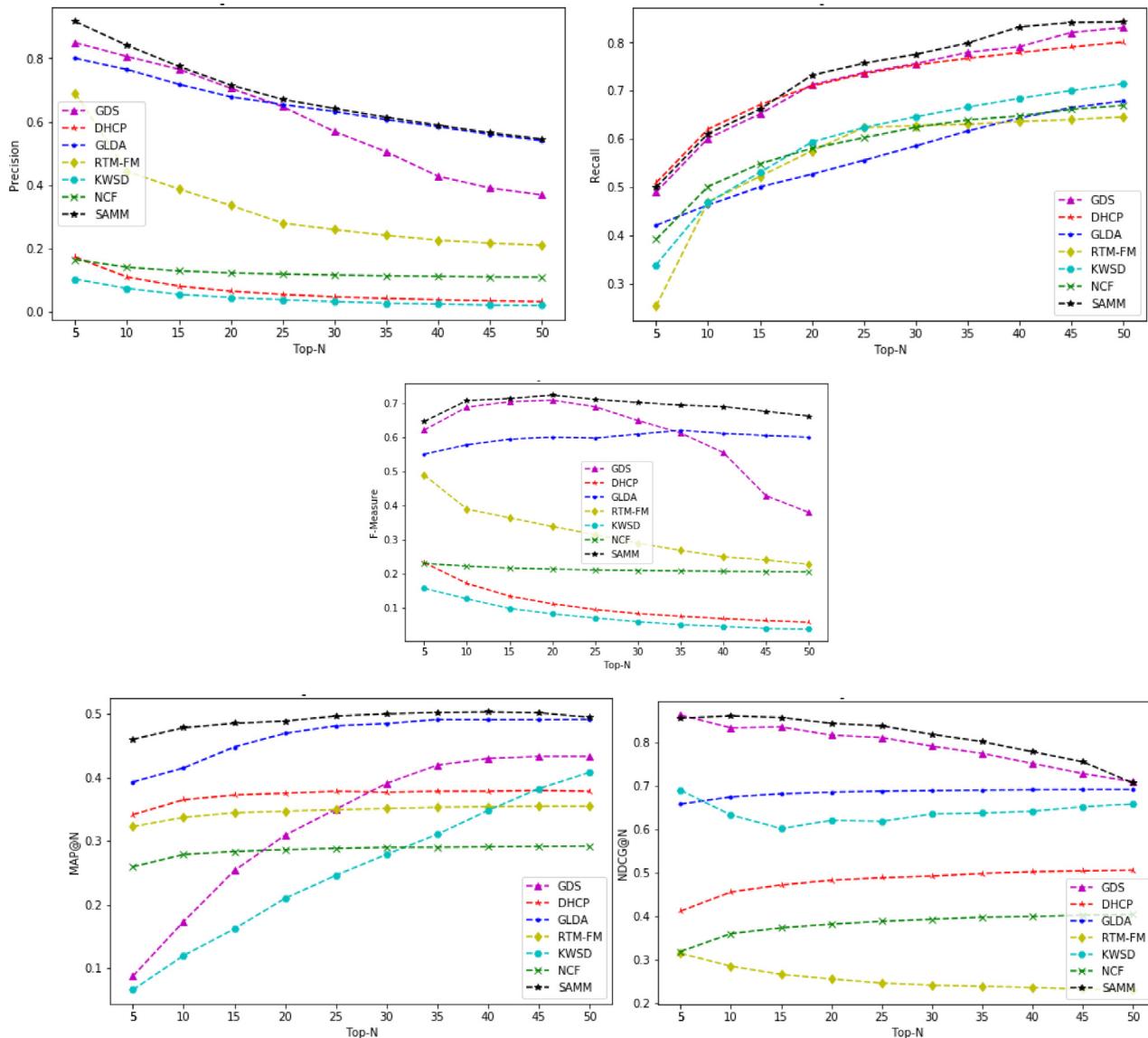


Fig. 17. The graphical representation of the results SAMM compared to baselines.

comprehensively covered in the model. UPON Lite is designed for the speedy development of a “lightweight” ontology, making it susceptible to ignoring other stakeholders. The Pellet Reasoner is utilized in evaluating the ontology, but it is just for an experimental tool that does not reflect the outcome of the real system in DME. As for the self-adaptive matchmaker, the algorithms (service relevance learning and MWS matchmaking algorithm) are designed and implemented based on a modified negative selection algorithm. Nevertheless, metaheuristic algorithms such as the negative selection algorithm used to solve a wide range of web service discovery, selection, and composition challenges are often unpredictable when it comes to the assessment of accuracy and effectiveness. Alternative options for reducing the time and space challenges of SRL in the future must also be debated.

8.2. Threats to validity

Validity is frequently associated with the accuracy of the research and the level of reliability of the results obtained from

the research (Andreini and Bettinelli, 2017). Threats are inevitable when experimenting. The threat to the validity of this research can come from any of the four major sources (Construct Validity, Internal Validity, External Validity, and Empirical Reliability). These threats are discussed as follows.

Construct validity refers to the extent to which the experiment setting reflects the concept or theory behind the experiment, ascertaining that the solution reflects the construct of the cause and that the result reflects the construct of the effect. In this research, a threat may come from the insufficient preoperational description of constructs, failure to appraise the validity of effectiveness measures, experimenter expectancies. To mitigate these threats, the constructs such as performance, and effectiveness, are sufficiently defined and are translated into observable measures such as the average precision (AP), F-measure, MAP at standard recall levels, NDCG to ensure correctness. These measures have a clear interpretation in MWS discovery and have the suitable discriminating power to ensure that observed results uphold or oppose the proposed concept.

Internal validity refers to the extent to which the results obtained in a research study are a function of the independent and dependent variables that were methodically manipulated, measured, and/or observed in the study rather than uncontrolled confounding variables. In this research, the threat may come from failure to set parameters appropriately, the absence of real instances of the problem, unclear data collection processes, and tools. Replication packages are provided in the web link "<https://github.com/salisugarba4all/SAMM>" to confirm the results of the framework, support enhancement of the approach and the observed conclusions. The PW API dataset from which the goals of the service are extracted is crawled from: <https://www.programmableweb.com>, moreover, the selected parameters, and values of the proposed self-adaptive MWS discovery framework for DME are clearly described in the experimental design. These will enable regeneration of published results from author-provided code and data under the same conditions of the experiment (Erik, 2021).

External validity refers to the extent to which the results of a research study can be generalized to a larger group (industrial practice), outside the sample instances used in the experiment. In this research, a threat may come from poorly defined target instances, vague object selection strategy, disregard for size, cost, and complexity. Given the discovery issues addressed by the framework under evaluation, the strategy of objects such as the ProgrammableWeb dataset offered MWS and user queries, that are collected from real-world problems for this research are well defined and justified in the experimental design to ensure the possibility of generalization.

9. Conclusions and future work

This study proposes a self-adaptive mobile web service (MWS) discovery framework for a dynamic mobile environment (DME) by leveraging NSA to deal with MWS proliferation, dynamic context, and challenges in the task of discovering relevant MWS. To address these challenges, firstly, an enhanced MWS categorization approach is designed to exploit both MWS documents and tags effectively, and then categorize other MWS based on semantic similarities. This is achieved by extracting the goals and tags from the functional description of MWS and then subsuming k-means in the modified negative selection algorithm (M-NSA) to create categories that contain similar MWS. By classifying the MWS into functionally similar categories, the matching is only performed on the MWS in the target category which improves the overall accuracy and effectiveness of the SAMM framework. Secondly, a semantically rich, reusable, and scalable context management model is designed, and an extensible meta-context ontology is proposed to represent the dynamic and heterogeneous context information for the SAMM framework. This is conducted using the lightweight unified process for ontology building (UPON-Lite) in collaboration with the feature-oriented domain analysis (FODA). The use of a well-organized context in the creation of the SRL model ensures context-inclusive matching rather than a separate context matching task. This had unraveled the context issues for the SAMM framework thereby improving the effectiveness of the SAMM framework. Thirdly, a SRL algorithm based on MNSA, and a self-adaptive matchmaker that adapts at runtime was designed to improve the accuracy and effectiveness of the self-adaptive MWS discovery framework in DME. Series of experiments carried out using publicly-available datasets to evaluate the performances of the proposed framework against the state-of-the-art frameworks exhibits significant improvements in terms of effectiveness and accuracy. The shortcomings mentioned in Section 8.2 will act as a reference for future work. For instance, Part-of-Speech (POS) tagging, and Deep Auto Encoder (AE) can be

applied for feature selection and representations. The ontology proposed is extendable but not adaptable, therefore, ontologies that adapt automatically to the evolution of the modeled domain can be used to improve the representation of both the MWS discovery domain as well as the user's profile in the future. Alternative options such as Deep Belief Networks (DBNs) can be explored to improve SRL despite the inefficient MWS data for training. Moreover, a prototype can be developed based on the proposed framework to improve the effectiveness of MWS discovery thereby and improving the service-based applications.

CRediT authorship contribution statement

Salisu Garba: Data curation, Formal analysis, Investigation, Methodology, Resources, Writing – original draft, Writing – review & editing. **Radziah Mohamad:** Project administration, Resources, Software, Supervision, Writing – review & editing. **Nor Azizah Saadon:** Resources, Visualization, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This research has been supported by Ministry of Higher Education (MOHE) through Fundamental Research Grant Scheme (FRGS/1/2021/ICT01/UTM/02/1).

References

- Abid, A., Khan, M.T., de Silva, C.W., 2017. Layered and real-valued negative selection algorithm for fault detection. *IEEE Syst. J.* 1–10. <http://dx.doi.org/10.1109/JST.2017.2753851>.
- Aguilar, J., Jerez, M., Rodríguez, T., 2018. CAMeOnto: Context awareness meta ontology modeling. *Appl. Comput. Inform.* 14, 202–213. <http://dx.doi.org/10.1016/j.aci.2017.08.001>.
- Andreini, D., Bettinelli, C., 2017. *Systematic literature review*. In: *Business Model Innovation*. Springer, pp. 1–23.
- Barakat, L., Miles, S., Luck, M., 2018. Adaptive composition in dynamic service environments. *Future Gener. Comput. Syst.* 80, 215–228. <http://dx.doi.org/10.1016/j.future.2016.12.003>.
- Blake, M.B., Cabral, L., König-Ries, B., Küster, U., Martin, D., 2012. Semantic web services: Advancement through evaluation. *Semant. Web Serv. Adv. Through Eval.* 1–359. <http://dx.doi.org/10.1007/978-3-642-28735-0>.
- Bobek, S., Nalepa, G.J., 2017. Uncertain context data management in dynamic mobile environments. *Future Gener. Comput. Syst.* 66, 110–124. <http://dx.doi.org/10.1016/j.future.2016.06.007>.
- Bouguettaya, A., Singh, M., Huhns, M., Sheng, Q.Z., Dong, H., Neiat, A.G., Mistry, S., Benatallah, B., Medjahed, B., Casati, F., Liu, X., Wang, H., Georgakopoulos, D., Malik, Z., Erradi, A., Wang, Y., Blake, B., Leymann, F., Papazoglou, M., Bouguettaya, A., Singh, M., Huhns, M., Sheng, Q.Z., Dong, H., Yu, Q., Neiat, A.G., Mistry, S., Benatallah, B., Medjahed, B., et al., 2017. A service computing manifesto : The next ten years. *Commun. ACM* 60, 64–72. <http://dx.doi.org/10.1145/2983528>.
- Cabrera, O., Franch, X., Marco, J., 2017. Ontology-based context modeling in service-oriented computing: A systematic mapping. *Data Knowl. Eng.* 110, 24–53. <http://dx.doi.org/10.1016/j.datalab.2017.03.008>.
- Cao, B., Frank Liu, X., Liu, J., Tang, M., 2017. Domain-aware mashup service clustering based on LDA topic model from multiple data sources. *Inf. Softw. Technol.* 90, 40–54. <http://dx.doi.org/10.1016/j.infsof.2017.05.001>.
- Cao, B., Liu, X., Li, B., Liu, J., Tang, M., Zhang, T., Shi, M., 2016. Mashup service clustering based on an integration of service content and network via exploiting a two-level topic model. In: Proc. - 2016 IEEE Int. Conf. Web Serv. ICWS 2016, pp. 212–219. <http://dx.doi.org/10.1109/ICWS.2016.35>.
- Cao, B., Liu, J., Tang, M., Zheng, Z., Wang, G., 2013. Mashup service recommendation based on user interest and social network. In: Proc. - IEEE 20th Int. Conf. Web Serv. ICWS 2013, pp. 99–106. <http://dx.doi.org/10.1109/ICWS.2013.23>.

- Cao, B., Liu, J., Wen, Y., Li, H., Xiao, Q., Chen, J., 2019. QoS-aware service recommendation based on relational topic model and factorization machines for IoT mashup applications. *J. Parallel Distrib. Comput.* 132, 177–189. <http://dx.doi.org/10.1016/j.jpdc.2018.04.002>.
- Chen, L., Hu, L., Zheng, Z., Wu, J., Yin, J., Li, Y., Deng, S., 2011. WTCluster: Utilizing tags for web services clustering. In: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. pp. 204–218. http://dx.doi.org/10.1007/978-3-642-25535-9_14.
- Chen, F., Li, M., Wu, H., Xie, L., 2017. Web service discovery among large service pools utilising semantic similarity and clustering. *Enterp. Inf. Syst.* 11, 452–469. <http://dx.doi.org/10.1080/17517575.2015.1081987>.
- Cheng, B., Li, C., Zhao, S., Chen, J., 2018. Semantics mining & indexing-based rapid web services discovery framework. *IEEE Trans. Serv. Comput.* <http://dx.doi.org/10.1109/TSC.2018.2831678>.
- Cheniki, N., Belkhir, A., Atif, Y., 2016. Mobile services discovery framework using dbpedia and non-monotonic rules. *Comput. Electr. Eng.* 52, 49–64. <http://dx.doi.org/10.1016/j.compeleceng.2016.02.016>.
- Chouiref, Z., Belkhir, A., Benouaret, K., Hadjali, A., 2016. A fuzzy framework for efficient user-centric web service selection. *Appl. Soft. Comput.* 41, 51–65. <http://dx.doi.org/10.1016/j.asoc.2015.12.011>.
- De Nicola, A., Missikoff, M., 2016. A lightweight methodology for rapid ontology engineering. *Commun. ACM* 59, 79–86. <http://dx.doi.org/10.1145/2818359>.
- Elgazzar, K., Hassanein, H., Martin, P., 2014. Daas: Cloud-based mobile web service discovery. *Pervasive Mob. Comput.* 13, 67–84. <http://dx.doi.org/10.1016/j.pmcj.2013.10.015>.
- Erik, G.O., 2021. The fundamental principles of reproducibility. *Phil. Trans. R. Soc. A* 379, 20200210.
- Fantechi, A., Gnesi, S., Semini, L., 2019. From generic requirements to variability. In: *CEUR Workshop Proceedings*.
- Fielding, R.T., 2000. REST: Architectural Styles and the Design of Network-Based Software Architectures (Doctoral dissertation). University of California.
- Garba, S., Mohamad, R., Saadon, N.A., 2020a. Search space reduction approach for self-adaptive web service discovery in dynamic mobile environment. In: Saeed, F., Mohammed, F., Gazem, N. (Eds.), *Emerging Trends in Intelligent Computing and Informatics*. Springer International Publishing, Cham, pp. 1111–1121.
- García, J.M., Ruiz, D., Ruiz-Cortés, A., 2010. A Model of User Preferences for Semantic Services Discovery and Ranking. In: *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6089 LNCS, pp. 1–14. http://dx.doi.org/10.1007/978-3-642-13489-0_1.
- Hamdan, N.M., Jawawi, D.N.A., 2012. The Integrated Method for a Systematic Approach in Developing Precision Farming Software Product Line Architecture. *Universiti Teknologi, Malaysia*.
- He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S., 2017. Neural collaborative filtering. In: *26th Int. World Wide Web Conf. WWW 2017*, pp. 173–182. <http://dx.doi.org/10.1145/3038912.3052569>.
- Jiang, B., Ye, L., Wang, J., Wang, Y., 2017. A semantic-based approach to service clustering from service documents. In: *2017 IEEE International Conference on Services Computing. SCC*, pp. 265–272. <http://dx.doi.org/10.1109/SCC.2017.41>.
- Ke, C., Huang, Z., 2012. Self-adaptive semantic web service matching method. *Knowl.-Based Syst.* 35, 41–48. <http://dx.doi.org/10.1016/j.knosys.2012.04.018>.
- Klusch, M., Kapahnke, P., 2012. The iSeM matchmaker: A flexible approach for adaptive hybrid semantic service selection. *J. Web Semant.* 15, 1–14. <http://dx.doi.org/10.1016/j.websem.2012.07.003>.
- Klusch, M., Kapahnke, P., Schulte, S., Lecue, F., Bernstein, A., 2016. Semantic web service search: A brief survey. *KI - Künstliche Intell.* 30, 139–147. <http://dx.doi.org/10.1007/s13218-015-0415-7>.
- Kotekar, S., Kamath, S.S., 2016. Enhancing service discovery using cat swarm optimisation based web service clustering. *Perspect. Sci.* 8, 6–8. <http://dx.doi.org/10.1016/j.pisc.2016.06.068>.
- Kotekar, S., Kamath, S.S., 2018. Enhancing web service discovery using meta-heuristic CSO and PCA based clustering. *Prog. Intell. Comput. Tech. Theory, Pract. Appl.* 519, 393–403. <http://dx.doi.org/10.1007/978-981-10-3376-6>.
- Liang, T., Chen, L., Ying, H., Wu, J., 2014. Co-clustering WSDL documents to bootstrap service discovery. In: *Proc. - IEEE 7th Int. Conf. Serv. Comput. Appl. SOCA 2014*, pp. 215–222. <http://dx.doi.org/10.1109/SOCA.2014.27>.
- Lu, Z.J., Li, G.Y., Pan, Y., 2017. A method of meta-context ontology modeling and uncertainty reasoning in SWoT. In: *Proceedings - 2016 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery. CyberC 2016*, pp. 128–135. <http://dx.doi.org/10.1109/CyberC.2016.34>.
- Mezni, H., Sellami, M., 2016. AWS-ont: An ontology for the self-management of service-based systems. In: *Proc. - 2015 IEEE 8th Int. Conf. Serv. Comput. Appl. SOCA 2015*, pp. 85–92. <http://dx.doi.org/10.1109/SOCA.2015.17>.
- Musen, M.A., 2015. The protégé project : A look back and a look forward. *Assoc. Comput. Mach. Specif. Interes. Gr. Artif. Intell.* 1, 4–12. <http://dx.doi.org/10.1145/2757001.2757003>.
- Pahlevan, A., Duprat, J.L., Thomo, A., Müller, H., 2015. Dynamis: Effective context-aware web service selection using dynamic attributes. *Futur. Internet* 7, 110–139. <http://dx.doi.org/10.3390/fi7020110>.
- Poveda-Villalón, M., Suárez-Figueroa, M.C., García-Castro, R., Gómez-Pérez, A., 2010. A context ontology for mobile environments. In: *CEUR Workshop Proc.* 626.
- Rupasingha, R.A.H.M., Paik, I., Kumara, B.T.G.S., 2017. Improving web service clustering through a novel ontology generation method by domain specificity. In: *Proceedings - 2017 IEEE 24th International Conference on Web Services. ICWS 2017*, pp. 744–751. <http://dx.doi.org/10.1109/ICWS.2017.134>.
- Ruta, M., Scioscia, F., Di Sciascio, E., 2015. A mobile matchmaker for resource discovery in the ubiquitous semantic web. In: *Proc. - 2015 IEEE 3rd Int. Conf. Mob. Serv. MS 2015*, pp. 336–343. <http://dx.doi.org/10.1109/MobServ.2015.76>.
- Saadon, N.A., Mohamad, R., 2011. WSMO-M: NFP-aware web service discovery for mobile computing. In: *2011 Malaysian Conf. Softw. Eng.* pp. 70–75. <http://dx.doi.org/10.1109/MySEC.2011.6140646>.
- Saadon, N.A., Mohamad, R., 2015. Semantic-based discovery framework for web services in mobile computing environment. *J. Technol.* 77, 25–38. <http://dx.doi.org/10.11113/jtv77.6183>.
- Scioscia, F., Ruta, M., Loseto, G., Gramegna, F., Ieva, S., Pinto, A., Di Sciascio, E., 2014. A mobile matchmaker for the ubiquitous semantic web. *Int. J. Semant. Web Inf. Syst.* 77–100. <http://dx.doi.org/10.4018/ijsws.2014100104>.
- Sellami, M., Bouchaala, O., Gaaloul, W., Tata, S., 2013. Communities of web service registries: Construction and management. *J. Syst. Softw.* 86, 835–853. <http://dx.doi.org/10.1016/j.jss.2012.11.019>.
- Shi, M., Liu, J., Cao, B., Wen, Y., Zhang, X., 2018. A prior knowledge based approach to improving accuracy of web services clustering. In: *2018 IEEE Int. Conf. Serv. Comput.* pp. 1–8. <http://dx.doi.org/10.1109/SCC.2018.00008>.
- Stavropoulos, T.G., Andreadis, S., Bassiliades, N., Vrakas, D., Vlahavas, I., 2016. The tomaco hybrid matching framework for SAWSL semantic web services. *IEEE Trans. Serv. Comput.* 9, 954–967. <http://dx.doi.org/10.1109/TSC.2015.2430328>.
- Taylor, M., 2018. Climate-smart agriculture: what is it good for? *J. Peasant. Stud.* 45, 89–107. <http://dx.doi.org/10.1080/03066150.2017.1312355>.
- Tian, G., Wang, Q., Wang, J., He, K., Zhao, W., Gao, P., Peng, Y., 2019a. Leveraging contextual information for cold-start web service recommendation. *Concurr. Comput.* 31, 1–17. <http://dx.doi.org/10.1002/cpe.5195>.
- Tian, G., Zhao, S., Wang, J., Zhao, Z., Liu, J., Guo, L., 2019b. Semantic sparse service discovery using word embedding and Gaussian LDA. *IEEE Access* 7, 88231–88242. <http://dx.doi.org/10.1109/ACCESS.2019.2926559>.
- Verma, R., Srivastava, A., 2018. A dynamic web service registry framework for mobile environments. *Peer-To-Peer Netw. Appl.* 11, 409–430. <http://dx.doi.org/10.1007/s12083-016-0540-6>.
- Win, N.N.H., Jianmin, B., Gang, C., Rehman, S.U., 2019. Self-adaptive qos-aware web service discovery using ontology approach. In: *Web Services: Concepts, Methodologies, Tools, and Applications. IGI Global*, pp. 822–841.
- Wu, J., Chen, L., Zheng, Z., Lyu, M.R., Wu, Z., 2014. Clustering web services to facilitate service discovery. *Knowl. Inf. Syst.* 38, 207–229. <http://dx.doi.org/10.1007/s10115-013-0623-0>.
- Xie, F., Wang, J., Xiong, R., Zhang, N., Ma, Y., He, K., 2019. An integrated service recommendation approach for service-based system development. *Expert Syst. Appl.* 123, 178–194. <http://dx.doi.org/10.1016/j.eswa.2019.01.025>.
- Xiong, R., Wang, J., Zhang, N., Ma, Y., 2018. Deep hybrid collaborative filtering for web service recommendation. *Expert Syst. Appl.* 110, 191–205. <http://dx.doi.org/10.1016/j.eswa.2018.05.039>.
- Zaki, M.Z.M., Jawawi, D.N.A., Hamdan, N.M., Halim, S.A., Mamat, R.R., Mahat, F.S., Omar, N.A., 2014. Multi-criteria architecture style selection for precision farming software product lines using fuzzy AHP. *Int. J. Adv. Softw. Comput. Appl.* 5, 1–23.
- Zhang, N., Wang, J., Ma, Y., He, K., Li, Z., Liu, X., (Frank), F., 2018. Web service discovery based on goal-oriented query expansion. *J. Syst. Softw.* 142, 73–91. <http://dx.doi.org/10.1016/j.jss.2018.04.046>.
- Zhao, X., Li, R., Zuo, X., 2019. Advances on QoS-aware web service selection and composition with nature-inspired computing. *CAAI Trans. Intell. Technol.* 4, 159–174. <http://dx.doi.org/10.1049/trit.2019.0018>.
- Zhao, X., Wen, Z., Li, X., 2014. QoS-aware web service selection with negative selection algorithm. *Knowl. Inf. Syst.* 40, 349–373. <http://dx.doi.org/10.1007/s10115-013-0642-x>.



Salisu Garba received the BSc. degree in computing and the MSc. degree in computing and information technology from the University of Wales. He is currently pursuing the Ph.D. degree in computer science with the Faculty of Engineering, School of Computing, Universiti Teknologi Malaysia. He is a Lecturer with the Department of Mathematics and Computer Science, Sule Lamido University, Kafin Hausa, Nigeria. His current research interests include semantic web service, mobile computing, self-adaptive systems, and Internet of Things. He is also a Member of IEEE, Computer Professionals of Nigeria, and the Nigeria Computer Society.



Radziah Mohamad is an Associate Professor in the School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia. She received her Bachelor of Software Engineering from Sheffield Hallam University, UK, Master of Science (Computer Science) and Ph.D. (Computer Science) from Universiti Teknologi Malaysia. She has been an academic staff at the Software Engineering Department for more than ten years. Prior to joining UTM, she was an associate consultant with Price Waterhouse for one year. She has authored or co-authored many academic papers in software engineering. Her research interests include service-oriented computing, software reuse, agent-oriented software engineering.



Nor Azizah Saadon is a Senior Lecturer in the School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia (UTM). She received her Bachelor of Information Technology (Computer Science) from Universiti Kebangsaan Malaysia (UKM), Master of Science (Computer Science) and Ph.D. (Computer Science) from Universiti Teknologi Malaysia (UTM). Prior joining UTM, she was a software engineer with Panasonic AVC Networks (M) Sdn. Bhd and was an IT Specialist at IBM Malaysia Sdn. Bhd for almost four years. She has been an academic staff at UTM for more than ten years. She involves in several research projects and her research interests include agent-oriented software engineering, service-oriented architecture, mobile and cloud.