IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON NOVEL LEARNING APPLICATIONS AND SERVICES FOR SMART CAMPUS

Received June 19, 2018, accepted July 15, 2018, date of publication July 19, 2018, date of current version August 15, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2857482

Crossing Scientific Workflow Fragments Discovery Through Activity Abstraction in Smart Campus

JINFENG WEN[1,2], ZHANGBING ZHOU[1,3], ZHENSHENG SHI[4], JUNPING WANG[5], YUCONG DUAN[6], AND YAQIANG ZHANG[1]

[1]School of Information Engineering, China University of Geosciences (Beijing), Beijing 100083, China
[2]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
[3]Computer Science Department, TELECOM SudParis, 91011 Evry Cedex, France
[4]PetroChina Research Institute of Petroleum Exploration and Development, Langfang 065007, China
[5]Laboratory of Precision Sensing and Control Center, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[6]College of Information Science and Technology, Hainan University, Haikou 570228, China

Corresponding author: Zhangbing Zhou (zhangbing.zhou@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61772479 and Grant 61662021, in part by the Open Foundation of State key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications under Grant SKLNST-2018-1-13, and in part by the Fundamental Research Funds for the Central Universities, China University of Geosciences (Beijing), China.

ABSTRACT Considering the knowledge-intensity and error-prone of developing scientific workflows from scratch, reusing and repurposing current workflows are the effective and efficient solution to support scientists for conducting novel experiments, and this strategy is deemed as important to achieve the objective of smart campus. An experiment may be relevant with one or multiple scientific workflows. This observation drives us to propose a technique that can discover and recommend cross-workflow fragments with respect to the requirement of novel experiments. Specifically, the functionally similar activities are clustered through adopting a modularity-based community discovery clustering technique, and they are represented as abstract activities. An abstract activity network model is constructed accordingly to reflect the invocation relations among abstract activities. Structural and semantic similar workflow fragments are discovered from the abstract activity network through the sub-graph matching algorithm. These fragments are instantiated through replacing abstract activities by appropriate activities in certain activity clusters. These instantiated workflow fragments are ranked and recommended for their reuse and repurposing purpose. Experimental evaluation results demonstrate that our technique is accurate and efficient on discovering and recommending appropriate cross-workflow fragments.

INDEX TERMS Scientific workflow, community discovery clustering, abstract activity, cross-workflow fragment recommendation.

I. INTRODUCTION

The explosive growth of scientific knowledge makes the inside-school education a challenge, especially for disciplines like life sciences where students are required to conduct complex scientific experiments often. Intuitively, the principles and wisdom of scientific experiments are encoded into and implicitly represented as scientific workflows. In this setting, reusing and repurposing scientific workflows can be regarded as an efficient and effective learning mechanism for research scientists when novel scientific experiments are to be designed and conducted. We argue that this is an indispensable ingredient for achieving the objective of smart campus [1], [2]. Generally, scientific workflows correspond to processes which should be executed recurrently by scientists to implement certain functionalities, where activities are specified upon recurring data and computational resources in terms of Web/REST services or mashup APIs, in order to promote the sharing of these recourses [3]. Along with the wide-adoption of Web 2.0 technology, nowadays scientific workflows can be publicly available on online repositories like *myExperiment*, *crowdLabs*, and *BioCatalogue*. As an example, *myExperiment* contains over 2,700 scientific workflows provided by systems like *Taverna 1/2*, *RapidMiner*, *Kepler*, and so on.

Considering the fact that developing a scientific workflow from scratch is definitely a knowledge-intensive and error-prone mission, when a scientist would like to conduct a novel scientific experiment, developing this requirement-oriented scientific workflow through reusing or repurposing *best-practices* evidenced by current scientific workflows should be a cost-effective and error-avoiding strategy [4]. When this requirement can be completely, or at least partially, satisfied by a single scientific workflow archived in a repository, the mechanism that can discover the most appropriate candidates is appropriate by adopting workflow similarity computation techniques [5]–[8]. It it worth emphasizing that a novel experiment may be relevant with multiple experiments that have been conducted by others in most application scenarios, and thus, whose workflow should be constructed through assembling carefully-discovered fragments contained in various scientific workflows in the repository. Consequently, discovering appropriate fragments from multiple scientific workflows, and facilitating the reuse and repurposing of these assembled fragments, is a promising research challenge.

As an example as shown by Fig. 1, a scientific workflow is typically represented as a directed graph, where vertices correspond to activities and edges reflect dependency relations specified upon contiguous activities [9]. This graph is mostly not large in size, but may be relatively complex in structure. Intuitively, developers may discover functionally-similar activities that have been developed by other scientific workflows from the repository, and try to reuse or repurpose these activities, when constructing new workflows, although some developers may prefer to create new, but functionally-similar, activities from scratch. This activity discovery procedure could be achieved through adopting techniques like service or resource discovery [10], [11]. However, this is usually a domain-expert oriented and laborious task, and the result may be inaccurate somehow, especially when a relatively large number of activities, as well as scientific workflows, have been achieved in the repository. In this setting, clusters of activities with diverse names and text descriptions should be identified, and they may be created by different developers, but are indeed functionality-equivalent or similar. For instance, there are totally 3,032 activities in *myExperiment*, and we observe that the activity "*common_pathway_describe*" in workflow "*Bio2RDF: Bind search, rdfise and load demo*" is actually similar to the activity "*common_pathway_link*" in workflow "*Bio2RDF: CPath search, rdfise and load demo*". If activities belonging to certain clusters are assumed as irrelevant completely, this strategy should negatively impact the reuse or repurposing of these activities in various workflow fragments. We argue that activities with same or similar functionalities should be considered as relevant when fragments crossing scientific workflows are to be reused or repurposed.

Workflow reuse and repurposing is a pressing research topic [12], and techniques have been proposed for

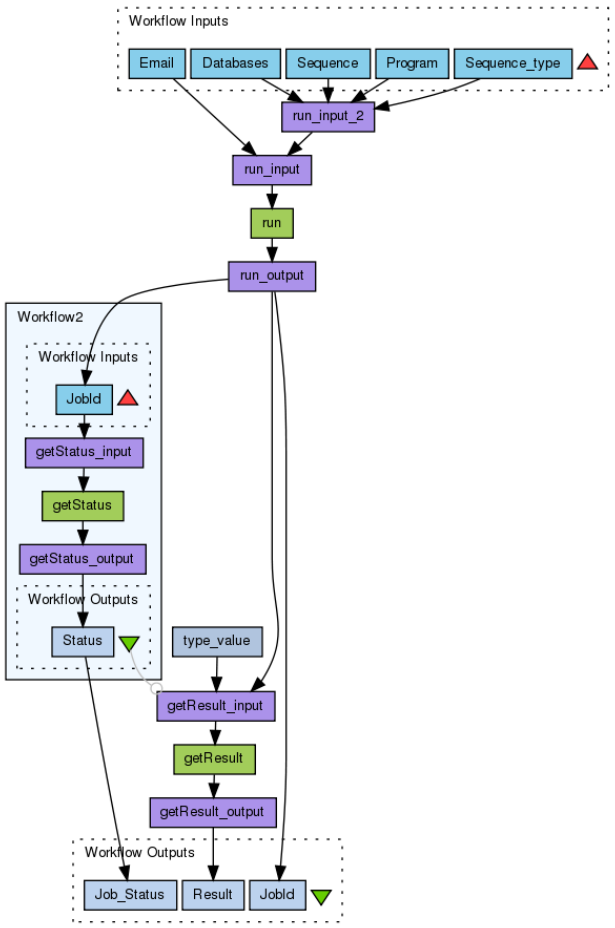


FIGURE 1. A sample scientific workflow from *Taverna 2* of *myExperiment* repository with the title "*NCBI BLAST (SOAP)*".

the exploration. Authors have developed a workflow similarity computation technique to promote the reuse and repurposing of certain workflows in *myExperiment*, where workflows are transferred into layer hierarchies, where a layer hierarchy reflects hierarchical relations between this workflow, its sub-workflows, and activities [5]. Similar approaches have been developed, which can be categorized into annotation-based [13], [14], structure-based [15], and data-driven [16]–[18] mechanisms. Besides, typical fragments can be retrieved from workflows to improve their sharing and reuse when possible [19], [20], and an efficient indexing mechanism has been developed in [21] to promote the search and reuse of service process fragments with various granularities. Generally, current techniques aim to study the reuse and repurposing of complete workflows or their fragments. As far as we know, composing fragments, which are contained in various workflows, has not been explored extensively. These techniques seldomly consider the functional relevance of activities in different workflows. In fact, this relevance has demonstrated its performance in [22], where *abstract* activities are discovered and represented as subprocesses, and thus, hierarchical process models,

rather than flat ones, can be elicited from event logs. In this setting, reusing and repurposing fragments that cross various workflows, while considering the activity relevance in various workflows, is a promising topic to be further explored.

To address this challenge, this article proposes a crossing workflows fragments discovery mechanism, where the activity relevance in workflows is considered. Our contributions are summarized as follows:

- The relevance of activities is quantified by the degree of semantic similarity between activities, which is calculated leveraging their names in string and descriptions in plain-text. Activities are clustered by adopting modularity-based community discovery clustering algorithm, and activities contained in a certain cluster are assumed to be functionality-relevant and thus can be represented by an *abstract* activity. Workflows are rewritten into their abstract format through replacing activities by their abstract counterparts.
- An abstract activity network model is constructed upon abstract workflows, where vertices correspond to abstract activities, and directed edges reflect invocation relations specified upon contiguous abstract activities. Given a novel requirement represented in terms of a directed graph, a sub-graph matching algorithm is adopted to generate candidate fragments composed by abstract activities, where these abstract activities may be contained in a single workflow, or may be located in various workflows. Candidate abstract activity fragments are instantiated by replacing abstract activities by suitable activities in a certain cluster, and these instantiated fragments are ranked according to their semantic and structural similarities with respect to the requirement specification.

Extensive experiments are conducted to evaluate the effectiveness and accuracy of this technique. The results show that this technique is efficient and accurate on ranking and recommending appropriate workflow fragments.

The rest of this article is organized as follows. Section II introduces relevant concepts. Section III presents the procedure for generating functionally-similar abstract activities. Section IV discovers and recommends fragments crossing scientific workflows. Section V presents experimental environment settings, and Section VI evaluates this technique with respect to the rivals. Section VII reviews and discusses related techniques, and finally, Section VIII concludes this article.

II. PRELIMINARIES

A. CONCEPT DEFINITION

Definition 1 (Scientific Workflow): A scientific workflow is a tuple $swf = (tl, dsc, ACT, LNK)$, where tl and dsc are the title and text description of swf, respectively. ACT refers to a set of activities in swf, and LNK represents a set of links that specifies data input and output relations between activities in ACT.

A sample scientific workflow is shown in Fig. 1, whose title is "*NCBI BLAST (SOAP)*". This workflow contains 11 activities which are implemented by Web/REST services.

Definition 2 (Activity): An activity is a tuple $act = (nm, dsc, tp)$, where nm is the name, dsc is the text description, and tp is the type, of act.

A sample activity is act "*getResult*" as shown in Fig. 1, whose tp is "*dataflow*". Generally, tp can be (i) *"wsdl"* and *"dataflow"*, which correspond to domain (or Web/REST) services, or (ii) *"beanshell"* and *"stringconstant"*, which correspond to shim services representing data transformation procedures between inputs and outputs of contiguous domain services.

B. SEMANTIC SIMILARITY COMPUTATION FOR ACTIVITIES

This section briefly introduces the semantic similarity computation for two activities $act_1 = (nm_1, dsc_1, tp_1)$ and $act_2 = (nm_2, dsc_2, tp_2)$, and this procedure has been developed in our previous work [23].

1) SEMANTIC SIMILARITY FOR ACTIVITY NAMES

An activity name can be decomposed into a sequence of words using word segmentation method [24], while regarding "_", *space*, and *uppercase* characters as divisional marks. For instance, $act_1.nm_1$ = "*common_pathway_describe*" is decomposed into a set WD_{aN1} = {*"common"*, *"pathway"*, *"describe"*}, and $act_2.nm_2$ = "*common_pathway_link*" derives WD_{aN2} = {*"common"*, *"pathway"*, *"link"*}. The minimum cost and maximum flow algorithm [25] is used to compute the distance (denoted *cost*) between WD_{aN1} and WD_{aN2}, and WordNet aims to calculate the semantic similarity for words. Therefore, semantic similarity $sim_{aN}()$ for $act_1.nm_1$ and $act_2.nm_2$ is computed as follows:

$$sim_{aN}(act_1.nm_1, act_2.nm_2) = 1 - \frac{cost}{max(SizOf(WD_{aN1}), SizOf(WD_{aN2}))} \quad (1)$$

where the function $SizOf(WD_{aN1}\ (or\ WD_{aN2}))$ returns the size of WD_{aN1} (or WD_{aN2}). Generally, $sim_{aN}()$ returns a value between 0 and 1, where 0 means totally different while 1 means the equivalent. For instance, the semantic similarity for $act_1.nm_1$ and $act_2.nm_2$ is computed as the value of 0.683.

2) SEMANTIC SIMILARITY FOR TEXT DESCRIPTION OF ACTIVITIES

Leveraging the word segmentation technique, the text description can be decomposed into a sequence of words, and the minimum cost and maximum flow algorithm and WordNet are adopted for computing the degree of similarity for a pair of text descriptions $sim_{aD}()$ accordingly. Generally, $sim_{aD}()$ returns a value between 0 and 1, where 0 means totally different while 1 means the equivalent. For instance, the semantic similarity for $act_1.dsc_1$ and $act_2.dsc_2$ is computed as the value of 0.808.

3) SEMANTIC SIMILARITY COMPUTATION FOR TWO ACTIVITIES

Leveraging the degrees of similarity for the name and text description of two activities act_1 and act_2, the semantic similarity for act_1 and act_2 is calculated as follows:

$$sim_{act}(act_1, act_2) = \alpha \times sim_{aN}(act_1.nm_1, act_2.nm_2) \\ (1-\alpha) \times sim_{aD}(act_1.dsc_1, act_2.dsc_2) \quad (2)$$

where the factor $\alpha \in [0, 1]$ reflects the importance of $sim_{aN}()$ with respect to $sim_{aD}()$. As explained in our previous work [23], the text description usually contains richer information than the name, and thus, can contribute more to the similarity computation of activities. Based on this observation we set α as 0.3 in our experiments. Note that α can be set to any value of appropriate when necessary. Generally, $sim_{act}(act_1, act_2)$ returns a value between 0 and 1. The larger the $sim_{act}()$ is, the more similar and relevant the activities act_1 and act_2 are. For instance, $sim_{act}(act_1, act_2)$ is approximately equal to 0.771.

Leveraging the semantic similarity between activities, a network can be constructed where vertices correspond to activities, and the weight upon edges specifies the distance between a certain pair of activities. Given the sample workflow as shown in Fig. 1 where the semantic similarity of *act* "*getResult*" and *act* "*getResult_output*" is 0.77, their distance is calculated as the value of 0.23 (i.e., 1 - 0.77). Activities with slight semantic correlations are far away from each other in the activity network. Thus, the edge between activities whose distance exceeds a threshold $thrd_{dis}$ should be pruned. The average path length (L) and clustering coefficient (C) of the network are employed as the factors to determine an appropriate value of $thrd_{dis}$, where (i) L is the average distance of all pairs of activities and reflects the degree of separation among activities, and (ii) C describes the degree of aggregation and noteworthy, such that a larger value of C specifies a more obvious community structure of the network. Since the semantic similarity ranges from 0 to 1, $thrd_{dis}$ starts at 1 and decreases at the gradient of 0.02 until reaches 0. This procedure ensures that an optimal $thrd_{dis}$ is not missed. L and C of each $thrd_{dis}$ are calculated respectively. In our case, $thrd_{dis} = 0.58$ corresponds to the optimal $L = 5.211$ and $C = 0.727$. There are 2798 valid activities in the *Taverna 2* repository of *myExperiment*. After removing (i) edges between activities whose distance exceed $thrd_{dis}$, and (ii) 258 isolated activities after removing edges, the network containing the remaining 2540 activities is constructed with an obvious community structure. Note that these 258 isolated activities should be post-processed in Section III-B.

III. ABSTRACT ACTIVITY GENERATION

This section aims to cluster activities in the activity network, and adopt abstract activities to represent clusters of functionality-similar activities.

A. MODULARITY-BASED ACTIVITY CLUSTERING MECHANISM

The edges between functionality-similar activities are relatively large in value, and they construct a community. In this setting, *Louvain* [26], [27], which is a modularity-based clustering algorithm, is applied to generate abstract activities. The density of links within communities as compared to links between communities can be measured leveraging modularity optimization, which is a scalar value between −1 and 1. The activity clustering procedure is presented as follows:

- *Step 1:* Assign each activity to an independent cluster initially.
- *Step 2:* Consider a neighboring activity j of a certain activity i and evaluate the gain of modularity ΔQ that would take place by removing i from its cluster and by placing it in the cluster of j. Note that ΔQ obtained by moving another activity i into a cluster CL can easily be computed by:

$$\Delta Q = \left[\frac{\sum_{in}+k_{i,in}}{2m} - \left(\frac{\sum_{tot}+k_i}{2m}\right)^2\right] \\ - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right] \quad (3)$$

  where $\sum_{in}$ is the sum of the distance inside CL, $\sum_{tot}$ is the sum of the distance incident to activities in CL, k_i is the sum of the distance incident to activity i, $k_{i,in}$ is the sum of the distance from i to activities in CL, and m is the sum of all distance in the network.
- *Step 3:* Record the neighbor with the largest ΔQ. If the maximum ΔQ is greater than 0, activity i is assigned to the cluster where the neighbor with the largest ΔQ is located.
- *Step 4:* Repeat step 2 & 3 until all activities assigned to clusters do not change any longer.
- *Step 5:* Compress the network and abstract all the activities in the same cluster into a new activity. The distances between the activities in the cluster are transformed into the distances of the rings of the new activity, and the distances between the clusters are transformed into the distances of the new activities;
- *Step 6:* Repeat step 1 until there are no more changes in the entire network and the maximum modularity Q is calculated as follows:

$$Q = \frac{1}{2m}\sum_{i,j}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta(c_i, c_j), \quad (4)$$

  where A_{ij} represents the distance between i and j, $k_i = \sum_j A_{ij}$ is the sum of the distances attached to vertex i, c_i is the cluster to which vertex i is assigned, δ-function $\delta(u, v)$ is 1 if u = v, and is 0 otherwise. m is set to $\frac{1}{2}\sum_{i,j} A_{ij}$.

Generally, the more significant community structure can be constructed with the increasing of Q, the community structure is reasonable when Q exceeds 0.3 [26], and optimal clusters can be constructed accordingly. For instance, 119 clusters

Algorithm 1 Isolated Activity Clustering

Require:
- CLS: a set of clusters generated by Section III-A
- Sim_{tl}: the values of similarity between activities
- NL_1: a list of activities in the activity network
- NL_2: a list of activities exit in the *myExperiment*, *Taverna 2* repository

Ensure:
- ACT_{abs}: a set of clusters covering all activities

1: $OutNd \leftarrow NL_2$ - NL_1
2: **for** $otn \in OutNd$ **do**
3: $thrd_{tmp} \leftarrow 0$, $clu^{otn}_{max} \leftarrow 0$, $sum_{tmp} \leftarrow 0$, $cnt \leftarrow 0$
4: **for** each $atn \in NL_1$ **do**
5: $sum_{tmp} \leftarrow sum_{tmp} + Sim_{(otn,atn)}$
6: $cnt \leftarrow cnt + 1$
7: **end for**
8: $thrd_{tmp} \leftarrow sum_{tmp} \div cnt$
9: $CLS_{tmp} = \emptyset$
10: **for** each $atn \in NL_1$ **do**
11: **if** $Sim_{(otn,atn)} > thrd_{tmp}$ **then**
12: $CLS_{tmp} \leftarrow CLS_{tmp} \cup \{clu_{atn}, Sim_{(otn,atn)}\}$
13: **end if**
14: **end for**
15: $clu^{otn}_{max} \leftarrow AvgMax\ (CLS_{tmp})$
16: $ACT_{abs} \leftarrow ACT_{abs} \cup clu^{otn}_{max}$
17: **end for**
18: $ACT_{abs} \leftarrow$ a set with clusters covering all activities in dataset

with the maximum Q 0.73 are generated as the most appropriate community structure.

B. ISOLATED ACTIVITY CLUSTERING

After *Louvain* is applied in the activity network, the type and quantity of clusters are determined, where CLS denotes the set of generated clusters. In particular, each cluster contains a set of functionality-similar activities. However, original activities not contained by the network (i.e., 258 isolated activities) are not assigned to corresponding clusters. Therefore, the isolated activity clustering procedure is presented by Algorithm 1, where the NL_1 is the activity set of the activity network while the NL_2 indicates all original activities in the repository. Similarity values between activities are stored in the Sim_{tl}.

Leveraging NL_1 and NL_2, make sure what the isolated activities (denoted $OutNd$) are (line 1). Given an otn from the $OutNd$ (line 2), the clustering procedure about this otn is carried out (lines 2-17). Note that otn have similarity values with any activities of the activity network. If otn directly selects the cluster with the largest similarity as its cluster, the size of clusters will affect the similarity calculation, resulting in an unreasonable selection strategy. In this setting, we set a similarity range (denoted $thrd_{tmp}$) for this otn to overcome the above problem (lines 4-7). $thrd_{tmp}$ is determined as the average similarity value, which is obtained by calculating the similarity sum (denoted sum_{tmp}) and count (denoted cnt) of otn and activities in the network (line 8). If the similarity value between an activity atn from NL_1 and otn is large in comparison with the $thrd_{tmp}$ (line 11), this value and cluster of atn are saved as a whole to a collection (denoted CLS_{tmp}) (line 12). By the function $AvgMax()$, CLS_{tmp} is analyzed comprehensively and the average similarity for each cluster is obtained within $thrd_{tmp}$ (line 15). Finally, the otn is assigned to the cluster with the largest average similarity. Each isolated activity is addressed in the same way until they all match similar categories (line 18).

C. ABSTRACT ACTIVITY SPECIFICATION

A cluster contains a set of functionally-similar activities, which can be represented as an *abstract* activity from the functional perspective.

Definition 3 (Abstract Activity): An abstract activity is a tuple $aact_{abs} = (INFO_{abs}, ACT_{abs})$, where $INFO_{abs}$ is the keyword information of the $aact_{abs}$, and ACT_{abs} is a set of activities in the $aact_{abs}$.

$INFO_{abs}$ expresses the representative semantic information to promote the fragment mining afterwards. Generally, it should be defined as a sequence of one or more words. In our context, the information of each abstract activity can be obtained in condensed form the essential content. Therefore, the keyword extraction technology, called *A Rapid Automatic Keyword Extraction* (*RAKE*) algorithm [28], is adopted. Specifically, *RAKE* is an unsupervised, domain- and language-independent method for extracting keywords from individual documents. All the names of activities pertaining to each $aact_{abs}$ are integrated into a document *AasInfo*, where commas are selected as the delimiter between names. For instance, $AasInfo_{114}$ = "*datum cluster protocol file transfer to transport secure, result metadata protocol file transfer and to transport local, result metadata protocol file transfer and to transport secure local*" is a document content of the $aact^{114}_{abs}$, whose ACT^{114}_{abs} consists of act_{2837}, act_{2840} and act_{2869}.

Based on our observation, keywords about abstract activities should rarely contain standard punctuation, stop words (i.e., *and* and *or*) or other words with minimal lexical meaning, where such words are too frequent and broadly used to aid users in their analyses. Therefore, the procedure of extracting keywords is illustrated considering the above problem as follows:

- *Candidate keywords:* The input parameters consist of a list of stop words (denoted *stpList*), a set of phrase delimiters, and a set of word delimiters. First of all, the document text is split into an array of words by the specified word delimiters, such as space character. For instance, $AasInfo_{114}$ is divided into the array $Array_{114}$ = (*"datum - cluster - protocol-file-to-transport - secure -, - result - metadata - protocol - file - transfer - and - to - transport - local -, - result - medatata - protocol - file - transfer-and-to - transport - secure - local"*), where *" - "* character is the symbol for dividing words in $Array_{114}$.

TABLE 1. The word co-occurrence table for content words in the $NodeInfo_{114}$.

WORD	datum	cluster	protocol	file	transfer	transport	secure	result	metadata	local
datum	1	1	1	1	1	0	0	0	0	0
cluster	1	1	1	1	1	0	0	0	0	0
protocol	1	1	3	3	3	0	0	2	2	0
file	1	1	3	3	3	0	0	2	2	0
transfer	1	1	3	3	3	0	0	2	2	0
transport	0	0	0	0	0	3	2	0	0	2
secure	0	0	0	0	0	2	2	0	0	1
result	0	0	2	2	2	0	0	2	2	0
metadata	0	0	1	1	1	0	0	1	1	0
local	0	0	0	0	0	2	1	0	0	2

TABLE 2. Word scores calculated from the word co-occurrence table.

WORD	datum	cluster	protocol	file	transfer	transport	secure	result	metadata	local
$deg(w)$	5	5	15	15	15	7	5	10	5	5
$freq(w)$	1	1	3	3	3	3	2	2	1	2
$deg(w)/\ freq(w)$	5	5	5	5	5	7/3	5/2	5	5	5/2

Therefore, $AasInfo_{114}$ is split into sequences of contiguous words at phrase delimiters and stop word positions. According to the $AasInfo_{114}$, commas are naturally identified as phrase delimiters. In this paper, the *stpList* from [29] is applied, which contains common words like *and*, *to* and so on in our text. Words within a sequence are assigned the same position in the text and together are considered a candidate keyword. For instance, the array $Array_{114}$ becomes $CanKey_{114}$ = (*"datum cluster protocol file - transport secure - result metadata protocol file transfer - transport local - result medatata protocol file transfer - transport secure local"*). The candidate keyword "*transport secure*" begins after the stop word *to* and ends with a comma. The following word "*result*" begins the next candidate keyword "*result metadata protocol file transfor*".

- *Keyword scores:* It makes sense to consider co-occurrences of words within candidate keywords, which are not identified by the application of an arbitrarily sized sliding window. For instance, the "*local*" word exists in "*transport local*" and "*transport secure local*" respectively, therefore, $(local, transport) = 2$ and $(local, secure) = 1$. Meanwhile, we also need to consider the number of word itself that appear, thus, $(local, local) = 2$. After every candidate keyword is identified and the table of word co-occurrences (shown in Table 1) is complete, a score is calculated for each candidate keyword. Leveraging three metrics: (i) word frequency ($freq(w)$), (ii) word degree ($deg(w)$), (iii) the ratio of degree to frequency ($deg(w)/freq(w)$), word scores are evaluated and calculated in Table 2. $deg(w)$ favors words that occur often and in longer candidate keywords. Words that occur frequently are favored by $freq(w)$ regardless of the number of words with which they co-occur. Words that predominantly occur in longer candidate keywords are favored by $deg(w)/freq(w)$. The score for each candidate keyword is computed in the sum of its member word scores ($deg(w)/freq(w)$). As a result, scores of candidate keywords in $CanKey_{114}$ are (("*datum cluster protocol file transfer*", 25.0), ("*result metadata protocol file transfer*", 25.0), ("*transport secure local*", 7.334), ("*transport local*", 4.833)).
- *Extracted keywords:* The top T scoring candidates are selected as the keyword information after candidate keywords are scored. Mihalcea and Tarau [30] argues that T is one-third the number of candidate keywords is appropriate. Therefore, "*datum cluster protocol file transfer*" becomes the $INFO_{abs}^{114}$ of the $aact_{abs}^{114}$ when the $CanKey_{114}$ of the sample text contains 4 candidate keywords.

Because a few candidate keywords may acquire the same score, the selection of keywords is important. For instance, "*datum cluster protocol file transfer*" and "*result metadata protocol file transfer*" scores are 25. Through comparing the similarities between these candidate keywords, it is found that the content and semantics of them are similar, Therefore, no matter which one is chosen will be reasonable when only top one-third can be taken. After *RAKE* algorithm has been adopted, $INFO_{abs}$ of each $aact_{abs}$ is available, which lays the foundation for further fragment mining. Besides, the structural information contributes to reflecting the relationship of (abstract) activities. Scientific workflows are rewritten into their abstract format leveraging abstract activities and an abstract activity network model is generated to accommodate collaboration patterns.

Definition 4 (Abstract Activity Network Model): An abstract activity network model *AbsN* is a tuple $(AACT_{abs}, LNK)$, where:

- $AACT_{abs}$ is a set of abstract activities.
- *LNK* is a set of links that connect abstract activities contained in $AACT_{abs}$.

Such an *AbsN* is constructed, where the vertices correspond to abstract activities while the directed link reflects the invocation relations specified upon contiguous abstract

activities. For instance, there is a link between $aact_{abs}^{50}$ and $aact_{abs}^{53}$, which means that they can collaborate on certain function. Specifically, the procedure of constructing the *AbsN* is illustrated as follows:

- *Step 1:* Edge weights between abstract activities in *AbsN* is set to 0 by default.
- *Step 2:* Each workflow is expressed as *swf*, whose all invoking relations are mapped to edges between abstract activities and set to 1. In particular, a query function is required to achieve the corresponding abstract activity by an original activity.
- *Step 3:* Iterate over the edges between abstract activities. If its value is 1, the edge information is entered into the *AbsN*. Until internal structures of all scientific workflows are processed.

It is noteworthy that keeping the self-connection of abstract activities is essential in *AbsN*. In real life, a requirement may be accomplished by one or more functions, and services with similar semantics generally collaborate to complete the whole or part of a function. The self-cooperative pattern is formed, which is in line with practical situation. Correspondingly, *AbsN* is provided with abundant abstract activity fragments, which is flexibly to deal with various requirements of users.

IV. WORKFLOW FRAGMENT RECOMMENDATION

Given an user's requirement, fragments can be generated and ranked according to our recommended strategy. Firstly, abstract activity fragments are mined from *AbsN* considering structural and semantic discovery, which effectively reduce the scope of the fragment search. Next, leveraging the candidate abstract activity fragments, the instantiated fragments crossing workflows are restored to recommend with respect to the requirement specification.

A. ABSTRACT ACTIVITY FRAGMENT RECOMMENDATION

This section aims to generate abstract activity fragments in an abstract level. We propose the abstract activity fragment discovery procedure from two aspects of structure and semantics.

1) STRUCTURALLY SIMILAR ABSTRACT ACTIVITY FRAGMENT DISCOVERY

Assuming the user is an expert in this domain, whose requirement (denoted *User*) is considered to be professional and accurate. In order to discover optimal or suboptimal abstract activity fragments from the *AbsN* for the *User*, its abstract structure (denoted $User_{abs}$) needs to be transformed. Consequently, the above question can be understood as the graph or sub-graph matching. In this setting, the most representative graph isomorphism algorithm, called *VF*2 [31], is applied in our technique.

To put it simply, the $User_{abs}$ and *AbsN* are regarded as two graphs $G_1 = (N_1, B_1)$ and $G_2 = (N_2, B_2)$, where N_1 and N_2 are abstract activity sets and B_1 and B_2 are edge sets. In addition, G_1 and G_2 are called target graph and query graph respectively. A matching process consists in the determination of a mapping M which associates abstract activities of G_1 with G_2. A mapping $M \subset N_1 \times N_2$ is said to be an isomorphism iff M is a bijective function that preserves the branch structure of the two graphs. The solutions of *VF*2 algorithm are obtained by computing some the possible partial solutions or exact matching. In this case, some abstract activity fragments that are structurally similar to $User_{abs}$ can be excavated from *AbsN*. For instance, the *User* is $\big((act_{26}, act_{27}, act_{28}), (act_{27} \rightarrow act_{28}, act_{28} \rightarrow act_{26})\big)$, whose $User_{abs} = \big((aact_{abs}^{14}, aact_{abs}^{17}), (aact_{abs}^{17} \rightarrow aact_{abs}^{14}, aact_{abs}^{14} \rightarrow aact_{abs}^{14})\big)$ is a target graph considering replacing activities with abstract activities. Generally, the mapping M is expressed as the set of pairs $(n - m)$ (with n $\in User_{abs}.N_1$ and m $\in AbsN.N_2$) representing the mapping of an abstract activity n in the $User_{abs}$ with an abstract activity m in the *AbsN*. Thus, after *VF*2 algorithm is applied, some solutions, such as $(N1.aact_{abs}^{14} - N2.aact_{abs}^{14}, N1.aact_{abs}^{17} - N2.aact_{abs}^{17})$, $(N1.aact_{abs}^{14} - N2.aact_{abs}^{14}, N1.aact_{abs}^{17} - N2.aact_{abs}^{70})$, $(N1.aact_{abs}^{14} - N2.aact_{abs}^{29}, N1.aact_{abs}^{17} - N2.aact_{abs}^{17})$, $(N1.aact_{abs}^{14} - N2.aact_{abs}^{14}, N1.aact_{abs}^{17} - N2.aact_{abs}^{16})$ and so on, are found to response the $User_{abs}$.

Specifically, the semantic matching about each abstract activity isn't considered in the application of *VF*2. If the semantic matching is added, only fragments of the semantic perfect matching rather than approximate matching are found. It is a fact that abstract activity fragments with the same structure and similar semantics are also thought to be what the *User* expects. Therefore, the semantic discovery is executed in the next procedure.

2) SEMANTICALLY SIMILAR ABSTRACT ACTIVITY FRAGMENT DISCOVERY

Leveraging the discovered solutions, this section aims to select semantically-similar abstract activity fragments from them. Since the $INFO_{abs}$ for each abstract activity has been generated, the average semantic similarity value for each solution is calculated by Algorithm 2.

Given a solution (denoted *sol*) from *SOLs* whose pair is expressed as $(n - m)$ (lines 1-3), n is the abstract activity (denoted $aact_{user}$) in the *User* while m is the matching abstract activity (denoted $aact_{AbsN}$) from the *AbsN* (lines 4-5). In particular, the function *FindAas*() is implemented to query the semantic information for $aact_{user}$ and $aact_{AbsN}$ (lines 6-7). Through Formula 1, their semantic information is obtained and denoted as *sim* (line 8). Each pair about *sol* will be processed in the same way and achieved similarities sum up to the variable SIM_{sum} (line 9). Specially, some solutions may be the partial structural fragments, whose number of abstract activities is no more than $User_{abs}$ (denoted Num_{usrN}). Such solutions are suboptimal compared to exact matching, which can partially meet the user's needs. Therefore, the average semantic similarity should be acquired by SIM_{sum} and Num_{usrN} (line 11), and added to results (line 12).

Algorithm 2 Average Semantic Similarity of Solutions

Require:
- AAS: a set of semantic information of abstract activities
- $SOLs$: a set of structurally-similar solutions discovered in the $AbsN$

Ensure:
- SIM_{sols}: a set of average semantic similarity values of solutions

for each $sol \in SOLs$ **do**
  $SIM_{sum} \leftarrow 0$, $SIM_{avg} \leftarrow 0$
  for each $(n - m) \in sol$ **do**
    $aact_{usr} \leftarrow$ n
    $aact_{AbsN} \leftarrow$ m
    $info_{usr} \leftarrow FindAas\ (AAS, aact_{usr})$
    $info_{AbsN} \leftarrow FindAas\ (AAS, aact_{AbsN})$
    $sim \leftarrow SIM\,(info_{usr}, info_{AbsN})$ through Formula 1
    $SIM_{sum} \leftarrow SIM_{sum} + sim$
  end for
  $SIM_{avg} \leftarrow SIM_{sum} \div Num_{usrN}$
  $SIM_{sols} \leftarrow SIM_{sols} \cup SIM_{avg}$
end for
$SIM_{sols} \leftarrow$ a set of similarity values of solutions, which represent the average semantic similarity between $User_{abs}$ and $SOLs$

Consequently, similarity values of candidate solutions are calculated through Algorithm 2 accordingly. For instance, SIM_{sols} is (1, 0.62601, 0.60925, 0.61410, 0.608485, ...) and these values are sorted as S = (1, 0.62601, 0.61410, 0.60925, 0.608485, ...) in descending order. Therefore, the solutions, whose similarity values are within the top k% (e.g., 10%), are selected as the set of appropriate candidate abstract activity fragments. Note that the above work has ensured that selected candidate abstract activity fragments are structurally and semantically similar or superior to $User_{abs}$.

B. INSTANTIATED FRAGMENT RECOMMENDATION

In order to identify candidate abstract activity fragments, we intend to restore the appropriate instantiated fragments. In this phase, instantiations are obtained and recommended to scientists afterwards. This procedure is presented as follows:

- *Step 1:* Keep structures of candidate abstract activity fragments unchanged. For instance, its structure is $(aact^{17}_{abs} \rightarrow aact^{14}_{abs}, aact^{14}_{abs} \rightarrow aact^{14}_{abs})$, where $aact^{17}_{abs}$ points to $aact^{14}_{abs}$ and $aact^{14}_{abs}$ has the self-connecting structure, and remains unchanged.
- *Step 2:* Abstract activities are replaced to the corresponding appropriate activities in clusters. Note that instantiated fragments are made up of original activities. For instance, $aact^{17}_{abs}$ contains of act_{27}, act_{34}, act_{47}, act_{49}, act_{50} and so on while $aact^{14}_{abs}$ is made up of act_{11}, act_{25}, act_{26}, act_{28}, act_{35} and so on. Therefore, instantiations, which refer to $\{(act_{27} \rightarrow act_{25}, act_{25} \rightarrow act_{25}), (act_{27} \rightarrow act_{28}, act_{28} \rightarrow act_{26}), (act_{27} \rightarrow act_{28}, act_{28} \rightarrow act_{28}), \ldots\}$, are restored.
- *Step 3:* Instantiated fragment sets are ranked and recommended. For comprehensive consideration of requirement-related fragments, two evaluation parameters about structure and semantics are adopted: the path length ratio (PL) and the average semantic similarity (AS) (refer to Algorithm 2). PL is the path length ratio to complete the fragment of user's requirement, which is the percentage of path lengths that instantiations has completed in $User$. For example, in an instantiated fragment there are 4 edges that are paths in the user's requirement (assuming the requirement has a total of 6 path lengths), PL is equal to $\frac{4}{6}$. In addition, AS represents the average semantic similarity between an instantiation and $User$, which is analogously calculated by Algorithm 2. The difference is that the semantic similarity calculation between abstract activities is replaced to the calculation between names of activities.

Generally speaking, similarity results SIM_{ins} of instantiated fragments are calculated by PL and AS as shown in Formula 5.

$$SIM_{ins} = (1 - \xi) \times PL + \xi \times AS \tag{5}$$

where $\xi \in [0, 1]$ reflects the importance of AS. Generally, instantiations of high semantic similarity should be more suitable to be recommended. Therefore, AS is assigned a higher weight than PL. For instance, assuming ξ is set to 0.6, similarity values of instantiations $\{(act_{27} \rightarrow act_{25}, act_{25} \rightarrow act_{25}), (act_{27} \rightarrow act_{28}, act_{28} \rightarrow act_{26}), (act_{27} \rightarrow act_{28}, act_{28} \rightarrow act_{28}), \ldots\}$ are calculated as {0.9250, 1, 0.9625, ...}. When all similarity values have been calculated, suitable instantiations (or workflow fragments) should be ranked. Consequently, the top $tp\%$ of instantiations are selected and the scientist will examine and determine which is the most appropriate with respect to his requirement.

V. EXPERIMENTAL SETTING

The prototype has been implemented in Java and C++ programs. Scientific workflows in the category of *Taverna 2* of the *myExperiment* online repository are collected for the experimental evaluation. There are 1571 scientific workflows in this category till to February 2, 2018. Since there are 795 workflows whose title or description is empty, which are assumed to be invalid, 776 workflows are remaining as valid. The distributions of the number of activities, and the number of edges, for these valid workflows are shown in Fig. 2. Generally, the majority of workflows is relatively small in size (containing less than 17 activities), and relatively complex in structure (containing less than 17 edges). Experiments are conducted on a desktop with an Intel(R) Core(TM) i5-6500 processor, a 6.00GB memory, and a 64-bit Windows 7 system.

A. DATA CLEANING AND EXPERIMENTAL SETUP

Since scientific workflows and activities may lack specifications, which may lead to an inaccuracy of similarity

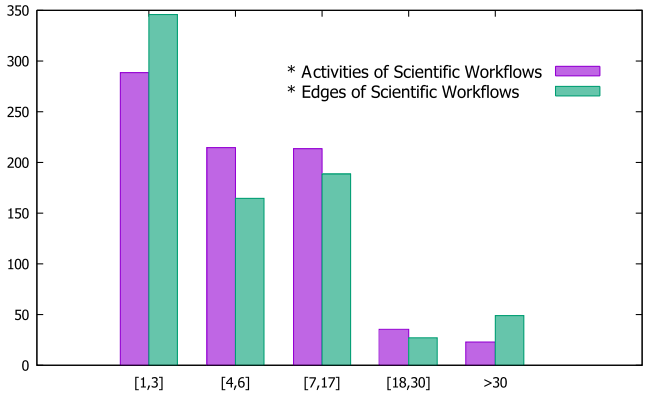


FIGURE 2. Histogram distributions for the number of activities and edges contained in scientific workflows. The symbol "** Activities (or Edges) of Scientific Workflows*" represents the numbers of activities (or edges).

computation, and thus, give rise to improper activity clustering and fragment mining, the dataset have been cleaned before experiments are conducted. Specifically,

- The title or description of 795 scientific workflows are empty, which are assumed as ones of incomplete information, and thus, are removed from the experimental dataset. Besides, activities, whose name or text description is empty, are removed from scientific workflows. In the removal of activities, we need to reconnect the graphs of workflows. The precursor activity and subsequent activity of the removed activity connect together in order to become a new graph for the purpose of retaining the partial functional fragment structure.
- Activities belonging to the shim service are uninterested for exploring the functionality of a workflow. Certainly, we should filter out activities of the shim service type. As aforementioned, the semantic similarity between words is computed by WordNet. However, improper words which are not recognized by WordNet are contained in the name or text description. We have handled them case-by-case as follows:
  1) There are 353 abbreviations which are used in the bioinformatics domain, but cannot be recognized by WordNet. An example is *kegg*, which means "*Kyoto Encyclopedia of Genes and Genomes*" after searching on the web. A conversion table is manually established, which aims to transform these abbreviations and their full descriptions. If a full description can't be found through browsing the web, such an abbreviation, like "*gsmda*", it is assumed to be dissimilar to any other word, and such abbreviations is removed.
  2) Regarding the abbreviation which is just a part of a word, we can convert it to the word of full by a table. An example is *occ*, which corresponds to *occurrence*.
  3) Words, whose meaning is not clear and information is incomplete, are complemented according to context and web network. An example is *btit*, which means "*gene database*" about the conversion of gene serial number in the bioinformatics domain.
  4) Pronoun is not recognized by WordNet, like *your*, and *my*. They are removed from their name or text description.
  5) Also, adjective and adverb words, like *chemical*, are not well recognized by WordNet. They are changed to the noun form *chemistry* which is suitable to be used for the similarity computation.
  6) When two or several words are joint without delimiters, they are separated manually. An example is *webservice*, which is divided into two words *web* and *service*.

Without loss of generality, scientists are assumed to have basic knowledge about the workflows published in the repository. When the requirement for a novel experiment is specified, a workflow is given for implementing this requirement. These requirements can be completely, or at least partially, satisfied by a single or multiple scientific workflows. Therefore, in order to ensure the rationality of the workflow as a requirement, we select workflow samples from the repository and modify them with certain principles to achieve the purpose of cross-workflow requirements. For facilitating the universality of this technique, the selection of workflow samples need to cover various complexity requirements, which also further verifies that some requirements may be satisfied by one workflow as a whole. According to distributions for the number of activities and edges of scientific workflows in Fig. 2, we analyze comprehensively and divide three groups $\{(1, 6), (7, 17), (18, 313)\}$ in terms of the number of activities. The proportions of the number of workflows that fall into these intervals are calculated as 65.8%, 24.4% and 9.8%. Suppose we select 20 samples from the repository, then numbers of samples respectively taken from these three intervals are calculated as 13, 5, and 2 by proportional distribution. Meanwhile, sample workflows are chosen evenly from datatset in our experiment.

B. FRAGMENT RECOMMENDATION WHEN A REQUIREMENT IS COMPLETELY SATISFIED BY A SINGLE SCIENTIFIC WORKFLOW

We first perform experiments when requirements can be is completely accomplished by a single scientific workflow. In this setting, sample workflows are selected and remained as they are from dataset. For a scientist's requirement, his most expected result obtained from this inquiry is an exact matched workflow fragment compared with his requirement. In fact, it is of core importance to identify appropriate workflow fragments, which are the most similar to the sample workflow of the requirement. Experiments have conducted for 20 sample workflows. As the result, 20 experiments return the *right* recommendation that contains sample workflows, which are ranked in the first place in terms of similarity SIM_{ins}. This means that our technique is appropriate when requirements are completely satisfied by a single scientific workflow. For instance, given the target graph of a sample workflow $swf_{user96} = \big((act_{184}, act_{185}, act_{186}),$

$(act_{184} \rightarrow act_{185}, act_{186} \rightarrow act_{184}, act_{186} \rightarrow act_{185})\big)$, workflow fragments are diagnosed and recommended as follows:

- *instantiation 1* $= \big((act_{184}, act_{185}, act_{186}), (act_{184} \rightarrow act_{185}, act_{186} \rightarrow act_{184}, act_{186} \rightarrow act_{185})\big)$ $(Sim_{total} = 1.0)$,
- *instantiation 2* $= \big((act_{184}, act_{185}, act_{381}), (act_{184} \rightarrow act_{185}, act_{381} \rightarrow act_{184}, act_{381} \rightarrow act_{185})\big)$ $(Sim_{total} = 0.84043)$,
- *instantiation 3* $= \big((act_{184}, act_{185}, act_{439}), (act_{184} \rightarrow act_{185}, act_{439} \rightarrow act_{184}, act_{439} \rightarrow act_{185})\big)$ $(Sim_{total} = 0.84043)$,
- *instantiation 4* $= \big((act_{184}, act_{185}, act_{397}), (act_{184} \rightarrow act_{185}, act_{397} \rightarrow act_{184}, act_{397} \rightarrow act_{185})\big)$ $(Sim_{total} = 0.840069)$,
- *instantiation 5* $= \big((act_{184}, act_{185}, act_{2679}), (act_{184} \rightarrow act_{185}, act_{2679} \rightarrow act_{184}, act_{2679} \rightarrow act_{185})\big)$ $(Sim_{total} = 0.839718)$.

C. FRAGMENT RECOMMENDATION WHEN A REQUIREMENT IS PARTIALLY SATISFIED BY MULTIPLE SCIENTIFIC WORKFLOWS

Workflows representing the requirement of scientists may be hardly the same as sample workflows. Generally, activities in multiple scientific workflows collaborate together to accomplish new requirements. However, there are some difference between multiple workflows, which can't be combined blindly. In this setting, our strategy is to modify the sample workflow according to certain principles, so that the obtained cross-workflow requirements are understandable and reasonable. Sample workflows with minor or major changes are instructed to conduct our experiment, when certain principles of operations including *insertion*, *deletion*, and *replacement* are applied:

- *Insertion:* Workflow fragments and activities that aren't included in a sample workflow swf_{sampl} are inserted as part of a testing workflow swf_{tst}. If some activities belong to abstract activities of swf_{sampl}, so swf_{tst} is usually more similar to swf_{sampl}, which means the kind of activities and connection structure in the swf_{tst} are close to swf_{sampl}. Thus, the swf_{tst} can be considered to be an upcoming cross-workflow requirement proposed by scientists. An example is the activity act_{i1} = "*runModel_input*" in the workflow *swf.tl* = "*Simple search*" as shown in Fig. 4.
- *Deletion:* Fragments and activities are deleted from a sample workflow swf_{sampl}, which can largely guide the development of novel experiments. An example is the activity act_2, act_5 and act_{11} as shown in Fig. 4.
- *Replacement:* Fragments and activities in a sample workflow swf_{sampl} are replaced by the others which are not specified in swf_{sampl}. Similar to the case of *insertion*, if the activities to be replaced are represented by activities in the same clusters, testing workflow is usually more similar to swf_{sampl}. An example is the activity act_8 = "*getResult_output*" which is to be replaced by act_{r8} = "*getJobState_output*" from another workflow, as shown in Fig. 3.

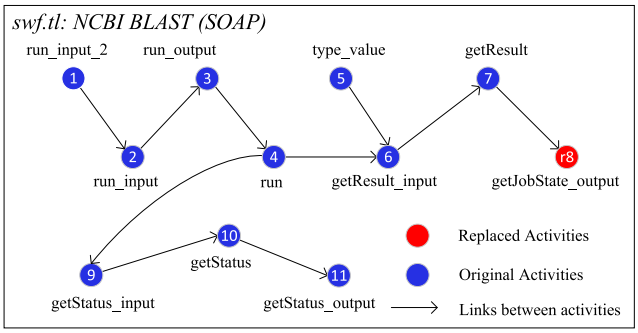


FIGURE 3. A testing workflow swf_{tst3} representing the user's requirement, which is developed from Fig. 1. Minor change is made by replacing act_8, i.e., act_{r8}, which belongs to the activity in $aact^{14}_{abs}$ that is the cluster that act_8 is in.

Experiments are conducted for the case of minor or major changes applied upon sample workflows.

1) MINOR CHANGES UPON SAMPLE WORKFLOWS

In this case, no more than 10% of fragments and activities are changed through the operation of insertion, deletion and replacement in a sample workflow swf_{sampl}. An example of a testing workflow graph is shown in Fig. 3, where a activity act_8 is replaced by the activity act_{r8} from the workflow *swf.tl* = "*Job_query*". Because act_8 named "*getResult_output*" and act_{r8} named "*getJobState_output*" are in the same category, demands for two workflow fragments are considered to be similar. Therefore, we can better use existing workflows to make cross-workflow fragment recommendations.

Regarding the 20 testing workflows devived from sample workflows, the results of workflow fragments recommendation for all 20 testing workflows are correct and the optimal solution, which is the same as the user's requirement, is recommended in the first place. An example is for $swf_{smpl15} = \big((act_{26}, act_{27}, act_{28}), (act_{27} \rightarrow act_{28}, act_{28} \rightarrow act_{26})\big)$, which the activity act_{27} is replaced by the activity act_{47} that belongs to the cluster where act_{27} is located, the most appropriate fragment is examined as the target solution, and fragments are diagnosed and recommended are ranked as follows:

- *instantiation 1* $= \big((act_{26}, act_{47}, act_{28}), (act_{47} \rightarrow act_{28}, act_{28} \rightarrow act_{26})\big)$ $(Sim_{total} = 1.0)$,
- *instantiation 2* $= \big((act_{25}, act_{47}, act_{28}), (act_{47} \rightarrow act_{28}, act_{28} \rightarrow act_{25})\big)$ $(Sim_{total} = 1.0)$,
- *instantiation 3* $= \big((act_{47}, act_{28}), (act_{47} \rightarrow act_{28}, act_{28} \rightarrow act_{28})\big)$ $(Sim_{total} = 0.9375)$,
- *instantiation 4* $= \big((act_{26}, act_{49}, act_{28}), (act_{49} \rightarrow act_{28}, act_{28} \rightarrow act_{26})\big)$ $(Sim_{total} = 0.9370)$,
- *instantiation 5* $= \big((act_{25}, act_{49}, act_{28}), (act_{49} \rightarrow act_{28}, act_{28} \rightarrow act_{25})\big)$ $(Sim_{total} = 0.9370)$.

The results show that our technology is appropriate and can return correct and reasonable results when minor changes upon sample workflows. what's more, accurate results are

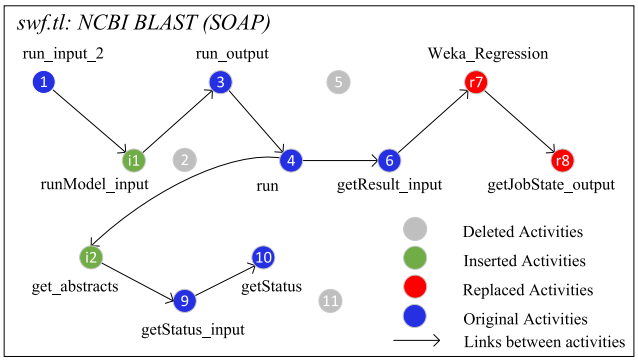


FIGURE 4. A testing workflow swf_{tst6} representing the user's requirement, which is developed from Fig. 1. Major change is made by inserting 2 activities, deleting 3 activities and replacing 2 activities.

recommended for minor modifications in complex cross-workflow requirements.

2) MAJOR CHANGES UPON SAMPLE WORKFLOWS

In this case, over 60% of fragments and activities are changed through the operation of insertion, deletion and replacement in a sample workflow swf_{sampl}. Consequently, testing workflows are actually novel fragments that are composed by multiple different scientific workflows. An example of test workflow is shown in Fig. 4. act_{i1} in the $aact^{25}_{abs}$ from the swf_{475} and act_{i2} in the $aact^{4}_{abs}$ from the swf_{562} are inserted into Fig. 4, and act_2, act_5 and act_{11} are deleted while act_7 and act_8 are replaced as act_{r7} and act_{r8}, respectively. Note that act_{r7} in the $aact^{43}_{abs}$ from the swf_{34} isn't the activity in cluster of act_7 while act_{r8} is the activity in same cluster of act_8.

Regarding the 20 testing workflows derived from sample workflows when major changes are applied, the results of recommendation for 19 testing workflows are correct. That shows that our technology is also applicable for large modified cross-workflow requirements. Even if there are no absolutely accurate instantiations, the most similar set will also be recommended to the user by our technique.

Note that 1 experiment returns similar results including not the exact instantiation as his requirement. This testing workflow swf_{tst19} is made up of workflows $swf_{70}.tl =$ "*G-language GenomeAnalysisEnvironment*" and $swf_{664}.tl =$ "*Get Gene Ids for Human*". Consequently, 21 edges are included in swf_{tst19}, are changed through the operation of insertion, deletion and replacement. Similarity values of the recommended workflow fragments are listed as follows:

- *instantiation 1* = ($Sim_{total} = 0.9825$),
- *instantiation 2* = ($Sim_{total} = 0.9801$),
- *instantiation 3* = ($Sim_{total} = 0.9766$),
- *instantiation 4* = ($Sim_{total} = 0.9735$),
- *instantiation 5* = ($Sim_{total} = 0.9688$).

Note that the size of instantiations is relatively large, so the specific graph structure isn't provided when enumerating the examples. We analyze the matching process and find that the self-connecting structure of $aact^{15}_{abs}$, which is contained in the abstract structure of the user requirement, was not included in candidate abstract activity fragments in Sec. IV-A. Through our analysis, this kind of structure isn't provided in *AbsN*. In other words, activities in $aact^{15}_{abs}$ don't exist any invoking relationship. However, relatively similar abstract activity fragments and instantiations have been recommended to the user for selection, and their functions performed are closer to the user's request. In particular, as data-intensive scientific processes increases, structures between abstract activities in *AbsN* will become more abundant and comprehensive to response to any novel scientists' requests.

VI. EXPERIMENTAL EVALUATION

The model assessment strategy is taken when experiments are conducted, and the performance metrics and results are presented as follows.

A. PERFORMANCE METRICS

We adopt commonly used metrics, namely Mean Absolute Error (*MAE*) and Root Mean Squared Error (*RMSE*), to measure the recommendation performance of our technique. A smaller value of *MAE* (and *RMSE*) indicate a better performance.

$$MAE = \frac{1}{N}\sum_{i} |Sswf_{true} - Sswf_{rec_i}| \tag{6}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Sswf_{true} - Sswf_{rec_i})^2}{N}} \tag{7}$$

where the symbol $Sswf_{true}$ represents the similarity value (denoted 1) of the exact fragment with respect to a certain requirement. The symbol $Sswf_{rec_i}$ indicates the similarity value about the recommended ith fragment according to our technique, where i is no more than the number of candidate fragments N.

To further evaluate the quality of the proposed method, we also use two other metrics, *precision* and *recall*, on the testing experiments. Generally, given a testing workflow swf_{tst}, a workflow swf_{ept} is assumed to be included in the *expected* list of recommendation (denoted SWF_{ept}). In our research, swf_{ept} retains the whole or part of the structure of swf_{tst}, whose activities are replaced by activities in the database. When similarity between swf_{tst} and swf_{ept} is no less than a pre-specified threshold $thrd_{ept}$, swf_{ept} is considered a fragment that user expects. We use the notation SWF_{rec} to denote the set of workflow fragments, which are actually recommended by our technique presented in this paper. The precision and recall are computed as follows:

$$precision = \frac{(|SWF_{ept} \cap SWF_{rec}|)}{|SWF_{rec}|} \tag{8}$$

$$recall = \frac{(|SWF_{ept} \cap SWF_{rec}|)}{|SWF_{ept}|} \tag{9}$$

where the symbol $|SWF_{rec}|$ refers to the number of fragments in the set SWF_{rec} while the symbol $|SWF_{ept}|$ specifies the number of elements in the set SWF_{ept}. And the symbol

$|SWF_{ept} \cap SWF_{rec}|$ refers to the number of fragments in the SWF_{ept} contained in the SWF_{rec}.

B. EVALUATING RESULT

We present the evaluating results of our technique as follows:

- Whether the modularity-based activity clustering can improve the recommended performance.
- What the impact of key parameters is upon the proposed model.

The results of our experiments in (i) Section V-B where a requirement is completely satisfied by a single scientific workflow, and (ii) Section V-C where the minor and major changes are applied to sample workflows when a requirement is partially satisfied by multiple scientific workflows, are adopted for the computation of *MAE*, *RMSE*, *precision* and *recall*.

1) IMPACT OF COMMUNITY CLUSTERING

To evaluate the impact of community clustering based on modularity, *Louvain*, we compare this clustering method with the *K-Means* clustering algorithm.

The performance of these two approaches is evaluated by comparing *MAE*, *RMSE*, *precision* and *recall* when key parameters, the number of instantiation *tp%* and pre-specified threshold $thrd_{ept}$, are set to 10% and 0.86, respectively. Fig. 5 shows that the clustering algorithm *Louvain* outperforms *K-Means* that transforms activities into functionality-relevant activities. Note that *K-Means* requires to manually set the parameter K. In order to maintain consistency with the number of *Louvain* clusters, we also set K to 119 to reduce the number of variable parameters on the exploration of clustering functionality. Specifically, *Louvain* obtains the smaller values of *MAE* (0.118347) and *RMSE* (0.033339) and the higher values of *precision* (0.887) and *recall* (0.838). The reason lies in that we prune the links between the distant activities according to the average path length (L) and clustering coefficient (C), which obtains the superior clustering network contributing to community discovery. Furthermore, *Louvain* clustering considers the semantic similarity between activities, and also synthesizes the overall structural relevance within the formative abstract activity, thus balancing the entire situation of the clustering network. Such implicit information cannot be easily discovered by *K-Means*, which will affect the discovery and recognition of fragments in subsequent processes.

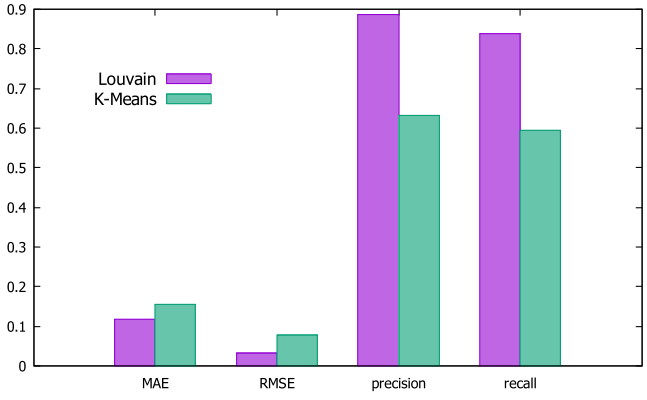


FIGURE 5. Impact of Louvain comparison MAE, RMSE, precision and recall.

2) IMPACT OF KEY PARAMETERS

As presented in the Section. IV-B, SWF_{rec} and SWF_{ept} are impacted by key parameters: *tp%* of instantiation recommendation and a pre-specified threshold $thrd_{ept}$. In this section, we study the impact of *tp%* and $thrd_{ept}$ to the *precision*, *recall*, *MAE* and *RMSE* in the following.

In our experiments, we firstly set $thrd_{ept}$ as 0.86, and tp% as 2%, 4%, 6%, 8%, 10%, 12%, 14%, 16%, 18%, 20%, 22%, 24%, respectively. Precision and recall are compared when (i) *Louvain* is adopted, or (ii) *K-Means* is applied instead. As shown in Fig. 6, due to the similar reason as presented in Fig. 5, the values of precision and recall for *Louvain* are larger than those for *K-Means*. This figure shows that when tp% is set to relatively large values, the precision drops to a large extent (from 0.982 to 0.453). This is due to the fact that more workflow fragment instantiations are recommended by our technique, which are actually not that relevant and not existing in SWF_{ept}. The recall is relatively more stable than the precision when tp% becomes large, since the set of SWF_{ept} is mainly determined by $thrd_{ept}$, which is a fixed value. Besides, when tp% is large than 16%, the recall is very high. This is due to the fact that the majority of the expected instantiations, which can be recommended by our technique, have been discovered and included in SWF_{rec}.

Fig. 7 shows the values of precision and recall where *tp%* is set to 20% and $thrd_{ept}$ is set to 0.78, 0.80, 0.82, 0.84, 0.86, 0.88, 0.90, 0.92, respectively. And the values of precision and recall for *Louvain* are also larger than those for *K-Means*. It is noteworthy that SWF_{rec} doesn't change when tp% is remained as a fixed value. However, the workflow fragments in SWF_{ept} decrease to an extent when $thrd_{ept}$ is set to a relatively large value. Therefore, the difference between the sets of $SWF_{spt} \cap SWF_{rec}$ and SWF_{ept} is decreased, which makes

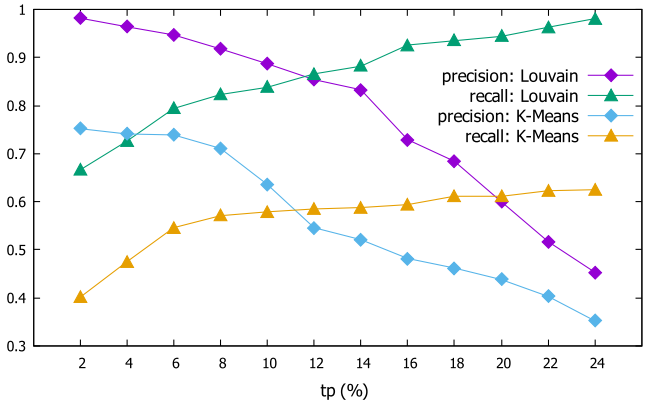


FIGURE 6. Comparison of precision and recall for Louvain and K-Means, when $thrd_{ept}$ is set to 0.86 and tp% is set to 2%, 4%, 6%, 8%, 10%, 12%, 14%, 16%, 18%, 20%, 22%, 24%, respectively.

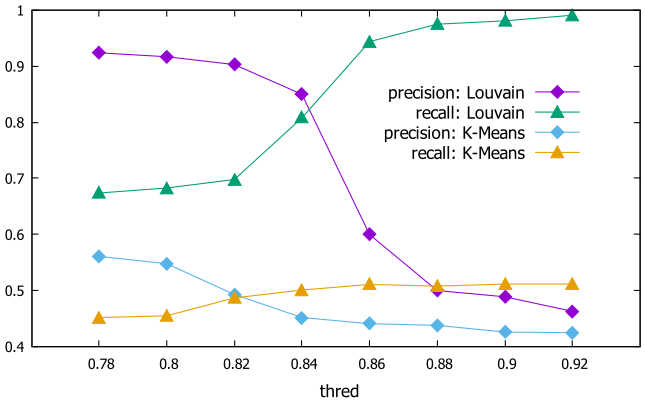


FIGURE 7. Comparison of precision and recall for Louvain and K-Means, when *tp*% is set to 20% and $thrd_{ept}$ is set to 0.78, 0.80, 0.82, 0.84, 0.86, 0.88, 0.90, 0.92, respectively.

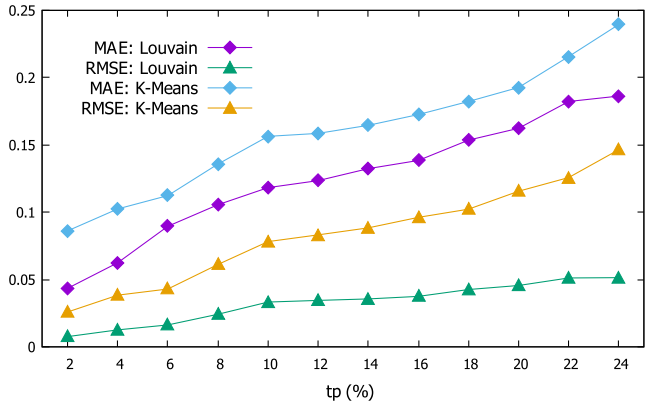


FIGURE 8. Comparison of *MAE* and *RMSE* for Louvain and K-Means, when $thrd_{ept}$ is set to 0.86 and tp% is set to 2%, 4%, 6%, 8%, 10%, 12%, 14%, 16%, 18%, 20%, 22%, 24%, respectively.

the increase of the recall. This figure shows that the recall is relatively stable when $thrd_{ept}$ changes from 0.78 to 0.82, and from 0.88 to 0.92, since there are quite few expected instantiations whose workflow fragment belongs to expected sets of workflow fragments within these two ranges. The recall increases to a large extent when $thrd_{ept}$ changes from 0.82 to 0.86, since the number of expected workflow fragments is large. Moreover, the number of workflow fragments in SWF_{ept} decreases when $thrd_{ept}$ is set to a relatively large value, which makes $|SWF_{ept} \cap SWF_{rec}|$ decrease as well. Therefore, the precision drops (from 0.924 to 0.463) along with the increase of $thrd_{ept}$.

Fig. 8 shows the comparison of *MAE* and *RMSE* for *Louvain* and *K-Means*, when $thrd_{ept}$ is set to 0.86 and *tp%* is set to 2%, 4%, 6%, 8%, 10%, 12%, 14%, 16%, 18%, 20%, 22%, 24%, respectively. Results indicate that our technique achieve the small error and the best performance compared with *K-Means*. When *tp%* is set to relatively large values, *MAE* and *RMSE* increase to a large extent. This is due to the fact that more low similarity instantiations are recommended by our technique. Although *MAE* and *RMSE* increase with the value of *tp%*, their values are maintained in an acceptable error range. Note that *MAE* and *RMSE* can only be influenced by the similarity value of the recommended instantiations, and the more the number of instantiations, the greater values of *MAE* and *RMSE*. However, when the *tp%* value is fixed, no matter how the $thrd_{ept}$ changes, the situation of recommended instantiations will not change, which only affect the change of *precision*, *recall*. Therefore, we do not need to discuss the effect of $thrd_{ept}$ on *MAE* and *RMSE*.

Generally speaking, the precision and recall should be balanced somehow for facilitating the recommendation of workflow fragments, such that a more number of appropriate and relevant workflow fragments should be recommended. The results of our experiments have shown that the precision drops and the recall increases when *tp%* and $thrd_{exp}$ increase. Therefore, we suggest that tp% and $thrd_{ept}$ should not be set to relatively large values for supporting real-world applications. Meanwhile, experiment shows that our technique is accurate on ranking and recommending appropriate fragments crossing multiple workflows, which is suitable for practical applications.

VII. RELATED WORK

Abundant scientific knowledge contributes to inside-school education, which is more significant for disciplines like sciences when complex scientific experiments are required. Thus, when novel scientific experiments are to be designed and conducted, reusing and repurposing scientific workflows can be regarded as an efficient learning mechanism for research scientists, which is an indispensable ingredient for achieving the objective of smart campus and others [1], [2], [32]. Scientific workflows play an important role in the reproducibility and replicability of scientific experiments. In the scientific experiments, a large variety of scientific workflow systems (e.g., Wings, Taverna, Galaxy, Vistrails) have been created to support scientists. When a scientist would like to conduct a novel scientific experiment, it is cost-effective to reuse and repurpose current workflows. Workflow reuse and repurposing is a promising research topic, and data on-line workflows built in e-Science have been the result of collaborative team efforts [12]. The authors have developed a workflow similarity computation technique to promote the reuse and repurposing of certain workflows in *myExperiment*, where workflows are transferred into layer hierarchies, and a layer hierarchy reflects hierarchical relations between this workflow, its sub-workflows, and activities [5]. Similar approaches have been developed, which can be categorized into annotation-based [13], [14], structure-based [15] and data-driven [16], [17] mechanisms. In general, these technologies explore similarity assessment issues from different perspectives, thereby suggesting techniques for adaptability between scientific workflows. There are techniques taking annotations into account for measuring the similarity of scientific workflows. In [13], a similar workflow search algorithm based on semantic type comparison has been proposed,

and the expanded experiment shows that semantics and ontology play significant role in service and workflow representation. The bag of tags and words are considered for assessing the similarity of scientific workflows [14]. Recommendation strategies are proposed to augment an in-progress workflow, leveraging the knowledge about workflows in the repository. Authors have adopted the similar principle to compute the similarity between scientific workflows [23]. The current state-of-the art in supporting the discovery of scientific workflows relies on annotations to be provided for these workflows by their authors, and for the potential re-users of these workflows to use the right keywords when searching for them. To improve this situation, in [15] the authors proposed an approach for measuring the structural similarity of scientific workflows. Note that a scientific workflow is assumed to be reduced to a directed acyclic graph in [15]. This assumption may not hold, since some workflows in the *myExperiment* repository is to be reduced to a directed cyclic graph. This is interesting, and it has inspired us to think of scientific workflows as cyclic graphs. Generally, these approaches are beneficial and complemental to structural-based approaches, when rich annotations are provided for specifying workflows. Although measuring the similarity of workflows based on annotations and structures can be effective, defining inexact and incomplete labels and the existence of multiple labels for similar activities cause challenges for determining similar processes. Recent attempts to consider data in business process management and the support of data modelling in business process standards have led to the creation of multiple business models with data access. In [16], a method considering data for measuring business process similarity is presented in which first the similarity of activities is measured according to their structures and behaviours in a process and also their data access. Then based on the similarity of activities, the similarity of processes is determined using the proposed algorithm. Starlinger [17] demonstrate the both approaches, which a hierarchical browsable view of the repository in which categories are derived using frequent itemset mining or latend Dirichlet allocation, may be used for effective data exploration. In addition, the question about data mechanism is researched by Zhu *et al.* [18] and Zhu *et al.* [33].

Besides, many requirements can be partially satisfied by a single scientific workflow, Therefore, typical fragments can be retrieved from workflows to improve their sharing and reuse when possible. Meanwhile, much of researches enter on reuse and re-purpose of their *workflow fragments*, where approaches to detection and recommendation are well developed. Lapeña *et al.* [19] proposed a novel approach, computer assisted CAO for models, that uses a multi-objective evolutionary algorithm with two objectives to rank relevant model fragment for reuse, which are easier to understand from the perspective of a software engineer. A keyword-based search method [34] for identifying the fragments that can be relevant for the needs of a given workflow designer is presented, and workflow fragments can be re-utilized and re-purposed by designers when specifying new workflows. The FragFlow approach, which detects workflow fragments automatically by analyzing existing workflow corpora with graph mining algorithm, is presented in [35]. How to disintegrate a workflow model and how to allocate its fragments to each of the components are considered to configure the underlying collaborative workflow system in [36] and [37]. In addition, how to detect most possible workflow fragments for recommendation in a meaningful way is a hot tendency. Sarno *et al.* [20] extract the common fragments using dependency graph calculation in a reconfigurable business process model. As presented in [38], for facilitating the understanding about the reuse and execution of scientific workflows, common workflow fragments from workflows in the repository are detected. However, these fragments in above approaches are obtained through each overall scientific workflow, rather than across various workflows like in our technique. It it worth emphasizing that a new experiment may be relevant with multiple experiments that have been conducted by others in most application scenarios, and thus, whose workflow should be constructed through assembling carefully-discovered fragments contained in various scientific workflows in the repository. Consequently, discovering appropriate fragments from multiple scientific workflows, and facilitating the reuse and repurposing of these assembled fragments, is a promising research challenge.

With a relatively large number of activities and scientific workflows increase in the repository, the activity discover procedure could be achieved by means of service discovery. Meanwhile, activities with same or similar functionalities should be considered as relevant. In fact, *abstract* activities in [22] has been discovered from event logs and represented as subprocesses contributing to the construction of process models. As presented in [39], a clustering frequency co-occurring activities mechanism is developed, where a cluster of activities is considered as a high level activity. However, if only frequency co-occurring activities are considered, functional research about activities are not comprehensive. What is more, the name or additional text description about activities can help scientist understand their real meanings compared with frequency more. Therefore, it is essential that all activities in various scientific workflows are involved to consider the functional clustering. In addition, for developing a new experiment that may be relevant with multiple experiments, functionally-similar activities that have been developed by other scientific workflows are more meaningful and has not been considered extensively. There are many traditional clustering algorithms, such as *K-Means*, *SNN*, *Graph-Skeleton* and so on, which have been applied in the service computing domain. When a complex network is used to clustering, community discovery clustering algorithms perform outstanding results and should also be worth considering in our technique for facilitating the formative of abstract activity.

Because of the existence of the name and text description in activities, which is more indicative of what they actually

mean, the semantic similarity is deemed to the connection between activities. Our technique computes the semantic similarity through WordNet between words [5], and does not assume the existence of a domain ontology. Thereafter, an activity network based similarity is generated through pruning links about the distance whose value is no less than a pre-specified threshold. Common data- and workflow-oriented motifs [40] represent kinds of data-operation activities and structural manners of realizing these activities, respectively, in scientific workflows. In fact, workflow design represents the best practice, which can be reused or repurposed for supporting research. Some functions or behaviors may be completed by semantically similar activities and structurally identical fragments, so that abstract activity fragments could be shared to use. Therefore, we accommodate the invocation relation specified upon contiguous abstract activities in the way of an abstract activity network, which represent many possible executable fragment templates for the fragment detection and recommendation. In the recommendation process, clusters formed by the *WfN* model haven't generated proper information for each cluster in [5]. However, in our experiments, it can speed up the fragment recognition and recommendation process. Therefore, we propose a strategy for generating key information for abstract activities. In [8], a novel empirical approach for mining ETL (Extract, Transform and Load) structural patterns using the *VF*2 graph matching algorithm is applied. Therefore, based on the idea of graph isomorphism, we consider it for the structural fragment mining. Consequently, fragment discovery are reasonably solved through semantic and structural matching on an abstract level. Moreover, a social network has been incorporated into sensor-cloud for sharing big data [41], and a social-network ananlysis has been applied to assess the reuse of scientific workflow fragments [9]. To summarize, these thoughts have promoted the study of our technology, which contributes to accurately detect and recommend workflow fragments.

In the case-based reasoning, a case is to be found for matching the user's query exactly, but similar cases may be found in some aspect which may be adapted for constructing novel fragments. This observation drives us that it is possible to find the exact solution to recommend without losing *right* case. In addition, discovering the fragments that are suitable for requirements have been explored in other ways [8], [42], [43]. In Roy Chowdhury *et al.* [42] propose to promote the reuse of mashup model patterns. A pattern weaving approach is proposed for promoting the reuse of composition knowledge. Meanwhile, [8] presented a pattern-based analysis of ETL workflows and the most frequent ETL patterns are discovered and identified. An efficient recommendation method for improving business process modeling is proposed in [44], where the patterns within processes are extracted from the repository through a graph-mining technique. Traditionally, pattern detection approaches based on predefined measures [45] have some limitations. How to set the values of predefined measures is a difficult question, which leads to the *pattern explosion* or *pattern shortage* issues. This means that these predefined measures should be avoided, at the same time, it is possible to detect workflow fragments across-the-board. In our experiments, an abstract activity network model is constructed without considering the setting of threshold according to our technique, in fact, abstract activity fragments could also been seen as patterns. Besides, experimental evaluation of our technique shows that the detection and recommendation of cross-workflow fragments is accurate and efficient.

VIII. CONCLUSION

This article proposes to a cross-workflow fragments discovery mechanism to promote the reuse and repurposing of these fragments which partially belong to various scientific workflows. Specifically, the degree of semantic similarity for pairs of activities in various scientific workflows are calculated. Abstract activities, which represent certain groups of functionally-similar activities, are generated through adopting the modularity-based activity clustering technique. An abstract activity network is constructed, whose vertices correspond to abstract activities, and directed edges reflect the invocation relation specified upon contiguous abstract activities. Given a scientist's requirement that is specified in terms of a workflow template, structural and semantic similar workflow fragments are discovered from the abstract activity network through the sub-graph matching algorithm. These fragments are instantiated through replacing abstract activities by appropriate activities in certain activity clusters. These instantiated workflow fragments are ranked and recommended for their reuse and repurposing purpose, where the factors including the path length ratio and average semantic similarity are adopted as the criteria. Evaluation results demonstrate that our technique is accurate and efficient on discovering and recommending appropriate cross-workflow fragments.

ACKNOWLEDGEMENT

The authors are grateful to the anonymous reviewers for their constructive comments.

REFERENCES

[1] X. Zhai, Y. Dong, and J. Yuan, "Investigating learners' technology engagement—A perspective from ubiquitous game-based learning in smart campus," *IEEE Access*, vol. 6, pp. 10279–10287, 2018.
[2] C. Zhu, J. J. P. C. Rodrigues, V. C. M. Leung, L. Shu, and L. T. Yang, "Trust-based communication for the industrial Internet of Things," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 16–22, Feb. 2018.
[3] J. Starlinger, B. Brancotte, S. Cohen-Boulakia, and U. Leser, "Similarity search for scientific workflows," *Proc. VLDB Endowment*, vol. 7, no. 12, pp. 1143–1154, 2014.
[4] A. Bolt, M. de Leoni, and W. M. van der Aalst, "Scientific workflows for process mining: Building blocks, scenarios, and implementation," *Int. J. Softw. Tools Technol. Transf.*, vol. 18, no. 6, pp. 607–628, 2016.
[5] Z. Zhou, Z. Cheng, L.-J. Zhang, W. Gaaloul, and K. Ning, "Scientific workflow clustering and recommendation leveraging layer hierarchical analysis," *IEEE Trans. Services Comput.*, vol. 11, no. 1, pp. 169–183, Jan./May 2018.
[6] R. Bergmann and Y. Gil, "Similarity assessment and efficient retrieval of semantic workflows," *Inf. Syst.*, vol. 40, pp. 115–127, Mar. 2014.

[7] J. Starlinger, S. Cohen-Boulakia, S. Khanna, S. Davidson, and U. Leser, "Effective and efficient similarity search in scientific workflow repositories," *Future Gener. Comput. Syst.*, vol. 56, pp. 584–594, Mar. 2016.

[8] V. Theodorou, A. Abelló, M. Thiele, and W. Lehner, "Frequent patterns in ETL workflows: An empirical approach," *Data Knowl. Eng.*, vol. 112, pp. 1–16, Nov. 2017.

[9] W. Tan, J. Zhang, and I. Foster, "Network analysis of scientific workflows: A gateway to reuse," *Computer*, vol. 43, no. 9, pp. 54–61, Sep. 2010.

[10] M. Klush, P. Kapahnke, S. Schulte, F. Lecue, and A. Bernstein, "Semantic web service search: A brief survey," *Künstliche Intelligenz*, vol. 30, no. 2, pp. 139–147, 2016.

[11] J. Zarrin, R. L. Aguiar, and J. P. Barraca, "Resource discovery for distributed computing systems: A comprehensive survey," *J. Parallel Distrib. Comput.*, vol. 113, pp. 127–166, Mar. 2018.

[12] A. Goderis, U. Sattler, P. Lord, and C. Goble, "Seven bottlenecks to workflow reuse and repurposing," in *Proc. Int. Semantic Web Conf.*, 2005, pp. 323–337.

[13] M. Krzywucki and S. Polak, "Workflow similarity analysis," *Comput. Inf.*, vol. 30, no. 4, pp. 773–791, 2011.

[14] J. Stoyanovich, B. Taskar, and S. Davidson, "Exploring repositories of scientific workflows," in *Proc. 1st Int. Workshop Workflow Approaches New Data-Centric Sci.*, 2010, Art. no. 7.

[15] J. Starlinger, S. Cohen-Boulakia, S. Khanna, S. B. Davidson, and U. Leser, "Layer decomposition: An effective structure-based approach for scientific workflow similarity," in *Proc. IEEE 10th Int. Conf. e-Sci.*, vol. 1, Oct. 2014, pp. 169–176.

[16] M. J. Amiri and M. Koupaee, "Data-driven business process similarity," *IET Softw.*, vol. 11, no. 6, pp. 309–318, Dec. 2017.

[17] J. Starlinger, "Similarity measures for scientific workflows," Ph.D. dissertation, Mathematisch-Naturwissenschaftliche Fakultat, Humboldt Univ. Berlin, Berlin, Germany, May 2016.

[18] C. Zhu, V. C. M. Leung, K. Wang, L. T. Yang, and Y. Zhang, "Multi-method data delivery for green sensor-cloud," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 176–182, May 2017.

[19] R. Lapeña, J. Font, C. Cetina, and O. Pastor, "Model fragment reuse driven by requirements," in *Proc. 29th Int. Conf. Adv. Inf. Syst. Eng. (CAiSE)*, 2017, pp. 12–16.

[20] R. Sarno *et al.*, "Workflow common fragments extraction based on WSDL similarity and graph dependency," in *Proc. Int. Seminar Intell. Technol. Appl.*, May 2015, pp. 309–314.

[21] C. Zeng, T. Zhang, and P. C. K. Hung, "Fast service process fragment indexing and ranking," *IEEE Trans. Services Comput.*, vol. 9, no. 5, pp. 672–685, Sep./Oct. 2016.

[22] R. Conforti, M. Dumas, L. García-Bañuelos, M. La Rosa, "BPMN Miner: Automated discovery of BPMN process models with hierarchical structure," *Inf. Syst.*, vol. 56, pp. 284–303, Mar. 2016.

[23] Z. Zhou, Z. Cheng, K. Ning, W. Li, and L.-J. Zhang, "A sub-chain ranking and recommendation mechanism for facilitating geospatial Web service composition," *Int. J. Web Services Res.*, vol. 11, no. 3, pp. 52–75, 2014.

[24] R. Daland, "Word segmentation, word recognition, and word learning: A computational model of first language acquisition," Ph.D. dissertation, Northwestern Univ., Evanston, IL, USA, 2009.

[25] C. Stein and J. Wein, "Approximating the minimum-cost maximum flow is P-complete," *Inf. Process. Lett.*, vol. 42, no. 6, pp. 315–319, 1992.

[26] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech., Theory Exp.*, vol. 2008, p. P10008, Oct. 2008.

[27] A. Lancichinetti and S. Fortunato, "Community detection algorithms: A comparative analysis," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 80, no. 5, p. 056117, 2009.

[28] S. Rose, D. Engel, N. Cramer, and W. Cowley, "Automatic keyword extraction from individual documents," in *Text Mining: Applications and Theory*. Hoboken, NJ, USA: Wiley, 2010, pp. 1–20.

[29] C. Fox, "A stop list for general text," *ACM SIGIR Forum*, vol. 24, nos. 1–2, pp. 19–21, 1989.

[30] R. Mihalcea and P. Tarau, "TextRank: Bringing order into text," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2004, pp. 1–8.

[31] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento, "A (sub)graph isomorphism algorithm for matching large graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 10, pp. 1367–1372, Oct. 2004.

[32] C. Zhu, L. Shu, V. C. M. Leung, S. Guo, Y. Zhang, and L. T. Yang, "Secure multimedia big data in trust-assisted sensor-cloud for smart city," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 24–30, Dec. 2017.

[33] C. Zhu, X. Li, V. C. Leung, L. T. Yang, E. C.-H. Ngai, and L. Shu, "Towards pricing for sensor-cloud," *IEEE Trans. Cloud Comput.*, to be published, doi: 10.1109/TCC.2017.2649525.

[34] K. Belhajjame, D. Grigori, M. Harmassi, and M. B. Yahia, "Keyword-based search of workflow fragments and their composition," in *Transactions on Computational Collective Intelligence XXVI*. Cham, Switzerland: Springer, 2017, pp. 67–90.

[35] D. Garijo *et al.*, "FragFlow automated fragment detection in scientific workflows," in *Proc. IEEE 10th Int. Conf. e-Sci.*, vol. 1, Oct. 2014, pp. 281–289.

[36] K. Kim, "A model-driven workflow fragmentation framework for collaborative workflow architectures and systems," *J. Netw. Comput. Appl.*, vol. 35, no. 1, pp. 97–110, 2012.

[37] I. Al Ridhawi, Y. Kotb, and Y. Al Ridhawi, "Workflow-net based service composition using mobile edge nodes," *IEEE Access*, vol. 5, pp. 23719–23735, 2017.

[38] D. Garijo, O. Corcho, and Y. Gil, "Detecting common scientific workflow fragments using templates and execution provenance," in *Proc. Int. Conf. Knowl. Capture*, 2013, pp. 33–40.

[39] C. W. Günther, A. Rozinat, W. M. Van Der Aalst, "Activity mining by global trace segmentation," in *Proc. Int. Conf. Bus. Process Manage.*, 2009, pp. 128–139.

[40] D. Garijo, P. Alper, K. Belhajjame, O. Corcho, Y. Gil, and C. Goble, "Common motifs in scientific workflows: An empirical analysis," in *Proc. IEEE Int. Conf. E-Sci.*, Oct. 2012, pp. 1–8.

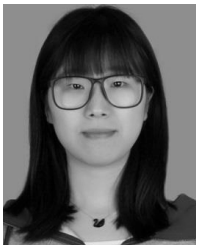
[41] C. Zhu, H. Zhou, V. C. M. Leung, K. Wang, Y. Zhang, and L. T. Yang, "Toward big data in green city," *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 14–18, Nov. 2017.

[42] S. Roy Chowdhury, F. Daniel, and F. Casati, "Recommendation and weaving of reusable mashup model patterns for assisted development," *ACM Trans. Internet Technol.*, vol. 14, nos. 2–3, 2014, Art. no. 21.

[43] T. Xu, T. Li, and X. Dong, "Efficient high utility negative sequential patterns mining in smart campus," *IEEE Access*, vol. 6, pp. 23839–23847, 2018.

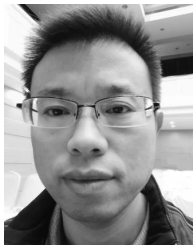
[44] Y. Li *et al.*, "An efficient recommendation method for improving business process modeling," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 502–513, Feb. 2014.

[45] R. P. J. C. Bose and W. M. Van der Aalst, "Abstractions in process mining: A taxonomy of patterns," in *Proc. Int. Conf. Bus. Process Manage.*, 2009, pp. 159–175.

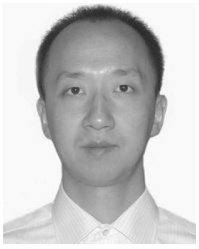


JINFENG WEN is currently pursuing the master's degree with the School of Information Engineering, China University of Geosciences (Beijing). Her research interests include services computing and business process management.

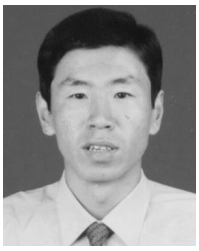
ZHANGBING ZHOU is currently a Professor with the China University of Geosciences (Beijing), China, and also an Adjunct Professor at TELECOM SudParis, France. His research interests include wireless sensor networks, services computing, and business process management.



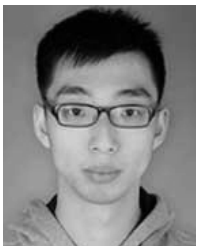
ZHENSHENG SHI is currently a Senior Engineer with the PetroChina Research Institute of Petroleum Exploration and Development. His research interests include services computing and wireless sensor networks.



YUCONG DUAN is currently a Professor with the College of Information Science and Technology, Hainan University. His research interests include services computing.



JUNPING WANG is currently an Associate Research Fellow with the Laboratory of Precision Sensing and Control Center, Institute of Automation, Chinese Academy of Sciences. His research interests include services computing and business process management.



YAQIANG ZHANG is currently pursuing the Ph.D. degree with the School of Information Engineering, China University of Geosciences (Beijing). His research interests include services computing and wireless sensor networks.