A process recommendation method using bag-of-fragments

Jiaxing Wang, Sichao Gui and Bin Cao*

College of Computer Science and Software Engineering,
Zhejiang University of Technology,
Hangzhou, 310023, China
Email: wjx@zjut.edu.cn
Email: sichao688@163.com
Email: bincao@zjut.edu.cn
*Corresponding author

Abstract: Process modelling is one of the key techniques in business process management, which needs to meet the frequent changes in custom and market demands effectively and efficiently. Most of existing methods focus on the structural feature of a process when modelling a process by using graph edit distance (GED) technology. However, GED is low-efficiency and the costs need to be adjusted for different scenarios. Besides, two process models with the same structure may contain different behaviours. To address this, we use a bag-of-fragments based on *m*, *n*-grams that are excerpts in terms of structure and behaviour to summary a process model. Given a process that is under modelling, we recommend the top *k* similar process models in the process repository for process modelers, which provides them the relevant decision support and assists them in modelling this process model. A prototype is implemented to show the practicality of the proposed technique.

Keywords: process model; *m*, *n*-gram; process difference; process recommendation; process similarity.

Reference to this paper should be made as follows: Wang, J., Gui, S. and Cao, B. (2019) ‘A process recommendation method using bag-of-fragments’, *Int. J. Intelligent Internet of Things Computing*, Vol. 1, No. 1, pp.32–42.

Biographical notes: Jiaxing Wang received her PhD in Control Science and Engineering from the Zhejiang University of Technology in 2019. She is currently a Postdoc in the College of Computer Science and Technology at the Zhejiang University of Technology. Her research interest is business process management.

Sichao Gui is a Master student in the College of Computer Science and Technology at the Zhejiang University of Technology. His research interest is business process management.

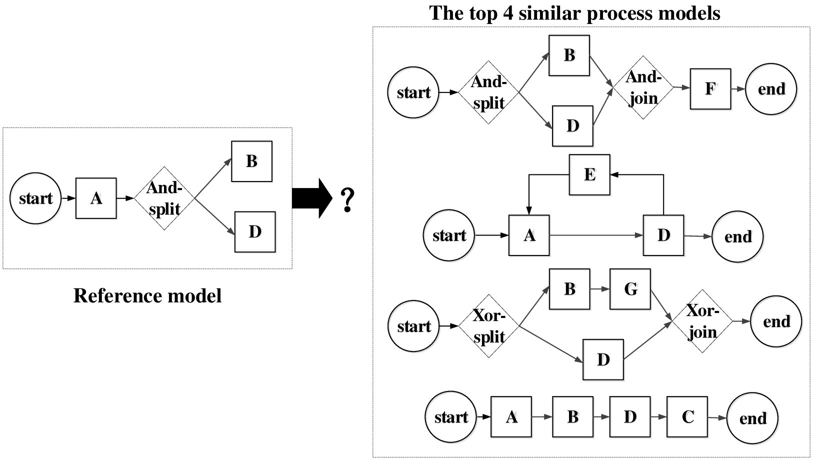
Bin Cao received his PhD in Computer Science from the Zhejiang University, China, in 2013. He then worked as a Research Associate in the Hong-kong University of Science and Technology and Noah’s Ark Lab, Huawei. He

joined the Zhejiang University of Technology, Hangzhou, China, in 2014, and is now an Associate Professor in the College of Computer Science. His research interests include business process management and data mining.

1 Introduction

Business process management (BPM) is more and more popular among companies and organisations, which leads to the creation of hundreds or even thousands of process models. For example, China CNR is a company regrouped from more than 20 subsidiary companies, where a total of more than 200,000 business process models were deployed in their own information systems before merged (Cao et al., 2016). Thus, business processes become one of the key tools for companies and organisations to handle different business operations. How to efficiently model a process in order to meet the frequent changes in customer demands and market is a challenge. However, modelling a process by hand is tedious, time-consuming and error-prone (Li et al., 2013), so the process modeler cannot quickly and fully understand the changed demands, which leads to the built process model cannot meet the real situation. In this way, the benefits of company will be affected since the accuracy of the decision for companies decreases.

Figure 1 An example of process recommendation



The top 4 similar process models

Reference model

To improve the business process modelling, many business intelligence (BI) methods were adopted, such as process recommendation. The main idea of process recommendation is to calculate the similarity between the ‘reference model’ and each process model in the repository, and select the top-*k* similar models as the recommendation models for ‘reference model’. Taking Figure 1 as an example, the user is currently modelling a process on the left hand, i.e., ‘reference model’, and he wants to know which node needs to be modeled and in which structure. Then, he can use the process recommendation technique, and reference the corresponding nodes from the top 4 similar process models in the process repository, which is shown in the right.

The traditional way is to take the topology structures of two process models as the input, and directly compute their similarity by using techniques such as graph edit distance (GED). However, the topology structure of a process model is static, which cannot reveal the ordered and unordered process control constructs such as parallel and conditional constructs. That is, the two process models contain the same structure, while they may contain different behaviours. Besides, directly computing the similarity between two process models based on GED is time-consuming since GED computing is the NP problem.

To overcome the above problems, we propose a bag-of-fragments based approach for process recommendation. We first transform the 'reference model' and each process model in the repository into their corresponding task-based process structure trees (TPSTs) and split these TPSTs into fragments that described by *m*, *n*-gram, which are excerpts to summary each TPST. These *m*, *n*-grams show not only structural but also behavioural features of a TPST. Next, we calculate the similarity between 'reference model' and every process model in the repository. Given two process models, a bag-of-fragments that describes the occurrence time of each *m*, *n*-gram in the union of two process models' *m*, *n*-gram set is created, and the similarity of two process models is represented by the similarity of their corresponding bag-of-fragments. Finally, we select the top *k* similar ones to be recommended for 'reference model', which provides process modelers the relevant decision support and assists them in modelling a new process.

Next, we highlight our contributions as follows:

- A set of *m*, *n*-grams are used to describe both the structural and behavioural feature of a process model.
- Bag-of-fragments that consists of the occurrence time of each *m*, *n*-gram in a process model is the excerpt to summary each TPST.

The rest of this paper is organised as follows. Section 2 sets the stage for the concepts used in this paper. The implementation is presented in Section 3. In Section 4, a prototype is developed. Section 5 reviews the related work, and Section 6 concludes this paper.

2 Preliminaries

This section presents a set of preliminaries that are important to set the stage for understanding this paper and its vision. In particular, we present the process modelling and the TPST respectively, which are the basis of our work.

2.1 Process modelling

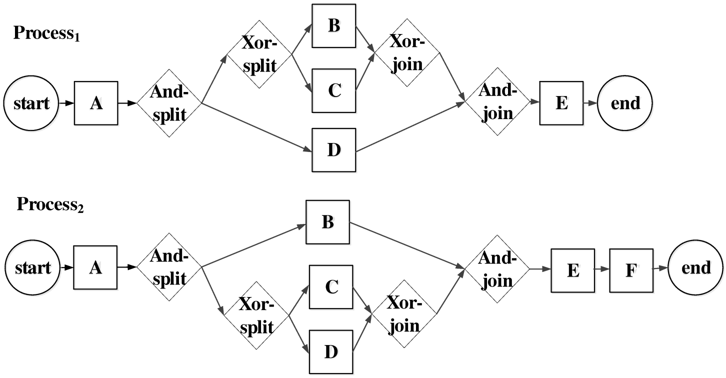
A process consists of a set of tasks and their relations to reach a goal, which can be described by a directed graph consists of four different kinds of nodes:

- *Activity nodes* denote the tasks to be performed to human or automated resources, which are represented with rectangles.

- *Gateway nodes* are decision points that route the execution flow among nodes, which are represented with diamond shapes and include *And-split*, *Xor-split*, *loop-split*, *And-join*, *Xor-join*, *loop-join*.
- *Start node* represents the entry point to a process, which describes as '*start*'.
- *End node* are the termination point of a process, which describes as '*end*'.

Taking $Process_2$ in Figure 2 as an example, its task node set is $\{A, B, C, D, E, F\}$, the gateway node set is $\{And\text{-}split, Xor\text{-}split, And\text{-}join, Xor\text{-}join\}$ that consists of parallel and conditional patterns. Its start and end nodes are *start* and *end*, and it contains 13 edges, such as $start \rightarrow A$ and $E \rightarrow F$.

Figure 2 Two process models with conditional and parallel patterns



There are four types of process patterns in a process model, i.e., parallel, conditional, loop and sequential patterns. For sequential pattern, there exists no gateway node and each node in it has exactly one incoming arc and one outgoing arc. In parallel and conditional patterns, the task nodes in them are simultaneously and exclusively executed, respectively. With regard to the loop pattern, its task nodes can be repeatedly executed.

Figure 3 Two task-based process structures (TPSTs)

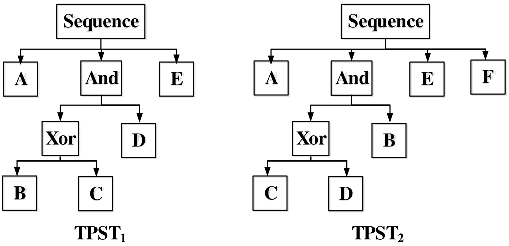


Figure 2 shows two process models that contain three types of control flow patterns. Taking $Process_2$ as an example, the whole process model is a sequential pattern that consists of task *A*, the parallel pattern, task *E* and *F*, and they are sequentially executed. Task *C* and *D* form a conditional pattern, where one of these two task nodes is chosen to be executed. The parallel pattern consists of the conditional pattern and task *D*, and they are simultaneously executed.

2.2 Task-based process structure tree

A process model can be transformed into a tree structure, i.e., process structure tree (PST) (Vanhatalo et al., 2009). In a PST, each leaf and route node separately correspond to a task node and a control flow pattern in the process model. To explicitly show the task nodes as well as the control flow patterns in a tree structure, we introduce the task-based process structure tree (TPST) (Fan et al., 2017) to represent a process model, which is a variant of a PST. The features of a TPST are listed in the following:

- The leaf nodes of a TPST represent the task nodes of its corresponding process model, the non-leaf nodes of a TPST represent the control flow patterns of its corresponding process model.
- There are four types of gateway nodes in a TPST: *Sequence*, *Loop*, *Xor* and *And*, which are equal to the sequential, loop, conditional and parallel pattern respectively.
- TPST is semi-ordered, that is, the child nodes in *Xor* and *And* have no order, while the child nodes of *Sequence* and *Loop* are ordered execution.

As shown in Figure 3, there are two TPSTs $TPST_1$ and $TPST_2$ that are transformed by two process models in Figure 2. Taking $TPST_1$ as an example, its leaf nodes are the task nodes of $Process_1$, i.e., $\{A, B, C, D, E\}$. Its non-leaf nodes are $\{Sequence, And, Xor\}$, where the root node of $TPST_1$ *Sequence* represents that the highest level of abstraction of $Process_1$, which is a sequential pattern.

3 Process recommendation

The main idea of our approach is to calculate the similarity between the 'reference model' and each process model in the process repository, and the top-*k* similar process models in the repository are selected as the recommendation process models of 'reference model'. For the next node to be modeled in the 'reference model', its top-*k* similar nodes are found in these recommendation process models. That is, these top-*k* similar nodes are the recommendation nodes. To measure the similarity between two process models, we use a set of *m*, *n*-grams to represent a process model, and the similarity between two process models are measured based on their *m*, *n*-grams. A *m*, *n*-gram extracts a part of a process model, which represents a local structural as well as behavioural feature of a process model. The more *m*, *n*-grams two process models share, the more similar they are. In a summary, the proposed approach is composed of three consecutive phases, namely, *m*, *n-gram extraction*, *similarity calculation*, and *process recommendation*. These three phases are described in detail in the rest of this section.

3.1 Phase 1: m, n-gram extraction

The goal of this phase is to extract a set of *m*, *n*-grams from each TPST that is transformed from a process model. *Grams* mean a set of small excerpts used to summary a tree (Finis et al., 2013). In this paper, we use *m*, *n*-gram to summary a TPST, where each *m*, *n*-gram is a besom-shaped subtree consisting of *n* gateway nodes and a chain of their child nodes with size *m*. The *m*, *n*-grams of a TPST partially capture both structure and behaviour of a process.

Figure 4 *m*, *n*-gram fragment extraction

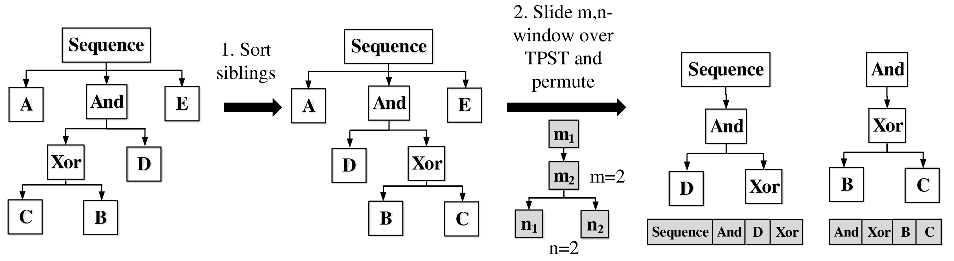
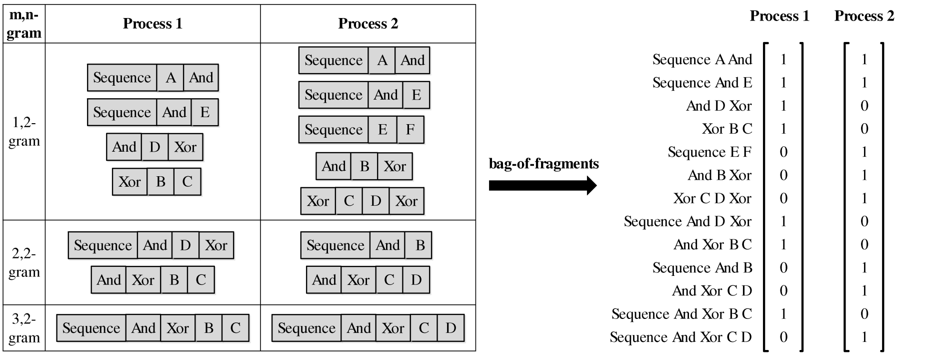


Figure 5 Bag-of-fragments



m,n-gram	Process 1	Process 2
1,2-gram	Sequence A And Sequence And E And D Xor Xor B C	Sequence A And Sequence And E Sequence E F And B Xor Xor C D Xor
2,2-gram	Sequence And D Xor And Xor B C	Sequence And B And Xor C D
3,2-gram	Sequence And Xor B C	Sequence And Xor C D

bag-of-fragments

$$\begin{array}{r|c|c} & \text{Process 1} & \text{Process 2} \\ \text{Sequence A And} & 1 & 1 \\ \text{Sequence And E} & 1 & 1 \\ \text{And D Xor} & 1 & 0 \\ \text{Xor B C} & 1 & 0 \\ \text{Sequence E F} & 0 & 1 \\ \text{And B Xor} & 0 & 1 \\ \text{Xor C D Xor} & 0 & 1 \\ \text{Sequence And D Xor} & 1 & 0 \\ \text{And Xor B C} & 1 & 0 \\ \text{Sequence And B} & 0 & 1 \\ \text{And Xor C D} & 0 & 1 \\ \text{Sequence And Xor B C} & 1 & 0 \\ \text{Sequence And Xor C D} & 0 & 1 \end{array}$$

The *m*, *n*-gram construction is illustrated in Figure 4 ($m = 2$, $n = 2$). First, the child nodes whose parent nodes are *And* and *Xor* need to be sorted lexicographically by labels, it is because the child nodes in these two kinds of process patterns has no order, while the child nodes in *Sequence* and *Loop* are ordered executed. That is, *Xor* and *D*, *C* and *B* that are the child nodes of *And* and *Xor*, respectively, are sorted into *D* and *Xor*, *B* and *C*. Then, for each gateway node *g* in the TPST, i.e., *Sequence*, *And* and *Xor*, a window with size $w \leq n$ is slided over the child nodes of *g* and *m*, *n*-grams are extracted. In this example, we set $m = 2$ and $n = 2$, so we can obtain two 2,2-grams. Finally, the *m*, *n*-grams are serialised into arrays of size $m + n$, which are listed in the bottom of each 2,2-gram.

3.2 *Phase 2: similarity calculation*

In this phase, the similarities between the 'reference model' and each process model in the process repository are calculated. The similarity between two process models is represented by the similarity of their respective bag-of-fragments. The bag-of-fragments of a process model is actually a feature vector, just like the bag-of-words that was first proposed for representing a text document and analysing in the text retrieval domain

(Tsai, 2012), it can be defined as follows.

Theorem 1 bag-of-fragments: Given two sets of serialisation of *m*, *n*-grams S_1 and S_2 that are from two TPSTs, we summarise the counts N_{ij} = *n* (s_i, t_j) (*i* = 1, 2, 3, ... and *j* = 1, 2) that denotes how often the serialisation $s_i \in \{S_1 \cap S_2\}$ occurred in a TPST t_j.

Taking Figure 3 as an example, the *m*, *n*-grams of two TPSTs are 1,2-grams, 2,2-grams and 3,2-grams, and their respective serialisations are listed on the left hand of Figure 5. In this example, we set *m* to 1, 2 and 3 since the gateway nodes only exist in the first 3 levels, and *n* to 2 since most of the gateway nodes have 2 child nodes. Users can customise the number of *m* and *n*, respectively. To construct the bag-of-fragments, we first get the union set of *m*, *n*-grams in two serialisations, and count the occurrence time of each *m*, *n*-gram in this union set. The bag-of-fragments of *Process* 1 and *Process* 2 are shown in the right side of Figure 5, which are [1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0] and [1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1], respectively. For example, the 1,2-gram consists of ‘Sequence’, ‘A’, and ‘B’ appears once in *Process* 1, while *Process* 1 does not contain the 3,2-gram.

Since the bag-of-fragments of a process model is a vector, we use the cosine similarity to measure the similarity between two bag-of-fragments, which represents the similarity between their corresponding two process models. Given two bag-of-fragments *A* = [A_1, ..., A_i, ..., A_n] and *B* = [B_1, ..., B_i, ..., B_n], their cosine similarity can be measured by the following equation:

$$\mathit{Similarity}(A, B) = \frac{\sum_{i=1}^{n}(A_i \times B_i)}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}} \tag{1}$$

For example, the similarity of two process models shown in Figure 2 is represented by the cosine similarity of their respective bag-of-fragments, which is 0.365.

3.3 Phase 3: process recommendation

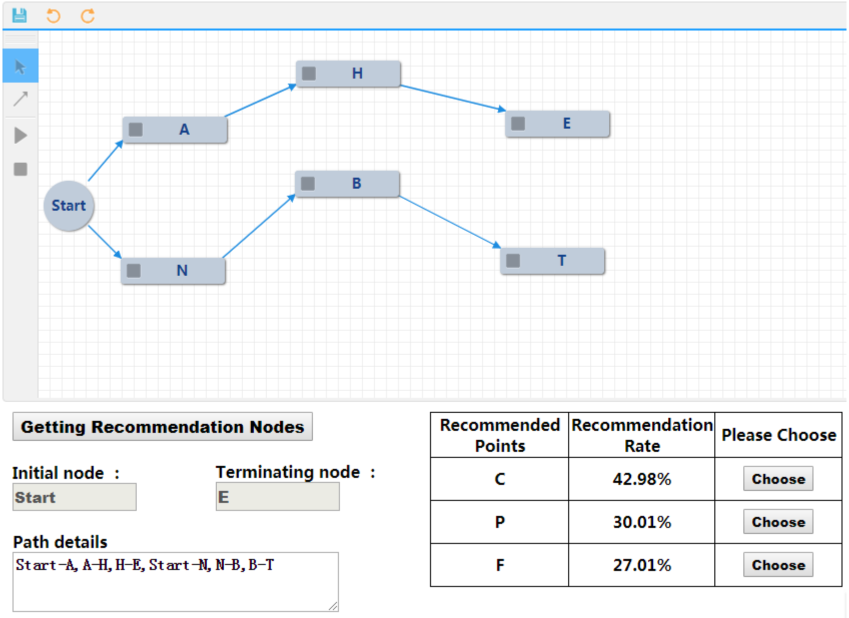
The objective of this phase is to recommend top-*k* similar process models $\{p_1, p_2, ..., p_k\}$ in the repository $\{p_1, p_2, ..., p_k, ..., p_n\}$ for the ‘reference model’ *r*, where *Similarity*(p_i, *r*) $\geq$ *Similarity*(p_j, *r*) (1 $\leq$ *i* $\leq$ *k*, and *k* < *j* < *n*). In this way, the recommended nodes for the next node to be modelled in ‘reference model’ are determined in these top-*k* similar process models. Since we know the current modeled node ‘c’, the nodes executed after node ‘c’ in these recommendation process models are found.

How to rank the top-*k* similar process models in the repository when there exists more than one process models $\{p_1, p_2, ..., p_m\}$ have the same similarity with the ‘reference model’ *r*? To solve this problem, we create the following strategy: we first perform the logical ‘and’ for the bag-of-fragments of $\{p_i\}$ (1 $\leq i \leq$ *m*) and *r*, and record the result into the map where the key is p_i and value is v_i that represents the number of 1. Then, the process models are ranked according to the number of 1 from high to low.

4 Prototype

Based on the proposed method, we develop a prototype system which allows users to model and store the process models, a snapshot of our prototype system is shown in Figure 6. The system allows users to view, store, generate and import/export process models. The main function of this system is to recommend a set of recommendation nodes for users once they press the button ‘get recommendation nodes’, and the recommendation nodes are ranked according to the recommendation rate from high to low. Users can choose the preferred node from the recommendation node list to model the current process.

Figure 6 Prototype (see online version for colours)



Initial node : Start

Terminating node : E

Path details

Start-A, A-H, H-E, Start-N, N-B, B-T

Recommended Points	Recommendation Rate	Please Choose
C	42.98%	Choose
P	30.01%	Choose
F	27.01%	Choose

The main page of the system consists of two modules. The big pane on the top shows the process model fragment that is under construction, where users can create different kinds of nodes and connect them by edges. The left side on the bottom panel gives the detail information of the current modeled process fragment, including the start node, end node, and the edges. The names and recommendation rates of the top 3 recommendation nodes are shown on the right hand of the bottom pane. Once the user determines which recommendation node to be used in the current modeled process fragment, he can press the button ‘Choose’ and the selected node will automatically appear in the main page and connected the related existing nodes by edges.

5 Related work

Process recommendation refers to the idea of traditional recommendation system (Sarwar et al., 2000), i.e., it helps process modelers build a process by analysing the relationship between the given fragment and each process model in the repository and recommend the most relevant one, which can speed up the process of modelling a process. The related work can be classified into two categories: process recommendation, and process similarity calculation.

The first category is process recommendation. Kluza et al. (2013) provided an overview of recommendation techniques in business process modelling, which introduced a categorisation and gave examples of recommendation methods. Cao et al. proposed a graph-based workflow recommendation for improving business process modelling, where graph mining method was used to extract the process patterns from the repository. Given a reference model that was under modelling, the candidate nodes with smaller GED were recommended (Cao et al., 2012). However, this method cannot handle the complicated control flow constructs such as parallel and loop constructs, thus it cannot meet the requirements in the real applications. Deng et al. (2016) presented a process recommendation system that can recommend proper nodes based on patterns mined from existing process repository, where three different recommendation strategies were designed. Fellmann et al. (2018) proposed a recommendation-based business process modelling approach, where the goal is to get the evidence of how well process modelling recommender systems might ease daily work of modelers.

The second category is process similarity calculation. Usually, there three similarity metrics to measure the similarity between two process models (Dijkman et al., 2011):

1 text similarity based on the label attached to nodes

2 structural similarity in terms of the topology

3 behavioural similarity relates to the execution path.

Wang et al. (2014) provided an overview of the field of querying business process models in terms of similarity calculation. Akkiraju and Ivan (2010) determined the semantic similarity between process models based on activity labels. Kunze and Weske (2010) presented an indexing approach based on metric trees, and GED was used to measure the similarity between two process models. In the later work, Huang et al. (2015) proposed an improved two-stage exact query approach based on graph structure similarity. Assy et al. measured the process similarity based on the contextual similarity, where the similarity of the context surrounding activities was considered. The global contextual similarity between process activities was computed by similarity resonance (Assy et al., 2018). Liu et al. (2019) proposed an approach to measure the business process behaviour similarity based on the so-called extended transition relation set, which was an extended transition relation set containing direct causal transition relations, minimum concurrent transition relations and transitive causal transition relations.

6 Conclusions and future work

In this paper, we propose a novel process recommendation technique, which improves the efficiency of process modelling by providing a reference to complete the process under construction. To overcome the drawbacks of GED based approaches, we use a bag-of-fragments that consists of a set of *m*, *n*-grams to describe the features of a process model in terms of structure and behaviour. In this way, the process model is split into different fragments and the similarity between 'reference model' and each process model in the repository can be efficiently calculated, and top-*k* nodes in the top-*k* similar models are recommended to the 'reference model'. To show the practicality of the proposed technique, we implement a prototype that helps users model a process by recommending top-*k* nodes, and users can choose the most suitable one.

The limitation of this work is that *m*, *n*-grams just show local structure and behaviour of a process model, which cannot reveal the global structural and behavioural features. In the future, we are going to take both local and global features into consideration for process recommendation. Besides, we just recommend the nodes to users in the prototype, which does not show the structure of these recommended nodes. Thus, how to connect the recommend node to the process model fragment that is under construction is decided by users.

Acknowledgements

This work was partially supported by: Zhejiang Provincial Natural Science Foundation of China (LY19F020030).

References

Akkiraju, R. and Ivan, A. (2010) 'Discovering business process similarities: an empirical study with sap best practice business processes', in *International Conference on Service-Oriented Computing*, Springer, pp.515–526.

Assy, N., Van Dongen, B.F. and Van Der Aalst, W.M. (2018) 'Similarity resonance for improving process model matching accuracy', in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, ACM, pp.86–93.

Cao, B., Yin, J., Deng, S., Wang, D. and Wu, Z. (2012) 'Graph-based workflow recommendation: on improving business process modeling', in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, ACM, pp.1527–1531.

Cao, B., Wang, J., Fan, J., Yin, J. and Dong, T. (2016) 'Querying similar process models based on the hungarian algorithm', *IEEE Transactions on Services Computing*, Vol. 10, No. 1, pp.121–135.

Deng, S., Wang, D., Li, Y., Cao, B., Yin, J., Wu, Z. and Zhou, M. (2016) 'A recommendation system to facilitate business process modeling', *IEEE Transactions on Cybernetics*, Vol. 47, No. 6, pp.1380–1394.

Dijkman, R., Dumas, M., Van Dongen, B., Käärik, R. and Mendling, J. (2011) 'Similarity of business process models: metrics and evaluation', *IS*, Vol. 36, No. 2, pp.498–516.

Fan, J., Wang, J., An, W., Cao, B. and Dong, T. (2017) 'Detecting difference between process models based on the refined process structure tree', *Mobile Information Systems*, Vol. 2017.

Fellmann, M., Zarvić, N. and Thomas, O. (2018) 'Business processes modeling recommender systems: user expectations and empirical evidence', *Complex Systems Informatics and Modeling Quarterly*, No. 14, pp.64–79.

Finis, J.P., Raiber, M., Augsten, N., Brunel, R., Kemper, A. and Färber, F. (2013) 'RWS-Diff: flexible and efficient change detection in hierarchical data', in *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, ACM, pp.339–348.

Huang, H., Peng, R. and Feng, Z. (2015) 'Efficient and exact query of large process model repositories in cloud workflow systems', *IEEE Transactions on Services Computing*.

Kluza, K., Baran, M., Bobek, S. and Nalepa, G.J. (2013) 'Overview of recommendation techniques in business process modeling', in *Proceedings of 9th Workshop on Knowledge Engineering and Software Engineering (KESE9)*, Citeseer, pp.46–57.

Kunze, M. and Weske, M. (2010) 'Metric trees for efficient similarity search in large process model repositories', in *International Conference on Business Process Management*, Springer, pp.535–546.

Li, Y., Cao, B., Xu, L., Yin, J., Deng, S., Yin, Y. and Wu, Z. (2013) 'An efficient recommendation method for improving business process modeling', *IEEE Transactions on Industrial Informatics*, Vol. 10, No. 1, pp.502–513.

Liu, C., Zeng, Q., Duan, H., Gao, S. and Zhou, C. (2019) 'Towards comprehensive support for business process behavior similarity measure', *IEICE Transactions on Information and Systems*, Vol. 102, No. 3, pp.588–597.

Sarwar, B., Karypis, G., Konstan, J., Riedl, J. et al. (2000) 'Analysis of recommendation algorithms for e-commerce', in *EC*, pp.158–167.

Tsai, C-F. (2012) 'Bag-of-words representation in image annotation: a review', *ISRN Artificial Intelligence*, Vol. 2012.

Vanhatalo, J., Völzer, H. and Koehler, J. (2009) 'The refined process structure tree', *Data & Knowledge Engineering*, Vol. 68, No. 9, pp.793–818.

Wang, J., Jin, T., Wong, R.K. and Wen, L. (2014) 'Querying business process model repositories'. *WWW*, Vol. 17, No. 3, pp.427–454.