



A Graph-based Method for Merging Business Process Models by Considering Semantic Similarity

Kelly Rossa Sungkono¹

Riyanarto Sarno^{1*}

Maisie Chiara Salsabila¹

Chintya Prema Dewi¹

¹*Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia*

* Corresponding author's Email: riyanarto@if.its.ac.id

Abstract: A process model describes business process flow as the activities that employees must carry out. Nowadays, many companies have similar business processes, so they do not establish their process model from scratch but build the model based on an existing process model or a combination of some process models. Several process mining methods approaches matching rules to define similarities of a model, and others consider the semantic side; however, none use the similarity to merge some business process models. This paper proposed graph-based semantic similarity, a method that merges two process models considering the semantic similarity between those activities. The utilized semantic similarity methods are SBERT and TF-IDF. The evaluations compare SBERT and TF-IDF with other methods and use a similarity method with the highest score in graph-based semantic similarity. Based on the semantic similarity score, graph-based semantic similarity with SBERT has higher similarity scores than existing graph-based semantic similarity, i.e., node similarity and Jaro-Winkler distance. With the highest similarity scores among existing methods, the evaluations also prove that graph-based semantic similarity with SBERT correctly combines business process models based on semantic similarity.

Keywords: Business process management, Business process model, Matching business process, Semantic similarity.

1. Introduction

The graph-based process model employs specific analytical procedures that facilitate data evaluation and analysis [1]. Furthermore, data from the event log is stored in a graph that is utilized to construct a process model to reduce conversion costs [2]. The definition of graph-based process discovery is a strategy for identifying, mapping and evaluating organizational processes [3]. Graph-based process discovery can assist businesses in detecting issues in organizational flow and identifying ways to improve management performance [4]. This procedure is also known as process mining [5, 6]. Process mining discovers, enhances, and optimizes the business processes of a company [1, 7]. A company does not establish an overall business process model from scratch. Instead, it builds the model by referring to an existing one or combining several. Constructing a combination of

models, called a generic process model, contingent on similarities of event labels.

There are existing studies that utilize similarity methods. In general, similarity methods are divided into three varieties: structural similarity [8, 9], behavioural similarity [10 – 13] and semantic similarity [14 – 16]. Besides those studies, some algorithms utilize graph-database to compare process models [17, 18]. Graph-database stores activities and their relationships directly, so the input data (event log) and the result can be in one platform. The drawback of several studies was only to find the score similarity of several activities and not utilize it to combine similar activities. The existing methods using a graph-database compare the names of activities based on the closeness of characters, not from a semantic point of view.

This research proposes graph-based semantic similarity, a method that utilizes graph database and natural language processing (NLP) method to

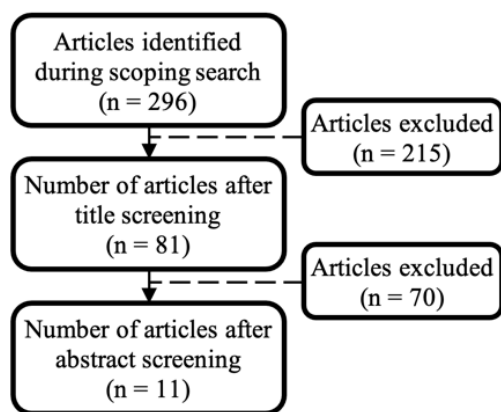


Figure. 1 Search strategy

merges two model processes contingent on the semantic similarity of those event labels. Various NLP approaches for measuring semantic similarity have been presented in recent years including cosine similarity [12, 19], term frequency–inverse document frequency (TF-IDF) [20, 21], bidirectional encoder representations from transformers (BERT) [22 – 24], and sentence bidirectional encoder representations from transformers (SBERT) [25].

The utilized similarity methods in graph-based semantic similarity are TF-IDF and SBERT. The TF-IDF method is a classic method that has been utilized for many years on NLP tasks. TF-IDF-based text similarity calculation approach maps passage to the access line area and transforms passage similarity into passage line gap [21]. SBERT is one of the most recent sentence embedding algorithms which evolved from the BERT approach. SBERT is a previous practice of BERT network modification which employs Siamese and triplet network architectures to produce closeness of relevant label embeddings for comparison [25]. Both TF-IDF and SBERT approaches calculate similarity by using cosine similarity formula.

This article is organized in such a way: Section 2 outlines the preliminary study, including a summary of available literature reviews and related works with the proposed method. The proposed graph-based semantic similarity details are declared in section 3, while the exploratory outcomes are highlighted in section 4. Lastly, section 5 contains the conclusion of the work and the discussion of the prospects.

2. Preliminaries

2.1 Search strategy

The search strategy is summarized in Fig. 1. Papers up to May 2022 are included in the search.

The outcomes are organized in succinct enclosing columns as review dimensions.

2.1.1. Search area

Determining a search area is for identifying and fine-tuning the search strategy and search phrases [18]. The purpose is to define search queries in ScienceDirect, the elected articles databases. Various search keywords are tested at this stage, including search terms from current literature reviews. Furthermore, multiple decisions are made along with the approach about which terms to include or reject from the query.

The highlighted terms in the search query are “business process model” and “similarity”. The terms “match” and “matching” were excluded since they returned irrelevant results.

2.1.2. Title and abstract screening

Firstly, articles are filtered based on the titles. The titles which mention business process” or “similarity” are considered. In this stage, 81 articles are selected. Then, 81 articles are reviewed based on their abstracts. The criteria considered during the screening are the primary focus of the paper is on matching databases such as data mining or matching between business processes, then papers about a novel approach to business process matching, and papers about clustering or classification of a set of databases.

2.1.3. Literature review

Following the screening, the final 11 articles for the review were chosen. Table 1 summarizes the review. Based on all previous studies in Table 1, studies using semantic similarity up to matching stage (not merging stage). Then, the existing graph-based method have not used semantic similarity to compare the activities.

2.2 Similarity metrics

There are three parameterized similarity metrics which are usually used in process model matching. The similarity metrics are intended to provide the answer to process model similarity queries. Table 2 presents examples of two business process models which are identical according to the similarity metric.

2.3 Semantic similarity

Words can be lexically and semantically similar. Words are lexically similar if their character

Table 1. Literature review

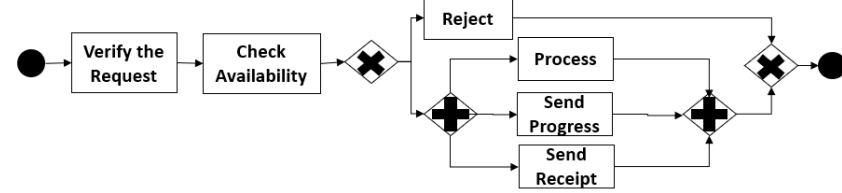
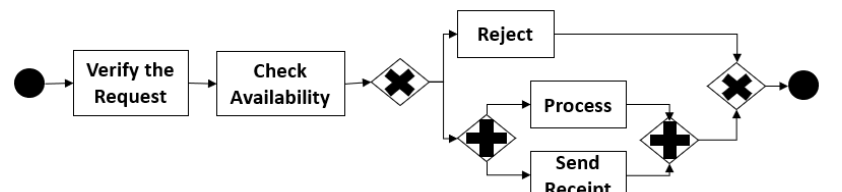
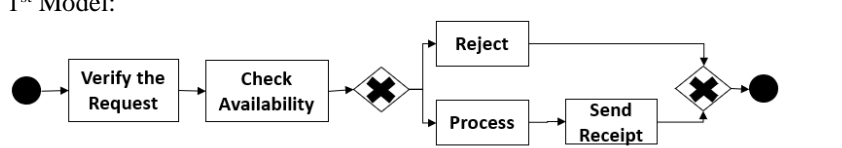
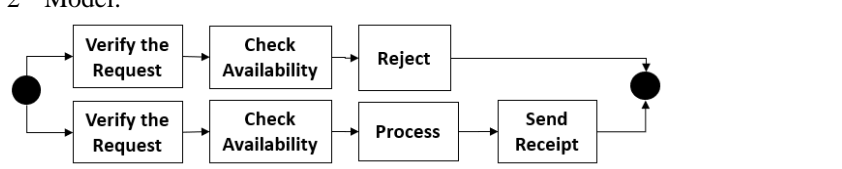
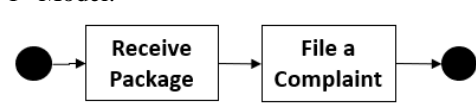
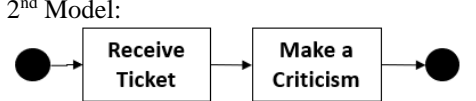
Paper Name	Similarity Type	Method	Contrast to this Study
Nuritha and Mahendrawati [9]	Structural		The structural similarity analyses the two operating models' comparability are depending on the graph architecture.
Nuritha [10], Estrada-Torres [11]	Behavioural		Behavioural similarity quantifies the similarity between two entities that may be related. The behavioural similarity between models is quantified using a metric. The metrics analyse the behaviour of models depending on the excerpt tree from the two operating models.
Jimenez-Molina [12], Bistarelli [13]	Behavioural, Semantic	Ontology	An ontology specifies data as procedural or declarative, implying that knowledge may support processes. Ontology-based behavioural similarity analyses and excerpts information of operating models to gather characteristics enclosed by event logs.
Zhou and others [14]	Semantic	WordNet	To encourage the restate or remodel of estate experimental procedures, this research developed a new crossing-workflow fragment discovery technique. BTM generates representative topics that measure the semantic applicability of events and procedure sections.
Shahzad and others [15]	Semantic	Word-embedding	The proposed method in this study used activity pair as an input and returns a compromise implying whether the knowledge combination is proportionate as an output. Essentially, the proposed method pairs the events labels and creates a sole point to reveal the closeness of inclined event sets.
Abdelakfi and others [16]	Semantic	Word2vec, Cosine Similarity	This paper proposed an agent-oriented method to portray cooperation of two human resources. The Classifier component of their proposed framework is restricted to WO sentence grouping. Word2vec is a neural-based model that predicts word relationships. Word2Vec is used to obtain distributed representations of every label surrounded by the corpus contents. The cosine similarity measure is used to generate synonymous words. This metric compares the similarity of non-zero two-word vectors.
Chang [17], Wang [18]	Contextual	Graph-database	Those papers implement their similarities method, i.e., node similarity and Jaro-Winkler Distance, in graph-database to measure the similarity of activities of process models. Those papers compare names of activities based on the closeness of characters, not from a semantic point of view.

sequences are similar. Semantically similar words have the same meaning, are opposites, are utilized in the equivalent approach, the ditto situation, and the variety of another [19]. Various natural language processing (NLP) tasks applies semantic similarity [20]. This study will employ semantic similarity to quantify the semantic similarity between the two operating models. Two text samples were classified similar in the early days if they included the same words or characters. The text was represented as real value vectors using approaches such as BoW and TF-IDF to help in the calculation of semantic similarity [20].

Semantic similarity algorithms often return a

rating or percentage of similarity instead of a binary choice of whether texts are similar or not. The terms semantic similarity and semantic relatedness are frequently interchanged. However, semantic relatedness takes a broader view, examining the shared semantic qualities of two words, in addition to accounting for semantic similarity across texts. For example, semantic matching type in Table 2, the phrases 'ticket' and 'locket' may be closely related, but they do not have the same semantic meaning, although the words 'ticket' and 'package' are. Thus, semantic similarity is one component of semantic relatedness [26].

Table 2. Type of process model matching

Type of Matching	Model
Structural	1 st Model: 
	2 nd Model: 
Behavioural	1 st Model: 
	2 nd Model: 
Semantic	1 st Model: 
	2 nd Model: 

2.4 Cosine similarity

Cosine similarity is a popular metric in information retrieval and related research. This metric represents a text document as a term vector. The similarity between two documents may be quantified using this approach by calculating the cosine point of the edge between the phrase lines of the two documents. The higher the similarity score between the term vectors of the document and the query, the more relevant the document and query are obtained [19]

Cosine similarity measurement between two-word vectors might produce misleading results when implemented syntactically [27]. The similarity between two vectors may be defined using vector similarity as Eqs. (1) and (2):

$$Sim(\vec{A}, \vec{B}) = \frac{\vec{A} \cdot \vec{B}}{|\vec{A}| |\vec{B}|} \tag{1}$$

$$Sim(\vec{A}, \vec{B}) = \frac{\sum_{k=1}^t w_{Ak} \times w_{Bk}}{\sqrt{\sum_{k=1}^t (w_{Ak})^2} \sqrt{\sum_{k=1}^t (w_{Bk})^2}} \tag{2}$$

The value of dimension is an occurrence of a term inside a document. A document can be described as a vector form as Eq. (3).

$$\vec{A} = w_{A1}, w_{A2}, \dots, w_{Ak} \tag{3}$$

As same as the document, the query of a term can be described as a vector form as presented in Eq. (4).

$$\vec{B} = w_{B1}, w_{B2}, \dots, w_{Bk} \quad (4)$$

where w_{Ak} and w_{Bk} ($0 < i < k$) are float values representing the density of each phrase found in a report. Every element of the line coincides to a phrase present in the document.

2.5 TF-IDF

Term frequency (TF) in Eq. (5) is the density with which a phrase occurs in a report, and the result is generally normalised to avoid bias toward a lengthier document [13].

$$TF_{d,t} = \frac{n_{d,t}}{N_d} \quad (5)$$

where $n_{d,t}$ represents the number of existences of the phrase t in report d , and N_d represents the value of phrase in report d .

Inverse document frequency (IDF) of a phrase in a text set reflects its relevance. IDF gives less weight to frequently occurring terms and more weight to infrequently occurring terms [21]. The following is Eq. (6) to calculate IDF.

$$IDF_t = \frac{D}{1 + |\{d_i | t \in d_i\}|} \quad (6)$$

where d_i represents the i th report, $|\{d_i | t \in d_i\}|$ represents the value of reports containing the term t , and D represents the value of reports. To avoid the denominator being 0, use $1 + |\{d_i | t \in d_i\}|$.

The TF-IDF algorithm [28], [29] is a frequently used analytical approach for extracting document feature terms. It primarily evaluates the importance of a term to document and document sets based on term frequency. It consists mostly of two phases: Term frequency (TF) and inverse document frequency (IDF) [30]. The classic term frequency inverse document frequency (TF-IDF) based document similarity calculation approach maps a document to the access line area and converts document similarity of report line gap [13]. Eq. (7) is used to calculate TF-IDF.

$$TF_{d,t} - IDF_t = TF_{d,t} \times IDF_t$$

$$TF_{d,t} - IDF_t = \frac{n_{d,t}}{N_d} \times \frac{D}{1 + |\{d_i | t \in d_i\}|} \quad (7)$$

TF-IDF did not affect the fact that terms may have diverse meanings and that various terms can be used to convey the same notion. Consider the words "Budi and Ani ate eggs and rice." and "Budi ate eggs and Ani ate rice." Although these two

statements include identical words, their meanings are not the same. Similarly, the phrases "Charlie is gluten intolerant." and "Charlie has celiac disease." express the same message but, the collection of terms is not the same. These approaches collected the lexical features of the document and were straightforward to apply; unfortunately, they neglected the semantic and syntactic aspects of the document [20]. Thus, TF-IDF is simple to use and computationally cheap.

2.6 Sentence-BERT (SBERT)

Bidirectional encoder representations from transformers (BERT) is a recent technique for communication modelling that is influenced by deep-learning algorithms and context-aware methodologies [22]. BERT is a pre-trained transformer network, which sets for various NLP tasks [23]. BERT encodes input as sub-words and learns sub-word embeddings. As a result, BERT is highly anisotropic. Lower layers create more context-specific representations than BERT. Increasing anisotropy is always associated with increased context-specificity [24].

Individual sentences are now being entered into BERT in recent studies to generate fixed-size sentence embeddings. The most typical method is to average the BERT output layer, which is referred to as BERT embeddings. SBERT was created to overcome this issue [25]. Sentence-BERT (SBERT) is a previous practice of BERT network modification that employs Siamese and triplet network architectures to produce semantically relevant label embeddings for comparison using cosine-similarity. It facilitates SBERT to be utilized for recent works that have been previously inapplicable for BERT. On advanced technologies, these similarity measurements may be done exceedingly efficiently, allowing SBERT to be utilized for both semantic similarity search and clustering.

To obtain a fixed-sized sentence embedding, SBERT conducts a pooling process on the BERT output. To fine-tune BERT, Siamese and triplet networks are created to update the weights to compare the resultant sentence embeddings using cosine similarity and ensure that they are semantically relevant. The network structure is determined by the training data provided. SBERT architecture is used in inference to compute similarity scores, for example. The cosine closeness of the two- label embeddings i and j are calculated (Fig. 2) [25].

2.7 Evaluation process

Activity correspondences between the first and the second business process model activity were manually identified. This assessment is required to establish whether the approach is suitable. The proposed method is then utilised to quantify the similarities between two business process model activities. The process results will be analysed to evaluate if they comply with the defined manual correspondence. If the correspondence results are correct, the similarity score of each correspondent becomes an evaluation metric for the accuracy of the proposed method. The optimum semantic similarity method will be determined based on the three factors.

3. Case study and proposed method

3.1 Case study

Event logs of first E-commerce and second E-commerce are used as a case study in this research. The event log will be processed in compliance with Section 3.2.

3.2 Flowchart

The flowchart of graph-based semantic similarity is shown in Fig. 3. This research is divided into three significant procedures: pre-processing procedure, semantic similarity procedure using natural language procedure (NLP) and merging process. Pre-processing and semantic similarity procedure is carried out using python, and then the merging process is conducted in Neo4j using cypher query language (CQL). There are two input data. Data are pre-processed to clean the data and then processed using NLP.

3.2.1. Pre-processing

Text pre-processing is needed to make data more structured and cleaner. Text pre-processing is the process of converting text from human language to machine-readable format for further process. There are several kinds of text pre-processing procedures. It is not necessary to do all of these all the time despite the importance of pre-processing. Thus, pre-processing procedures must be carefully selected and applied. Four pre-processing procedures have been chosen:

1. Case folding, which transforms words into a lower-case structure

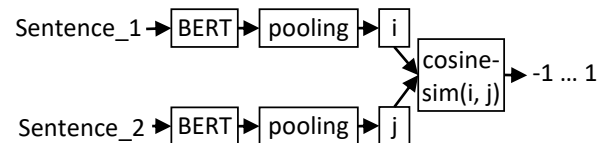


Figure. 2 SBERT Architecture

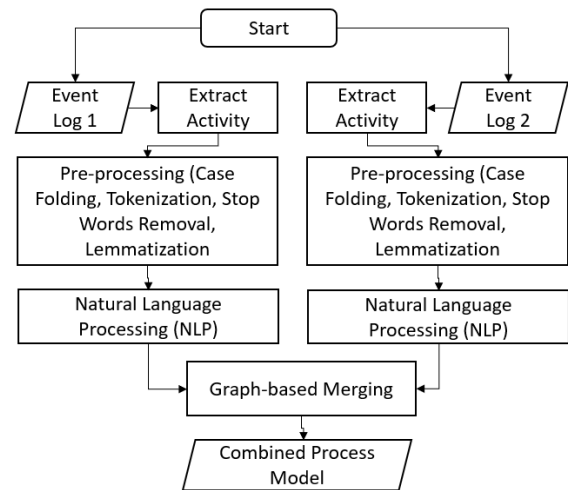


Figure. 3 Flowchart of graph-based semantic similarity

2. Tokenization, which splits up a larger body of text into words.
3. Stop words removal, which removes stop words such as 'and', 'it', 'in', etc.
4. Lemmatization, which rebounds the paltry or glossary scheme of a label.

3.2.2. Natural language processing (NLP)

Python dictionaries are made to store word-data values in word-key value pairs after pre-processing the data. Dictionaries are optimized to retrieve word values when the word key is known. Some complex correspondences are far from trivial. Furthermore, the identified complex correspondences are disputable. Dictionaries can assist in resolving the issue. The following process is sentence vectorizing in the TF-IDF method and sentence embedding using a pre-trained model for the SBERT method. The vectorized and embedded sentences are calculated using cosine similarity. Then, for each correspondence, the results will be compared to obtain the average similarity result.

3.2.3. Graph-based merging

Similarity score and the correspondence activity are inserted as two of the rule condition in the CQL rule as a part of the merging process. The constructed cypher rule algorithms are presented in Table 3. The similarity score mentioned in Algorithm 3 is obtained from the previous NLP procedure.

Table 3. Pseudocode of Graph-based Merging

Algorithm 1 : Convert Data Type Rule	
Input : <i>Activity₁ = Nodes activity of 1st model</i>	
1	Set <i>similarity score property data type in Activity₁ to float data type in new property</i>
Output : <i>Data type is converted</i>	
Algorithm 2 : Merging on The Same Graph	
Input : <i>A,D = Case activity nodes of the model B,C = Activity nodes of the model</i>	
1	Foreach <i>A, B, C, and D do</i>
2	if <i>property name of B==property name of A</i>
3	and <i>property name of C==property name of D</i>
4	and <i>property prep of C==property prep of B</i>
5	and <i>property name of C!=property name of B</i>
6	and <i>property id of D>property id of A</i>
7	call <i>the APOC merging function</i>
8	return <i>all</i>
Output : <i>Nodes with similar activity is merged</i>	
Algorithm 3 : Merging Different Graph	
Input : <i>A = Case activity nodes of the 1st model B = Activity nodes of the 1st model C = Case activity nodes of the 2nd model D = Activity nodes of the 2nd model</i>	
1	Foreach <i>A, B, C, and D do</i>
2	if <i>property name of B==property name of A</i>
3	and <i>property name of D==property name of C</i>
4	and <i>property correspondence activity of B==property prep of D</i>
5	and <i>property similarity score of B > similarity score</i>
6	call <i>the APOC merging function</i>
7	return <i>all</i>
Output : <i>Nodes with similar activity is merged</i>	

The similarity score mentioned in Algorithm 3 is obtained from the previous NLP procedure.

4. Evaluation

The correspondence between the two business process models is defined manually in advance, as stated in section 2.6. This assessment is required to establish whether the approach is suitable. Table 4 shows manually defined correspondences between two business process case studies. The correspondences are further manually categorized into three types: subject-predicate-object structured activities paired with predicate-object structured activities (Table 4(a)), predicate-object structured activities paired with subject-predicate-object structured activities (Table 4(b)), and both pairs are subject-predicate-object structured activities (Table 4(c)). This additional manually categorized assists in determining the accuracy of the results of the proposed method. The proposed method is then

Table 4. Correspondences between two business process (a) First E-Commerce PO and Second E-Commerce SPO activity label structures with different semantic, (b) First E-Commerce SPO and Second E-Commerce PO activity label structure with different semantic, and (c) First E-Commerce SPO and Second E-Commerce SPO activity label structure with different semantic

(a)	
First E-Commerce	Second E-Commerce
Choose Flight Ticket	Buyer Chooses Products
Choose Payment Method	Buyer Chooses Payment Method
Receive E-Ticket	Buyer Receives Package
(b)	
First E-Commerce	Second E-Commerce
User Makes a Complaint	File a Complaint
User Requests a Refund	Request a Refund Purchase
User Requests Reschedule	Request a Return Product
User Receives Complaint Request Confirmation	Confirm Complaint Request
(c)	
First E-Commerce	Second E-Commerce
User Receives Refund	Buyer Receives Refund
User Reschedules the Flight	Buyer Returns Product
User Gives Feedback	Buyer Gives Rating
	Buyer Writes Review

Table 5. Semantic similarity result

Algorithms	Similarity Result		
	Instance 4(a)	Instance 4(b)	Instance 4(c)
Graph-based Semantic Similarity with TF-IDF	1	0.707	1
Graph-based Semantic Similarity with SBERT (Paraphrase-MiniLM-L6-v2 model)	1	0.908	1
Graph-based Semantic Similarity with SBERT (all-MiniLM-L6-v2 model)	1	0.882	1
Node Similarity in Neo4j [17]	0	0	0
Jaro-Winkler Distance in Neo4j [18]	0.498	0.175	0.560

utilised to compute the similarities between two business process model activities.

Several similarity algorithms were applied and

tested as shown in Table 5. The proposed algorithms, a graph-based semantic similarity with TF-IDF and a graph-based semantic similarity with SBERT, are compared with existing graph methods, i.e., Node Similarity [17] and Jaro-Winkler Distance [18].

The similarity score in Table 5 was determined by analysing the average similarity score among correspondences in Table 4. The TF-IDF approach generates the lowest score among the proposed methods. It is because TF-IDF is a bag-of-word (BoW) based technique, so it does not capture semantics and it is not contextual sensitive [11]. Consequently, the result is not as accurate as an embedding-based method. The node similarity procedure available in Neo4j generates a score of 0 because the results are randomly paired, and each pair returns integer scores only. Jaro-Winkler distance approach that ran in Neo4j generates a higher score than Node Similarity but lower than the score from TF-IDF and SBERT approaches.

Similarity results are then exported as new columns in the event log for the merging process of graph-based semantic similarity. NLP procedures and merging are conducted in different tools because NLP procedures that were conducted using python on different tools gave better results than NLP procedures that were provided or performed using proposed CQL in Neo4j. It is essential because a higher semantic similarity score is the main parameter of this study. Thus, graph-based semantic similarity with SBERT is the best method applied to this study case. Figs. 4, 5, and 6 present the generic process models constructed by graph-based semantic similarity based on case Tables 4(a), 4(b) and 4(c), respectively. The results prove that graph-based semantic similarity with SBERT can construct generic process models based on semantic similarity.

5. Conclusion

The proposed graph-based semantic similarity merges several process models based on semantic similarity to provide a generic process model. TF-IDF and SBERT are semantic similarity methods presented in this study.

The evaluation uses e-commerce business processes, i.e., first E-commerce and second E-commerce, to determine the best semantic similarity applied in graph-based semantic similarity. There are three evaluation cases. First is the subject-predicate-object structured activities of first E-commerce paired with predicate-object structured activities of second E-commerce (FirstSPO-Second PO). The second is predicate-object structured activities of first E-commerce paired with subject-predicate-object of second E-commerce (FirstPO-SecondSPO). The last is that both pairs are subject-predicate-object structured activities (FirstSPO-SecondSPO).

In the first and third cases, TF-IDF and SBERT have the highest similarity among other methods; however, SBERT has the highest score in the second case. It is because SBERT is not a BoW-based technique like TF-IDF, so it does capture semantics and is contextually sensitive. Then, the graph-based semantic similarity with SBERT is compared with existing graph-based similarity, i.e, node similarity and Jaro-Winkler distance. Graph-based semantic similarity with SBERT has higher similarity scores than existing graph-based semantic similarity, i.e., Node similarity and Jaro-Winkler distance. With the highest similarity scores among existing methods, the evaluations also prove that graph-based semantic similarity with SBERT correctly combines business process models based on semantic similarity to construct the generic process model.

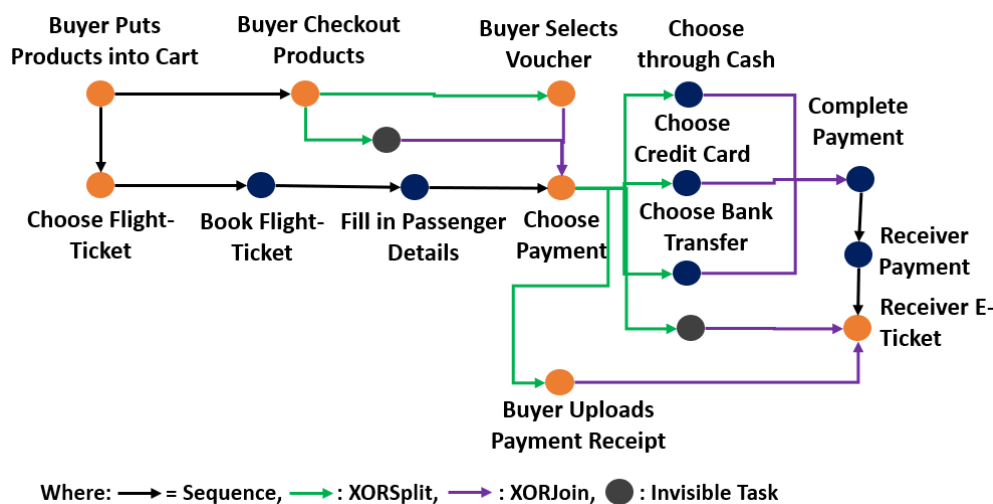


Figure. 4 Merging process result case Table 4(a)

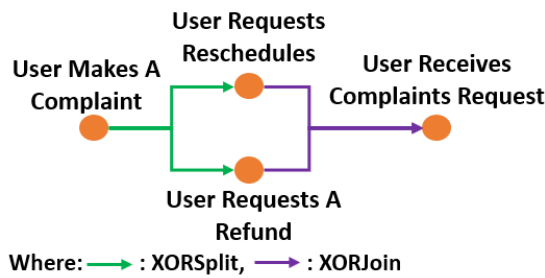


Figure. 5 Merging process result case Table 4(b)

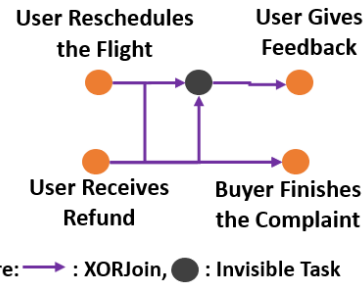


Figure. 6 Merging process result case Table 4(c)

This study would be a helpful starting point for researchers to find novel approaches to quantify semantic similarity in business process models and discover a novel approach to merging relations and nodes simultaneously.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Supervision, RS; Conceptualization, KRS; methodology, MCS and KRS; formal analysis: MCS and CPD; writing- original draft preparation: MCS and CPD; writing- review and editing: MCS and KRS.

Acknowledgements

The authors gratefully acknowledge financial support from the Institut Teknologi Sepuluh Nopember for this work, under project scheme of the Publication Writing and IPR Incentive Program (PPHKI) 2022, from Lembaga Pengelola Dana Penelitian (LPDP) under Riset Inovatif-Produktif (RISPRO) Program, and from the Indonesian Ministry of Education and Culture under Penelitian Terapan Unggulan Perguruan Tinggi (PTUPT) Program.

References

- [1] R. Sarno, K. R. Sungkono, and R. Septiarakhman, "Graph-Based Approach for Modeling and Matching Parallel Business Processes", *International Information Institute (Tokyo). Information*, Vol. 21, No. 5, pp. 1603-1614, 2018.
- [2] W. M. P. V. D. Aalst, "Using free-choice nets for process mining and business process management", In: *Proc. of 2021 16th Conference on Computer Science and Intelligence Systems (FedCSIS)*, pp. 9–15, 2021.
- [3] K. R. Sungkono, A. S. Ahmadiyah, R. Sarno, M. F. Haykal, M. R. Hakim, B. J. Priambodo, M. A. Fauzan, and M. K. Farhan, "Graph-based Process Discovery containing Invisible Non-Prime Task in Procurement of Animal-Based Ingredient of Halal Restaurants", In: *Proc. of 2021 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*, Apr. 2021, pp. 134–140. doi: 10.1109/APWiMob511111.2021.9435261.
- [4] M. Z. N. Maulana, R. Sarno, and K. R. Sungkono, "Process Discovery of Collaboration Business Process Containing Invisible Task in Non-Free Choice by using Modified Alpha++", In: *Proc. of 2021 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*, pp. 73–79, 2021.
- [5] A. Augusto, M. Dumas, and M. L. Rosa, "Automated discovery of process models with true concurrency and inclusive choices", In: *Proc. of International Conference on Process Mining*, pp. 43–56, 2021.
- [6] K. R. Sungkono, U. E. N. Rochmah, and R. Sarno, "Heuristic linear temporal logic pattern algorithm in business process model", *International Journal of Intelligent Engineering and Systems*, Vol. 12, No. 4, pp. 31–40, 2019, doi: 10.22266/ijies2019.0831.04.
- [7] K. R. Sungkono and R. Sarno, "Constructing Control-Flow Patterns Containing Invisible Task and Non-Free Choice Based on Declarative Model", *International Journal of Innovative Computing, Information and Control (IJICIC)*, Vol. 14, No. 4, 2018.
- [8] I. G. Anugrah and R. Sarno, "Business Process model similarity analysis using hybrid PLSA and WDAG methods", In: *Proc. of 2016 International Conference on Information Communication Technology and Systems (ICTS)*, pp. 231–236, 2016, doi: 10.1109/ICTS.2016.7910304.
- [9] I. Nuritha and E. R. Mahendrawathi, "Structural similarity measurement of business process model to compare heuristic and inductive miner algorithms performance in dealing with noise", *Procedia Computer*

- Science*, Vol. 124, pp. 255–263, 2017.
- [10] I. Nuritha and E. R. Mahendrawathi, “Behavioural similarity measurement of business process model to compare process discovery algorithms performance in dealing with noisy event log”, *Procedia Computer Science*, Vol. 161, pp. 984–993, 2019.
- [11] B. E. Torres, M. Camargo, M. Dumas, L. G. Bañuelos, I. Mahdy, and M. Yerokhin, “Discovering business process simulation models in the presence of multitasking and availability constraints”, *Data & Knowledge Engineering*, Vol. 134, p. 101897, 2021.
- [12] A. J. Molina, J. G. Villegas, and J. Fuentes, “ProFUSO: Business process and ontology-based framework to develop ubiquitous computing support systems for chronic patients’ management”, *Journal of Biomedical Informatics*, Vol. 82, pp. 106–127, 2018.
- [13] S. Bistarelli, T. D. Noia, M. Mongiello, and F. Nocera, “Pronto: an ontology driven business process mining tool”, *Procedia Computer Science*, Vol. 112, pp. 306–315, 2017.
- [14] Z. Zhou, J. Wen, Y. Wang, X. Xue, P. C. K. Hung, and L. D. Nguyen, “Topic-based crossing-workflow fragment discovery”, *Future Generation Computer Systems*, Vol. 112, pp. 1141–1155, 2020.
- [15] K. Shahzad, S. Kanwal, K. Malik, F. Aslam, and M. Ali, “A word-embedding-based approach for accurate identification of corresponding activities”, *Computers and Electrical Engineering*, Vol. 78, pp. 218–229, 2019, doi: 10.1016/j.compeleceng.2019.07.011.
- [16] M. Abdelakfi, N. Mbarek, and L. Bouzguenda, “Mining Organizational Structures from Email Logs: an NLP based approach”, *Procedia Computer Science*, Vol. 192, pp. 348–356, 2021.
- [17] V. Chang, Y. K. Songala, Q. A. Xu, and B. S. C. Liu, “Scientific Data Analysis using Neo4j”, In: *Proc. of 4th International Conference on Finance, Economics, Management, and IT Business*, pp. 75–84, 2022.
- [18] Y. Wang, J. Qin, and W. Wang, “Efficient approximate entity matching using jaro-winkler distance”, In: *Proc. of International Conference on Web Information Systems Engineering*, pp. 231–239, 2017.
- [19] W. H. Gomaa, A. A. Fahmy, and others, “A survey of text similarity approaches”, *International Journal of Computer Applications*, Vol. 68, No. 13, pp. 13–18, 2013.
- [20] D. Chandrasekaran and V. Mago, “Evolution of semantic similarity—a survey”, *ACM Computing Surveys (CSUR)*, Vol. 54, No. 2, pp. 1–37, 2021.
- [21] J. Wang, W. Xu, W. Yan, and C. Li, “Text similarity calculation method based on hybrid model of LDA and TF-IDF”, In: *Proc. of the 2019 3rd International Conference on Computer Science and Artificial Intelligence*, pp. 1–8, 2019.
- [22] M. M. M. Gniewkowski, T. Walkowiak, and M. Błedkowski, “Text document clustering: Wordnet vs. TF-IDF vs. word embeddings”, In: *Proc. of the 11th Global Wordnet Conference*, pp. 207–214, 2021.
- [23] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding”, *arXiv Preprint arXiv:1810.04805*, 2018.
- [24] K. Ethayarajh, “How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings”, *arXiv Preprint arXiv:1909.00512*, 2019.
- [25] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence embeddings using siamese BERT-networks”, In: *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pp. 3982–3992, 2019, doi: 10.18653/v1/d19-1410.
- [26] M. A. H. Taieb, T. Zesch, and M. B. Aouicha, “A survey of semantic relatedness evaluation datasets and procedures”, *Artificial Intelligence Review*, Vol. 53, No. 6, pp. 4407–4448, 2020.
- [27] F. Rahutomo, T. Kitasuka, and M. Aritsugi, “Semantic cosine similarity”, In: *Proc. of the 7th International Student Conference on Advanced Science and Technology ICAST*, Vol. 4, No. 1, p. 1, 2012.
- [28] S. Robertson, “Understanding inverse document frequency: on theoretical arguments for IDF”, *Journal of Documentation*, Vol. 60, No. 5, pp. 503–520, 2004.
- [29] W. Zhang, T. Yoshida, and X. Tang, “A comparative study of TF* IDF, LSI and multi-words for text classification”, *Expert Systems with Applications*, Vol. 38, No. 3, pp. 2758–2765, 2011.
- [30] D. Kim, D. Seo, S. Cho, and P. Kang, “Multi-co-training for document classification using various document representations: TF-IDF, LDA, and Doc2Vec”, *Information Sciences*, Vol. 477, pp. 15–29, 2019.