Process Model Similarity Techniques for Process Querying

Andreas Schoknecht, Tom Thaler, Ralf Laue, Peter Fettke, and Andreas Oberweis

Abstract Organizations store hundreds or even thousands of models nowadays in business process model repositories. This makes sophisticated operations, like conformance checking or duplicate detection, hard to conduct without automated support. Therefore, querying methods are used to support such tasks. This chapter reports on an evaluation of six techniques for similarity-based search of process models. Five of these approaches are based on Process Model Matching using various aspects of process models for similarity calculation. The sixth approach, however, is based on a technique from Information Retrieval and considers process models as text documents. All the techniques are compared regarding different measures from Information Retrieval. The results show the best performance for the non-matching-based technique, especially when a matching between models is difficult to determine.

1 Introduction

Companies and other organizations own lots of business process models and store them in so-called business process model repositories to describe and structure their business operations. These repositories can contain hundreds, or even thousands, of models (see, e.g., the collections mentioned in [14] and [25]), which makes sophisticated operations like conformance checking, duplicate detection, or the reuse of

A. Schoknecht (✉) · A. Oberweis
Institute AIFB, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
e-mail: andreas.schoknecht@kit.edu; andreas.oberweis@kit.edu

T. Thaler · P. Fettke
German Research Center for Artificial Intelligence (DFKI) and Saarland University, Saarbrücken, Germany
e-mail: tom.thaler@dfki.de; peter.fettke@dfki.de

R. Laue
University of Applied Sciences Zwickau, Zwickau, Germany
e-mail: ralf.laue@fh-zwickau.de

© Springer Nature Switzerland AG 2022

A. Polyvyanyy (ed.), *Process Querying Methods*,
https://doi.org/10.1007/978-3-030-92875-9_16

(parts of) models hard to conduct without automated support. Therefore, querying methods are used, for instance, to detect duplicate model fragments automatically.

Process model querying methods like the ones described in various chapters of this book can be used to find models containing a specified model fragment. This chapter, however, focuses on a different kind of querying approach, which is called *similarity-based search* [8]. When using similarity-based search, a process model is used as a query with the intention to find similar models in a repository.

Many similarity-based search techniques have been published (see, e.g., the survey in [24] for an overview). These techniques can be classified into two categories. Approaches from the first category are based on an underlying alignment between the activities or other nodes of the compared process models, which is also called *Process Model Matching* [2]. Before calculating a final similarity value, these approaches require an alignment between the model nodes. Techniques from the second category do not require such an alignment but use other means like process model metrics or document vectors created from the textual content of models.

This chapter provides an assessment of the performance of six techniques for similarity calculation of process models in the context of similarity-based search. Thereby, we extend our analysis described in [28] by comparing the matching-based similarity approaches with the *LS3* technique [23], which does not require a matching of process models for determining similarity values. Besides, we discuss further evaluation results of these approaches regarding essential measures from the Information Retrieval area such as Precision, Recall, F-Measure, and R-Precision [16].

The rest of this chapter is organized as follows: Sect. 2 provides fundamental definitions that are necessary to understand the subsequent sections. Afterward, we discuss related work and the relation of process model similarity to process querying in Sect. 3. The compared similarity techniques are then presented in Sect. 4. The setup of the comparative evaluation, the results of the evaluation, and the limitations of our analysis are discussed in Sect. 5. Finally, Sect. 6 provides a conclusion of this chapter and an outlook on future research.

2 Foundations

Fundamentals regarding business process models and the calculation of similarity values for process models are introduced in this section. First, Sects. 2.1 and 2.2 introduce definitions for process models and process model instances, respectively. Afterward, Sect. 2.3 describes Process Model Matching, which is an essential part for calculating a similarity value in most existing process model similarity techniques. We examine process model similarity in detail in Sect. 2.4. Finally, Sect. 2.5 provides background on the measures used in our evaluations.

2.1 Business Process Model

Similarity measurement in the context at hand primarily focuses on business process models, where it is distinguished between informal, semi-formal, and formal representations [5]. The models of interest typically have semi-formal or formal characteristics and are mostly represented as EPCs [10], BPMN diagrams, [19] or Petri Nets [18]. However, a business process model should not be understood as a model of a particular modeling language, but as a model of a particular model class describing business processes. Hence, an abstract definition of a process model covering the wide range of existing modeling languages is needed as the foundation. This definition requires an adequate generic representation of the graph structure and labeled nodes, as these are essential components of existing similarity measures.

Several generic formalizations of business process models are proposed in the literature, which generally address specific intentions. An analysis of these formalizations is described in [24], which resulted in the following definition.

Definition 2.1 (Business Process Model) A business process model $M = (N, A, L, \lambda)$ is a directed graph consisting of three sets N, A, and L and a partial function $\lambda : N \rightarrow L$ such that

- $N = F \cup E \cup C$ (F, E, C pairwise disjoint) is a finite non-empty set of nodes with
  - $F \subseteq N$: a finite non-empty set of activities (also called functions, transitions, tasks)
  - $E \subset N$: a finite set of events
  - $C \subset N$: a finite set of connectors (also called gateways)
- $A \subseteq N \times N$ is a finite set of directed arcs (also called edges) between two nodes $n_i, n_j \in N$ defining the sequence flow.
- L is a finite set of textual labels.
- λ assigns to each node $n \in N$ a textual label $l \in L$.

Although further node types such as organizational units and resources are relevant for describing business processes, they only play a minor role for existing similarity measurement. Hence, in this work, we abstract from them.

2.2 Business Process Instances

While business process models describe a business process on an abstract level, a business process instance represents an execution of a business process. An execution can either be observed in the real world or simulated. Business process instances are typically described as so-called traces (cf. [4]).

Definition 2.2 (Trace, Trace Length) A trace σ of a process model $M = (N, A, L, \lambda)$ is a valid sequence of activities from F. A trace denotes the order in which the activities are executed. It is written as $\sigma =< f_1, \ldots, f_i, \ldots, f_n >$,

where $1 \leq i \leq n$. f_i may be equal to f_j with $i \neq j$ as it is possible that an activity occurs more than once in a trace. The length of a trace $|\sigma|$ is the number of activities in the trace.

Note that the term valid trace means that a trace cannot contain any sequence of activities but only sequences which can actually be executed, i.e., which are allowed by the semantics of the process model.

2.3 Business Process Model Matching

Structural correspondences of model elements are often the basis for calculating the similarity between business process models. In that sense, matching describes the procedure of taking two models as input, referred to as the source and target, and producing a number of matches between the elements of these two models as output based on a particular correspondence notion [21].

The more specific term Process Model Matching refers to the matching of single nodes, sets of nodes, or node blocks of one process model to corresponding elements of another process model based on criteria like similarity, equality, or analogy [26]. Referring to [31], it is generally distinguished between elementary and complex node matches, which are defined as follows:

Definition 2.3 (Elementary/Complex Node Match) A match m is denoted by a tuple (N_1, N_2) of two sets of nodes. A match (N_1, N_2) is called elementary match iff $|N_1| = |N_2| = 1$ and complex match iff $|N_1| > 1 \vee |N_2| > 1$.

There are various approaches that approximate correspondences, respectively matches, between (sets of) nodes of models. A common technique is the consideration of (normalized) edit distances [7] of node labels like the Levenshtein distance [15]. Other approaches described in [2, 3] additionally apply techniques from the area of Natural Language Processing (NLP), thereby taking into account, e.g., semantic information of node labels concerning synonyms, homonyms, and antonyms.

2.4 Business Process Model Similarity

Similarity measures quantify the similarity between business processes models, while similarity is interpreted in different manners. Several dimensions of similarity have been identified and studied in the literature, e.g., the graph structure and state space of a process model, the syntax and semantics of process model labels, the behavior of a process or the similarity perceived by a human, as well as combinations of these dimensions [24].

Independently from the interpretation of similarity, a similarity value is usually expressed either on an interval or on a ratio scale. This provides the frame for a typical operationalization of business process model similarity in a metric space. Such a metric fulfills the properties of non-negativity, symmetry, identity, and triangle inequality [33]. However, as shown in [11], most of the existing process model similarity measures do not fulfill the abovementioned properties. Depending on the similarity measurement objective, there might be good reasons for violating particular properties. For example, if a similarity measure is used for searching process models, it might be acceptable to violate the symmetry property.

In the specific "part-of search" scenario, the search query would be a process model fragment. The similarity value should be one iff a process model contains the query fragment. On the contrary, when interchanging the query fragment and the process model containing the fragment, the resulting similarity value should be lower. Essentially, fulfilling the symmetry property is not a necessary requirement for that application.

2.5 Evaluation Measures

For the evaluation of the similarity-based search techniques presented in Sect. 5, Precision, Recall and F-Measure are used. Precision is defined as the fraction of relevant and obtained results (true positives TP) to all obtained results (B), Recall is defined as the fraction of relevant and obtained results to all relevant results (A), and F-Measure is defined as the harmonic mean of Precision and Recall. Formally, these values are calculated as follows:

$$P = \frac{|TP|}{|B|}, \quad R = \frac{|TP|}{|A|}, \quad F = 2 \cdot \frac{P \cdot R}{P + R}.$$

In addition, we calculated R-Precision and Precision-at-k values to evaluate ranked retrieval results. R-Precision measures Precision for a query with respect to the first $|A|$ models, whereby $|A|$ is the amount of relevant results only. R-Precision is therefore defined as the fraction $\frac{|TP|}{|A|}$ with $|TP|$ being the relevant and obtained documents. The difference to Recall is that not all retrieved results are taken into account, but only the $|A|$ highest ranked results. Precision-at-k does not use the $|A|$ highest ranked models but considers the first k models instead. Hence, the following fraction is calculated: $\frac{|TP|}{k}$, again with $|TP|$ being the relevant and obtained documents. For further details on all the used evaluation measures, we refer the reader to [16].

3 Process Model Querying and Similarity-Based Search

Process model querying approaches and similarity-based search techniques pursue the same goal: to provide users with a search functionality to satisfy their information needs more efficiently compared to manual browsing of model repositories. But while the goal may be the same, the used means are different. Process model querying approaches provide some kind of query language, which can be used to describe queries. These queries represent conditions which must be fulfilled by models from a repository to be contained in the query result.

Some query languages allow to find possible execution traces through textual query formulation, e.g., models that allow the execution of activity B after activity A. Other query languages allow for the graphical modeling of queries comparable to process modeling itself. In this context, a query is represented as a model fragment, which must be contained in a model to be returned as a query result. Typically, these query languages provide means to increase the variability of query formulation with special query elements like a path connector or wildcard nodes. Finally, some query languages incorporate Process Model Matching to widen the search scope of the queries.

Instead, similarity-based search uses an existing process model as a query and returns all models from the repository which have a similarity value with the query above a certain threshold. Therefore, similarity measures on process models are required to apply similarity-based search. Besides, most of the similarity-based approaches use Process Model Matching as the foundation for similarity calculation [24].

When comparing process model querying with similarity-based search, their commonality is the basic idea of providing users with search functionalities for process model repositories. Additionally, both approaches can rely on Process Model Matching for finding suitable query results. The main difference, however, is their search approach. While querying techniques use specific *query languages* to formulate a query, an *existing process model* is used as query input in the similarity-based search. Furthermore, querying techniques typically do not apply similarity measures on process models to widen the search scope but use other means like wildcard nodes.

Furthermore, similarity-based search can be related to the *Process Querying Framework* described in [20]. Similarity-based search for process models also requires some kind of process model repository for determining query results. Similarity-based search techniques require that such repositories contain business process models as one specific kind of behavior models mentioned in [20]. Additionally, for some techniques, other behavior models like event logs, execution traces, or alignments might be required or must be computed from the process models. A query itself is composed of a process model for which similar models should be detected in a repository. Besides a query model, it can be useful to provide a threshold value for specifying how similar resulting models should be compared to the query model. The intent of a query is always the same: retrieving

similar models. Hence, similarity-based search is not geared toward manipulating or deleting models.

Regarding the *Prepare* part of the Process Querying Framework, the performance of similarity-based search might be increased by indexing or caching mechanisms. For example, the efficiency of queries with the document vector-based LS3 approach [23] is increased when the document vectors of process models are stored in an index. In this case, the document vector generation has to be performed only for the query model. The document vectors for all models from the repository can be retrieved from the index and do not have to be generated for each query.

Yet, not all techniques might be equally well supported by the index structures. The calculation of matches between a query model and the models from the repository are not as easily indexable or cacheable as the document vectors from the previous example. This is due to the fact that the calculation of matches is always dependent on the query model and the possible result models. One difficulty is, for instance, that the matches between a query model and one process model from a repository cannot be used to infer matches between the query model and another process model without additional computations. The same applies for a new query model. Even if matches between other query models and the models from the repository are known, it is not possible to use these directly due to different terminologies in labels or model structures.

With this in mind, it is also difficult to envision a filter mechanism for matching-based similarity techniques, which can be used in the *Execution* part of the Process Querying Framework. If, for example, two process models pm_1 and pm_2 from a repository only have a low similarity score for a specific matching-based similarity technique and if the similarity value between a query model qm and pm_1 is also low, pm_2 cannot automatically be excluded from similarity calculation, i.e., pm_2 cannot be filtered. The reason is again that matching-based similarity techniques highly depend on the calculated matches. For the LS3 approach, however, a filtering of results could be applied based on the angles between the document vectors. For the two example models mentioned above, pm_2 could be filtered from similarity calculation if the angle between qm and pm_1 is too big and the angle between qm and pm_2 would be even bigger.

4 Selection of Similarity Techniques

In order to evaluate the practical applicability and the limitations of the current state of research for similarity search, we need to identify and select proper similarity measurement techniques. This selection is based on the findings in [28]. As the analysis in [28] showed, most similarity techniques produce highly correlating similarity values. Hence, we only compare five of the eight approaches. The other three approaches were not considered, since they already showed a very high correlation with at least one of the selected ones. The selected techniques [1, 9, 12, 29, 32] differ in the dimensions used for similarity calculation and in their complexity so that the

Table 1 Functional characteristics of all compared techniques

Dimension/reference	LS3 [23]	SSCAN [1]	CF [29]	FBSE [32]	LAROSA [12]	LCST [9]
Natural Lang.: Syntax		x	x	x	x	x
Natural Lang.: Semantics	x				x	
Graph structure		x	x	x	x	
Behavior			x			x
Model as text	x					
Model as element labels		x	x	x	x	x

selection should provide for a differentiated evaluation in the similarity-based search context. All of these similarity approaches use matches to calculate the similarity of process models. Therefore, we intentionally included another technique [23] in the evaluation, which does not use matches for similarity calculation.

Table 1 contains an overview of the techniques used in the evaluation. The calculation and setup details are described in the following subsections.

4.1 Latent Semantic Analysis-Based Similarity Search

The *Latent Semantic Analysis-Based Similarity Search* (LS3) approach [23] is based on Latent Semantic Analysis [13], which is a technique from the Information Retrieval area for searching similar documents. The basic idea of LS3 is to construct so-called document vectors from process models. These document vectors form a Term-Document Matrix, in which each column represents a process model, i.e., a document vector, and each row represents a term[1] from all process models in a repository. The entries of the matrix contain weighted frequency values describing the weight of a certain term in a specific model.

Afterward, singular value decomposition is applied to decompose the constructed Term-Document Matrix of the process model repository into the product of three matrices. These matrices are used to construct another matrix with reduced dimensionality. The document vectors in the reduced matrix span a vector space which is used for calculating the similarity of process models. The similarity of two process models is thereby calculated as the cosine of the angle between their document vectors.[2] We did not include a classical Information Retrieval approach in our comparison as LS3 performed better in an experimental evaluation [22].

[1] In this context, a term should be understood as a word or a meaningful unit of words (e.g., statue of liberty).

[2] For calculating the similarity values, we used the code available at https://github.com/ASchoknecht/LS3.

4.2 Similarity Score Based on Common Activity Names

The similarity of two process models according to [1] (SSCAN) is calculated based on the number of identically labeled activities. We used the implementation proposed in the RefMod-Miner[3] to determine similarity values.

4.3 Causal Footprints

In the approach from [29] (CF), each process model is transformed into a so-called footprint vector, and the similarity of two models is determined as the cosine of the angle of their footprint vectors. A footprint vector consists of the activities of the process model as well as of two behavioral relations for each activity. The first relation contains all activities that are executed before an activity and the second relation contains all activities that are executed after that activity.

Hence, calculating the causal footprints requires a node matching between two process models. Although there is a proposal of a semantic node similarity measure, the used implementation from http://rmm.dfki.de considers two activities as a match if both have the same label.

4.4 Feature-Based Similarity Estimation

The technique described in [32] (FBSE) uses the syntactical natural language dimension as well as the graph-structural dimension to determine similarity values. Regarding the syntactical dimension, a Levenshtein distance-based similarity value between activity labels is calculated. For the graph-structural dimension, five roles (start, stop, split, join, and regular) are used to characterize an activity. The graph-structural similarity is then based on the common roles of two activities (so-called role feature similarity). Two activities are considered as equivalent if both the syntactic label similarity and the role feature similarity surpass an individual threshold. Finally, the similarity between two process models is defined as the ratio of equivalent activities to the overall number of activities in both models.

We used the implementation from http://rmm.dfki.de to determine similarity values. Thereby, the thresholds were set as proposed in the original paper, and the resulting similarity matrix was optimized using the greedy algorithm described in [32].

[3] RefMod-Miner as a Server: http://rmm.dfki.de and Code on GitHub: https://github.com/tomson2001/refmodmine.

4.5 La Rosa Similarity

The similarity calculation of [12] (LAROSA) is based on the graph-edit distance similarity described in [6]. The basic idea of the technique is to determine matches between two process models and to additionally consider the graph structure of models by calculating a graph-edit distance. The matches in [12] are based on the Levenshtein distance of the node labels and on a linguistic similarity measure using a lexical database. The greedy algorithm in [6] for finding the optimal graph-edit distance has been used with the original implementation. The parameter values were set as described in [12].

4.6 Longest Common Sets of Traces

The approach proposed in [9] (LCST) uses the traces of two process models M_1 and M_2 to quantify their similarity. Therefore, the two components trace compliance degree $cd_{trace}(\sigma_1, \sigma_2)$ and trace maturity degree $md_{trace}(\sigma_1, \sigma_2)$ are used, whereby σ_1 is a trace of M_1 and σ_2 is a trace of M_2. The trace compliance degree covers the extent to which a process adheres to ordering rules of activities, while the trace maturity degree covers the extent to which the activities of the other model are recalled. Both components are defined based on the length of their longest common subsequence lcs, such that $cd_{trace}(\sigma_1, \sigma_2) = \frac{|lcs(\sigma_1,\sigma_2)|}{|\sigma_2|}$ and $md_{trace}(\sigma_1, \sigma_2) = \frac{|lcs(\sigma_1,\sigma_2)|}{|\sigma_1|}$. Based on that, the compliance and maturity degree between two process models are defined as the sum of the maximum trace compliance and trace maturity degrees. Finally, two components are used to express in how far the traces of one model are reflected by the traces of another model.

To provide a comparable similarity value, the average of both components is calculated and interpreted as the final similarity value. The matches required by this approach are determined using the Levenshtein distance-based similarity calculation between two activity labels with a minimum threshold of 0.9. We used the implementation from http://rmm.dfki.de to determine similarity values.

5 Evaluation

The selected process model similarity measurement techniques are evaluated in this section. First, we present the used data collection (Sect. 5.1) and describe the evaluation design (Sect. 5.2). Afterward, the evaluation results are presented in Sect. 5.3, followed by a discussion of the results and the limitations in Sect. 5.4.

5.1 Dataset

The dataset for the comparison is based on the model collection used in [28]. The idea is to conduct an experimental analysis of similarity measures to characterize their behavior in specific application scenarios. For that purpose, one can distinguish laboratory and field investigations. In laboratory investigations, the process models are (possibly synthetically) generated in a controlled environment, while in field investigations, they are generated by human modelers. Since the results of a laboratory investigation cannot easily be transferred to the field, the field setting should be considered as well. We finally use three different groups of samples with different characteristics which are partially taken from a large process model corpus [27]. In contrast to [28], we added four additional laboratory model sets from Camunda™ training sessions. All used model sets with their specific characteristics are described below:

1. **Field models:** To develop these models, no restrictions regarding the labeling of model elements were given to the modeler(s). Thus, in these models, equal or similar aspects might be modeled in a different manner and expressed with different words. A dataset, containing such models from the domain of university admission (9 models) and the domain of birth registration (9 models), is provided in [2].
2. **Models from controlled modeling environments:** Models are created in a controlled environment, wherein different modelers independently model the same process based on a natural language text description. As a terminology is provided in the textual description, it is assumed that this terminology is used by the modelers as well. Student exercises[4] (18 models) serve as an adequate dataset. Additionally, models from Camunda™ training sessions[5] were included in this group (40 models). An analysis based on this dataset covers a laboratory investigation.
3. **Mined models:** The process models in this group are derived using process mining techniques. Thus, the node labels are linguistically harmonized and are (1) unambiguous and (2) consistent over the whole collection (matching problem is essentially evaded as model elements representing the same real-world activity are labeled identically). The models from Dutch governance presented in [30] fulfill this requirement (80 models). However, one can argue whether they are synthetically created in a laboratory sense or, as the processes are executed in the real-world, whether they are derived from the field.

The overall model collection contains 156 distinct models, which were compared to one another in every possible combination. This leads to similarity calculations for 24,336 business process model pairs; both directions were checked as some of the similarity measures are not symmetric (pseudo-metrics).

[4] Model set "Exams" is available in the model repository at http://rmm.dfki.de.

[5] The original models can be retrieved from https://github.com/camunda/bpmn-for-research.

5.2 Query Results

Before we were able to determine the query results for each similarity measurement technique, we needed to calculate the similarity values for all model pairs with each technique. The interval used for the similarity values was [0, 1], with higher values meaning more similar and lesser values meaning less similar.

The similarity values, calculated with the above mentioned techniques, were used as the foundation for calculating the evaluation measures in the second step. Therefore, a gold standard containing the relevant models to a specific query model was needed to determine Precision, Recall, and F-Measure values. As the underlying processes of the model collection are different, we decided to use all models related to a specific process as the relevant models. For example, when one of the University Admission models was used as a query model, we considered all of the nine University Admission models to be the relevant models for this specific query. Note that we did not remove the query model from the model dataset for querying as we also wanted to analyze how the search approaches handle models identical to the query model.

To finally calculate Precision, Recall, and F-Measure values, we used a threshold value θ on the similarity values. Only models having a similarity value equal to or above the threshold value with respect to a query model were deemed as a query result.

Regarding the R-Precision and Precision-at-k evaluation measures, we did not need a threshold value. For a query model, we simply ranked all models in descending order according to their similarity values. Afterward, we calculated Precision of the first $|R|$ results to determine R-Precision. This means, for instance, that we determined the R-Precision for one of the University Admission models based on the nine models with the highest similarity values compared to this model. The first nine models are used because the gold standard for one of the University Admission models contains nine models. We also calculated the Precision-at-5 values by calculating Precision based on the 5 highest ranked models. We decided to use $k = 5$ for the Precision-at-k measure to examine the first results, which are most likely to be viewed by a user of such a search functionality. Besides, we used quite a low value for k as the amount of relevant models was mostly nine or ten. Only for the models from the student exercise, 18 relevant models were available.

5.3 Evaluation Results

Table 2 shows the results for the similarity-based search experiment described in the previous section. The first two parts contain the macro and micro average values for Precision, Recall, and F-Measure. The macro average calculates the average over all queries, while the micro average is calculated by summing up true positives and the amount of retrieved and relevant results before computing Precision, Recall, and

Table 2 Statistics for query results (P: Precision, R: Recall, and F: F-Measure)

			LS3 [23]	LCST [9]	FBSE [32]	CF [29]	SSCAN [1]	LAROSA [12]
Macro	P	AVG	0.92	0.81	0.26	0.90	**0.96**	0.87
		STD	0.18	0.31	0.19	0.20	0.15	0.26
	R	AVG	**0.89**	0.55	0.59	0.64	0.56	0.79
		STD	0.21	0.43	0.24	0.40	0.43	0.30
	F	AVG	**0.87**	0.47	0.33	0.66	0.60	0.80
		STD	0.19	0.34	0.19	0.36	0,39	0.27
Micro	P		0.86	0.47	0.20	0.87	**0.95**	0.88
	R		**0.89**	0.52	0.59	0.59	0.53	0.78
	F		**0.87**	0.49	0.30	0.71	0.68	0.83
R-Precision			**0.95**	0.50	0.37	0.82	0.78	0.88
Precision-at-5			**0.99**	0.62	0.56	0.91	0.93	0.93
F = 1			**79**	20	0	52	54	43
Threshold			$\theta = 0.79$	$\theta = 0.47$	$\theta = 0.91$	$\theta = 0.64$	$\theta = 0.42$	$\theta = 0.27$
Calculation time			2s	>1d	54min	~1d	12min	2h

F-Measure. The third part contains the results for R-Precision, Precision-at-5, as well as the amount of queries with the F-Measure value of 1. Finally, the last row contains the threshold value, which maximized the macro average F-Measure value of each considered search technique. The best result for each evaluation measure is marked bold.

Regarding the unranked Precision, Recall, and F-Measure, four search techniques showed very good results. LS3, CF, SSCAN, and LAROSA reached (at least for some of the measures) high results. SSCAN got the highest Precision value (0.96), which is expected, as this approach counts identically labeled nodes. Thus, there is a high probability that two models with many identically labeled nodes, in fact, describe the same process. With respect to the Recall and F-Measure values, LS3 reached the highest scores of 0.89 and 0.87, respectively. However, for CF and SSCAN, Recall is the critical measure as their values are significantly lower (0.64 and 0.56). While LAROSA never reached the highest values, all evaluation values are comparably high. Additionally, LAROSA also received the second-highest scores for R-Precision and Precision-at-5. Only LS3 reached higher values for these ranked evaluation measures. Besides, CF and SSCAN again got good to very good values. The LCST and FBSE search techniques, however, reached only low values for all the considered evaluation measures.

Finally, LS3 and SSCAN reached outstanding results. Both approaches show a very good performance not only in terms of calculation complexity and calculation time but also regarding the evaluated measures. LS3 shows the best F-Measure values overall as well as the best R-Precision and the best Precision-at-5. On the contrary, SSCAN reaches the best Precision. Depending on the actual scenario, it might be meaningful to decide on the particular goal. A high Precision stands for a high probability that a query result is relevant in terms of the search argument, while

Table 3 Micro average values regarding easy and hard matching of models (P: Precision, R: Recall, and F: F-Measure)

		LS3 [23]	LCST [9]	FBSE [32]	CF [29]	SSCAN [1]	LAROSA [12]
Easy	P	0.75	**0.95**	0.25	0.94	0.94	0.82
	R	0.97	0.63	0.64	0.97	0.98	**0.99**
	F	0.85	0.75	0.36	**0.96**	**0.96**	0.90
Hard	P	**1.00**	0.28	0.17	0.69	**1.00**	0.98
	R	**0.82**	0.42	0.54	0.25	0.12	0.60
	F	**0.90**	0.33	0.26	0.37	0.21	0.74

a high Recall ensures that a great fraction of the expected results is found. Hence, both Precision and Recall are valid isolated criteria for queries, but an aggregation (F-Measure) of them makes sense for unknown scenarios as well.

In contrast to the lightweight LS3 and SSCAN approaches (in terms of the estimated calculation time[6]), CF and LCST are very expensive to calculate. Both require a derivation of traces or parts of traces to calculate a similarity value. Since the state space of a process model can explode under certain circumstances, such a calculation might even become impossible. Against that background, there is a risk of running into a situation where the approaches cannot be applied. This is indicated by the calculation times mentioned in Table 2, although these values do not allow to derive any reliable statement on performance. In fact, it cannot be expected that the implementations are optimized with regard to performance. Most of them (all but LS3) perform a pairwise comparison, which require a separate loading and parsing of the analyzed model files for each pairwise calculation. We tried to eliminate that problem by performing the task of loading and parsing in isolation, which was possible for all approaches except of LAROSA. Moreover, all approaches considering models as elements need to interpret the source data and instantiate each single node as a dedicated object. Finally, in the best case, the calculation times only state an indication of performance but do not necessarily allow to derive a reliable statement about a practical applicability.

Table 3 shows the micro average results for Precision, Recall, and F-Measure divided into two categories based on the matching difficulty. In the easy part, only the models from the Dutch municipalities dataset are included (80 models). Calculating a matching between these models is simple as the same real-world activities are labeled identically. The hard part contains the models from the field and controlled modeling categories (76 models).

The numbers from Table 3 highlight one essential difference between the LS3 approach and the three top-ranked matching-based search techniques CF, SSCAN, and LAROSA. For the easy part, LS3 is outperformed by the three matching-

[6] The actual calculation time depends on the implementation. In case of a mapping-based similarity calculation (which is the case for all evaluation measures expect of LS3), the calculation of the mapping needs to be considered as well.

based techniques regarding Precision and F-Measure. Recall values are very high for all the four approaches. This clearly shows that especially CF and SSCAN can calculate very good query results in case of easy matching. Yet, the evaluation shows an inversing result for the hard part. In the case of a difficult matching, the LS3 approach outperforms the matching-based techniques. Especially, the Recall values for CF and SSCAN drop to very low values. The problem for both approaches is determining matches as they use simple matching calculations (both do only match in the case of identical labels).

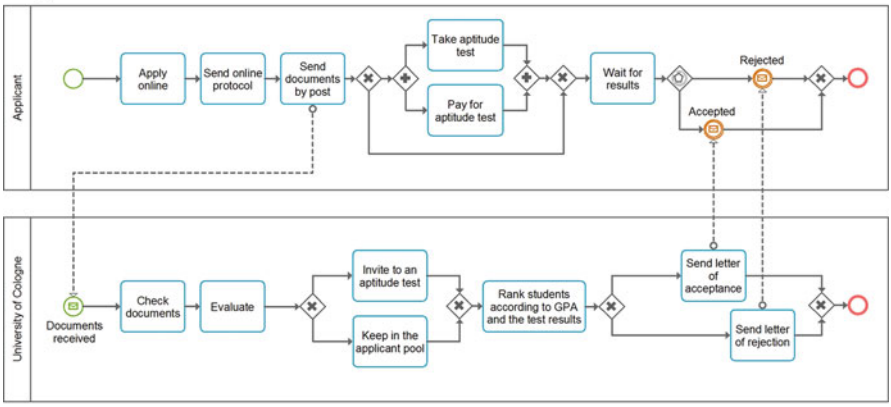
5.4 Discussion and Limitations

A limitation of the presented evaluation, and also for all other similar evaluations, is the choice and the size of the model corpus (the used dataset). Since neither the overall set of existing process models is available in a single corpus nor the overall number of existing process models is known, it is not possible to select a specific number of models randomly (which would be necessary to determine the statistical significance). Instead, as mentioned above, we selected models that are (1) appropriate for the evaluation scenario and (2) heterogeneous. Appropriate in this case means that a search in the model set is meaningful—there exist models with a naturally given similarity, so that the expected query results can be determined. Heterogeneity describes the character of the models caused by their origin, i.e., the domain, the modelers background or his modeling experience. This highly influences the complexity of the matching problem: as mined models are automatically derived, a matching problem by itself does not exist. Linguistically similar labels are probable, if the models are designed based on a consistent textual description—they are rather improbable, if this is not the case.

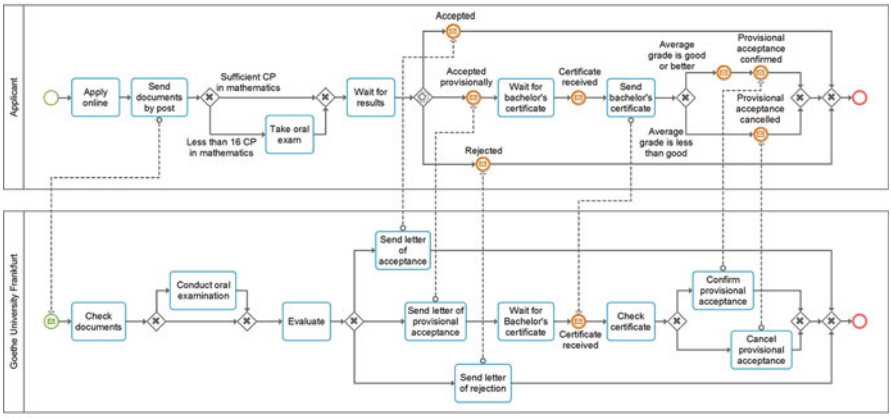
The similarity calculation uncovered limitations in the approaches "La Rosa similarity" (LAROSA) [12] and "Longest common subsequence of traces" (LCST) [9]. 9 of the 156 models in the evaluation dataset could not be processed by the original LAROSA algorithm. Although it was not possible to identify the reason for that, this led to a reduction of the model combinations to be processed by 2727, or, conversely, the similarity of 21,609 out of the overall 24,336 model combinations were successfully calculated. In case of "Longest common subsequence of traces," 45 of the 156 models could not be processed. Therefore, the similarity could only be calculated for 12,015 model combinations. The challenge for this approach lies in the necessity to calculate all the theoretically possible execution traces for a particular model, since the real-world traces are not available. The used implementation applies the approach of [17] to derive traces; loops were passed only once. Based on the used connectors and the size (in terms of nodes and edges) of a model, this can become very expensive in time and memory. Based on some preceding tests, it was decided to set a trace calculation limit of 40 seconds per model, which led to 41 cancellations. Syntactical errors were a second reason for which the traces of four models could not be calculated. Since no

Type: Models from controlled modeling environments | **Dataset:** University Admission

Cologne



Frankfurt



Technique	Sim-Value	Technique	Sim-Value
LS3	0,9000	CF	0,6300
LCST	0,5543	SSCAN	0,4828
FBSE	0,8966	LAROSA	0,4028

Fig. 1 Similarity values for two selected models

other approach requires a syntactical correctness of the process model, this is an important limitation. Nevertheless, a similarity analysis with this approach might be meaningful in specific scenarios, e.g., when (1) the real-world traces are known, so that it is not necessary to calculate them based on the model and (2) the intention is to analyze the execution behavior instead of the process concept.

Figure 1 shows two selected models from the dataset with the corresponding similarity values. The example shows several of the above discussed aspects in a concrete setting of the evaluation. First, LCST was not able to deliver a

similarity value since the runtime threshold for calculating all possible traces of 40 seconds was exceeded for at least one of the two models. Running the technique without a time limit, it delivered the similarity value after 35 minutes. Second, the implementation of LAROSA threw an exception since it was not able to handle additional object types, which seem to be unknown by the algorithm. We solved that problem for this model pair by manually removing the organizational elements/lanes. Third, we see significant differences in the resulting similarity values. Having a look at the element labels of the models uncovers a high similarity for a human (both describe a process for a University Admission) although there are different wordings in the process descriptions. Especially, the use of different expressions, such as "take oral exam" vs. "conduct oral examination", is challenging for purely syntactical similarity measures, like SSCAN.

6 Conclusion and Outlook

Based on the practical empirical evaluation, it can be stated that different process model similarity measures lead to substantially different similarity values. The reason for that is founded in (1) different competencies regarding the characteristics of the model dataset (easy vs. hard cases) and (2) the algorithmic approaches for calculating similarity. While FBSE, SSCAN, and LAROSA calculate a similarity value based on a particular node matching only, CF and LCST additionally focus on behavioral characteristics, which are derived from the model structure. In contrast to that, LS3 is the only evaluated measure, which does not require any mapping.

The measures were evaluated with a focus on process model search for process querying, wherefore different relevance criteria can be argued. On the one hand, a high Precision can be desired in order to ensure that all delivered result items are relevant for the search. A high Recall or a high F-Measure can be argued as desirable, as this improves the completeness of the result. Nevertheless, the consideration of additional similarity criteria (like the model structure for CF and LCST) did not lead to an improvement of the measurement results in terms of the expected output (process model result list). LS3 showed an outstanding performance regarding the F-Measure. Also, the results of SSCAN are convincing with constantly high Precision values.

Although the evaluation was executed in the best possible way, there are several threats to validity, which cannot be eliminated in such experiments:

1. **Selection of the models:** For the reason of statistical significance, it would be necessary to randomly select a number of process models from the ground set of existing process models. Since this ground set is unknown, the selection can never be seen as random. Instead, we selected a meaningful mix of synthetic and real-world models.
2. **Validity of the gold standard:** The gold standard is generally determined by humans. Thus, the process of reaching the gold standard is challenging. A

consistent understanding of similarity, correspondence, and the model content is necessary to reach a consensus of the truth. The gold standard represents this consensus, which might be debated again if an additional human is being involved in creating the consensus.

3. **Configuration of the evaluated similarity measurement techniques:** The configurability of similarity measurement techniques allows an adjustment that considers the characteristics of the problem to solve, e.g., depending on the origin of the model data. Since the search goal is not necessarily known, we chose the recommended standard configuration, while other settings may lead to significantly better results.

With these limitations, we can conclude that the measures with the lowest functional complexity (SSCAN and LS3) bring the most appropriate results within the application scenario of process model querying. Since this also affects an outstanding calculation performance in terms of time and consumed resources, they should be further evaluated for the purpose of querying large model repositories to validate their practical applicability.

References

1. Akkiraju, R., Ivan, A.: Discovering business process similarities: An empirical study with SAP best practice business processes. In: Maglio, P.P., Weske, M., Yang, J., Fantinato, M. (eds.) 8th International Conference on Service Oriented Computing (ICSOC), San Francisco, USA. Lecture Notes in Computer Science, vol. 6470, pp. 515–526. Springer (2010). https://doi.org/10.1007/978-3-642-17358-5_35
2. Antunes, G., Bakhshandeh, M., Borbinha, J., Cardoso, J., Dadashnia, S., Francescomarino, C.D., Dragoni, M., Fettke, P., Gal, A., Ghidini, C., Hake, P., Khiat, A., Klinkmüller, C., Kuss, E., Leopold, H., Loos, P., Meilicke, C., Niesen, T., Pesquita, C., Péus, T., Schoknecht, A., Sheetrit, E., Sonntag, A., Stuckenschmidt, H., Thaler, T., Weber, I., Weidlich, M.: The process model matching contest 2015. In: Kolb, J., Leopold, H., Mendling, J. (eds.) 6th International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA 2015), Innsbruck, Austria. Lecture Notes in Informatics, vol. P-248, pp. 127–155. Gesellschaft für Informatik (2015)
3. Cayoglu, U., Dijkman, R., Dumas, M., Fettke, P., García-Bañuelos, L., Hake, P., Klinkmüller, C., Leopold, H., Ludwig, A., Loos, P., Mendling, J., Oberweis, A., Schoknecht, A., Sheetrit, E., Thaler, T., Ullrich, M., Weber, I., Weidlich, M.: Report: The process model matching contest 2013. In: Lohmann, N., Song, M., Wohed, P. (eds.) Business Process Management Workshops. Lecture Notes in Business Information Processing, vol. 171, pp. 442–463. Springer, Beijing, China (2014). https://doi.org/10.1007/978-3-319-06257-0_35
4. de Medeiros, A.K.A., Aalst, W.M.v.d., Weijters, A.J.M.M.: Quantifying process equivalence based on observed behavior. Data Knowl. Eng. **64**(1), 55–74 (2008). https://doi.org/10.1016/j.datak.2007.06.010
5. Desel, J., Juhás, G.: "What is a Petri net?" informal answers for the informed reader. In: Ehrig, H., Padberg, J., Juhás, G., Rozenberg, G. (eds.) Unifying Petri Nets: Advances in Petri Nets, pp. 1–25. Springer, Berlin, Heidelberg (2001). https://doi.org/10.1007/3-540-45541-8_1
6. Dijkman, R., Dumas, M., García-Bañuelos, L.: Graph matching algorithms for business process model similarity search. In: Dayal, U., Eder, J., Koehler, J., Reijers, H.A. (eds.) 7th International Conference on Business Process Management (BPM), Ulm, Germany. Lecture

Notes in Computer Science, vol. 5701, pp. 48–63. Springer (2009). https://doi.org/10.1007/978-3-642-03848-8_5
7. Dijkman, R.M., Dumas, M., Dongen, B.F.v., Käärik, R., Mendling, J.: Similarity of business process models: Metrics and evaluation. Information Systems **36**(2), 498–516 (2011). https://doi.org/10.1016/j.is.2010.09.006
8. Dumas, M., García-Bañuelos, L., Dijkman, R.M.: Similarity search of business process models. IEEE Data Eng. Bull. **32**(3), 23–28 (2009)
9. Gerke, K., Cardoso, J., Claus, A.: Measuring the compliance of processes with reference models. In: Meersman, R., Dillon, T., Herrero, P. (eds.) On the Move to Meaningful Internet Systems (OTM), Confederated International Conferences, CoopIS, DOA, IS, and ODBASE, Part I, Vilamoura, Portugal. Lecture Notes in Computer Science, vol. 5870, pp. 76–93. Springer (2009). https://doi.org/10.1007/978-3-642-05148-7_8
10. Keller, G., Nüttgens, M., Scheer, A.W.: Semantische Prozeßmodellierung auf der Grundlage "Ereignisgesteuerter Prozeßketten (EPK)". Tech. rep., Institut für Wirtschaftsinformatik, Universität Saarbrücken (1992)
11. Kunze, M., Weidlich, M., Weske, M.: Behavioral similarity - A proper metric. In: 9th International Conference on Business Process Management, Clermont-Ferrand, France, pp. 166–181 (2011)
12. La Rosa, M., Dumas, M., Uba, R., Dijkman, R.M.: Business process model merging: An approach to business process consolidation. ACM Trans. Softw. Eng. Methodol. (TOSEM) **22**(2), 11:1–11:42 (2013). https://doi.org/10.1145/2430545.2430547
13. Landauer, T.K., Foltz, P.W., Laham, D.: An introduction to latent semantic analysis. Discourse Processes **25**(2-3), 259–284 (1998). https://doi.org/10.1080/01638539809545028
14. Lau, C.K., Fournier, A.J., Xia, Y., Recker, J., Bernhard, E.: Process Model Repository Governance at Suncorp. Tech. rep., Business Process Management Research Group, Queensland University of Technology (2011). http://apromore.org/wp-content/uploads/2011/12/Suncorp-project-report-2.pdf
15. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Sov. Phys. Doklady **10**(9), 707–710 (1966)
16. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge, England (2008). https://doi.org/10.1017/CBO9780511809071
17. Mendling, J.: Detection and Prediction of Errors in EPC Business Process Models. Ph.D. thesis, Wirtschaftsuniversität Wien (2007)
18. Murata, T.: Petri nets properties, analysis and applications. Proc. IEEE **77**(4), 541–580 (1989). https://doi.org/10.1109/5.24143
19. Object Management Group (OMG): Business Process Model and Notation (BPMN) (2011). http://www.omg.org/spec/BPMN/2.0
20. Polyvyanyy, A., Ouyang, C., Barros, A., van der Aalst, W.M.P.: Process querying: Enabling business intelligence through query-based process analytics. Decis. Support Syst. **100**, 41–56 (2017). https://doi.org/10.1016/j.dss.2017.04.011
21. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB J. **10**(4), 334–350 (2001). https://doi.org/10.1007/s007780100057
22. Schoknecht, A., Fischer, N., Oberweis, A.: Process model search using latent semantic analysis. In: Dumas, M., Fantinato, M. (eds.) Business Process Management Workshops: BPM 2016 International Workshops, Rio de Janeiro, Brasilien, Revised Papers. Lecture Notes in Business Information Processing, vol. 281, pp. 283–295. Springer (2017). https://doi.org/10.1007/978-3-319-58457-7_21
23. Schoknecht, A., Oberweis, A.: LS3: Latent semantic analysis-based similarity search for process models. Enterp. Modell. Inf. Syst. Archit. **12**(2), 1–22 (2017). https://doi.org/10.18417/emisa.12.2
24. Schoknecht, A., Thaler, T., Fettke, P., Oberweis, A., Laue, R.: Similarity of business process models — A state-of-the-art analysis. ACM Comput. Surv. **50**(4), 52:1–52:33 (2017). https://doi.org/10.1145/3092694

25. Song, L., Wang, J., Wen, L., Wang, W., Tan, S., Kong, H.: Querying process models based on the temporal relations between tasks. In: 15th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW), Helsinki, Finland, pp. 213–222. IEEE Computer Society, Helsinki, Finland (2011). https://doi.org/10.1109/EDOCW.2011.12
26. Thaler, T., Hake, P., Fettke, P., Loos, P.: Evaluating the evaluation of process matching techniques. In: Kundisch, D., Suhl, L., Beckmann, L. (eds.) Multikonferenz Wirtschaftsinformatik (MKWI), Paderborn, Germany, pp. 1600–1612. University of Paderborn, Paderborn, Germany (2014)
27. Thaler, T., Dadashnia, S., Sonntag, A., Fettke, P., Loos, P.: The IWi Process Model Corpus. Tech. rep., Institute for Information Systems (IWi) at the German Research Center for Artificial Intelligence (DFKI) (2015)
28. Thaler, T., Schoknecht, A., Fettke, P., Oberweis, A., Laue, R.: A comparative analysis of business process model similarity measures. In: Dumas, M., Fantinato, M. (eds.) Business Process Management Workshops: BPM 2016 International Workshops, Rio de Janeiro, Brasil, Revised Papers. Lecture Notes in Business Information Processing, vol. 281, pp. 310–322. Springer (2017). https://doi.org/10.1007/978-3-319-58457-7_23
29. van Dongen, B.F., Dijkman, R.M., Mendling, J.: Measuring similarity between business process models. In: Bellahsene, Z., Léonard, M. (eds.) 20th International Conference on Advanced Information Systems Engineering (CAiSE), Montpellier, Frankreich. Lecture Notes in Computer Science, vol. 5074, pp. 450–464. Springer (2008). https://doi.org/10.1007/978-3-540-69534-9_34
30. Vogelaar, J., Verbeek, H., Luka, B., van der Aalst, W.M.: Comparing business processes to determine the feasibility of configurable models: A case study. In: Business Process Management Workshops, Clermont-Ferrand, France, pp. 50–61 (2011)
31. Weidlich, M., Dijkman, R., Jan, M.: The ICoP framework: Identification of correspondences between process models. In: 22nd International Conference on Advanced Information Systems Engineering (CAiSE), Hammamet, Tunisia (2010)
32. Yan, Z., Dijkman, R., Grefen, P.: Fast business process similarity search with feature-based similarity estimation. In: Meersman, R., Dillon, T., Herrero, P. (eds.) On the Move to Meaningful Internet Systems (OTM), Confederated International Conferences CoopIS, IS, DOA and ODBASE, Part I, Hersonissos, Greece. Lecture Notes in Computer Science, vol. 6426, pp. 60–77. Springer (2010). https://doi.org/10.1007/978-3-642-16934-2_8
33. Zezula, P., Amato, G., Dohnal, V., Batko, M.: Similarity search - The metric space approach. Adv. Database Syst., **32** (2006)