Predicting the Quality of Process Model Matching

Matthias Weidlich[1], Tomer Sagi[1], Henrik Leopold[2], Avigdor Gal[1], and Jan Mendling[3]

[1] Technion - Israel Institute of Technology, Technion City, 32000 Haifa, Israel
{weidlich,stomers7}@tx.technion.ac.il, avigal@ie.technion.ac.il
[2] Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany
henrik.leopold@wiwi.hu-berlin.de
[3] Wirtschaftsuniversität Wien, Augasse 2-6, A-1090 Vienna, Austria
jan.mendling@wu.ac.at

Abstract. *Process model matching* refers to the task of creating correspondences among activities of different process models. This task is crucial whenever comparison and alignment of process models are called for. In recent years, there have been a few attempts to tackle process model matching. Yet, evaluating the obtained sets of correspondences reveals high variability in the results. Addressing this issue, we propose a method for predicting the quality of results derived by process model matchers. As such, prediction serves as a case-by-case decision making tool in estimating the amount of trust one should put into automatic matching. This paper proposes a model of prediction for process matching based on both process properties and preliminary match results.

1 Introduction

Process models have been widely established as a tool to manage business operations. They may be created for different purposes, such as process documentation or workflow implementation and in different contexts, e.g., for different organisational units or at different points in time. Many use cases require the comparison and alignment of process models, for instance, the validation of a technical process implementation against a business-centred specification model [1] or clone detection within a process model repository [2]. The need for comparing process models fostered research on *process model matching*, which refers to the task of creating correspondences among activities of different process models.

Recently, there have been a few attempts to tackle process model matching [1,3,4,5]. Typically, the developed matchers relied on the rich literature of schema and ontology matching [6,7] with emphasis on string comparison and graph matching. Evaluating the outcome of these works shows that the empirical quality is subject to high variability even within a single dataset. While matchers yield high precision and recall for some matching tasks, they entirely fail for others. This raises the question of *how to distinguish matching tasks for which matchers yield high quality matches from those for which results are poor.*

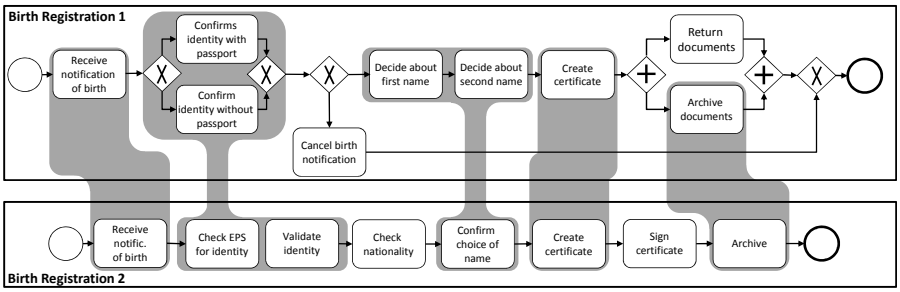

Fig. 1. Example of two business process models and their correspondences.

In this work, we offer preliminary observations on how to use matching prediction [8] to tackle this question. We observe that both properties of process models and the similarity between their characteristics impact the accuracy of a specific match task. In particular, differences in syntactic and semantic aspects of labels as well as structure and behaviour of the models can be taken into account. Based on such process properties and similarity characteristics, we develop a statistical model to predict the quality of a match result for a given task.

The rest of the paper is organized as follows. Section 2 introduces the process model matching problem and the state-of-the-art in schema matching prediction. Our approach to prediction for process model matching is detailed in Section 3. Section 4 reviews related work, before Section 5 concludes the paper.

2 Background

This section first reviews the matching problem for process models, before explaining the background of schema matching prediction.

The Matching Problem. For two process models with A_1 and A_2 as their sets of activities, process model matching aims at identifying activity correspondences that represent the same behaviour in both models. Following Gal [6], we subdivide the matching process into first and second line matching. A *first line matcher* operates on the process models, compares some of their attributes such as activity labels or the process structure, and produces a similarity matrix M(A_1,A_2) over activities with $|A_1|$ rows and $|A_2|$ columns. A *second line matcher* works on one or more similarity matrices, e.g., by thresholding or combining them. Certain second line matchers create a binary similarity matrix M'(A_1,A_2) with entries being either 0 or 1, the latter represents correspondences.

Figure 1 illustrates the matching problem with two processes for registering a newborn. Although both processes are similar, the lower process slightly deviates from the upper one. Considering the highlighted correspondences between both models, it becomes apparent that some matches are more easily identified than others. A straightforward correspondence is the one between *Receive notification of birth* and *Receive notific. of birth* as measures for first line matching, e.g., the

Levenshtein distance, indicate high values in a similarity matrix. However, to identify complex correspondences involving sets of activities, like the one between *Decide about first name* and *Decide about second name* with *Confirm choice of name*, the usage of semantic knowledge and the model structure is required. For instance, lexical databases can help to identify that words such as *decide* and *confirm* are close in their meaning. Other differences relate to models in their properties. The upper process contains three splits, whereas the lower one represents a plain sequence of tasks. Hence, we anticipate that there may be several complex correspondences that are generally harder to identify.

Schema Matching Prediction. Our approach relies on recent results on *schema matching prediction* [8]. In the absence of a ground truth, matchers perform a "best effort" matching without any indication of the prospective success of their efforts. Schema matching prediction provides an assessment mechanism that supports schema matchers in this context. Predictors foretell the success of a matcher in identifying correct correspondences by analysing the matcher's pair-wise similarity scores.

Sagi and Gal [8] argue for the importance of *tunability* to support prediction of different qualities, putting emphasis, e.g., on precision or recall. Thus, prediction models compose various, loosely correlated predictors into a statistical model. The weights of participating predictors are tuned, so that the combined prediction correlates well with the desired quality criterion. To accommodate for tunable prediction models, our work leverages a set of matrix evaluation functions, termed *matrix predictors* [8]. These predictors encode different assumptions on how a particular value distribution in a similarity matrix indicates the likelihood of a successful match. Each predictor is applied to a similarity matrix M(A_1,A_2) obtained by a first line matcher and yields a non-binary prediction value. Here, we give two examples for such predictors.

- An entry (a_i, a_j) in M(A_1,A_2) is dominant, if it has the highest value in the respective row and column of the matrix. The *Dominants* matrix predictor measures the ratio of dominant values and $k = \min\{|A_1|, |A_2|\}$.
- The *Binary Matrix* predictor measures the distance between M(A_1,A_2) and the closest ideal matrix M'(A_1,A_2) in a vector space, where M'(A_1,A_2) is required to be a binary matrix (with the entries being either 0 or 1).

3 Prediction for Process Model Matching

This section introduces our approach to predicting the quality of process model matching. Figure 2 illustrates the major components of the proposed prediction architecture. Given a pair of process models, prediction may either be based solely on process properties (top) or on a similarity measure (bottom). In the first case, properties of both process models are extracted and process property predictors derive a score for the model pair. In the second case, we obtain a set of similarity matrices over the activities of the processes. Then, matrix predictors exploit characteristics of these matrices to obtain a prediction score per model pair and similarity measure.

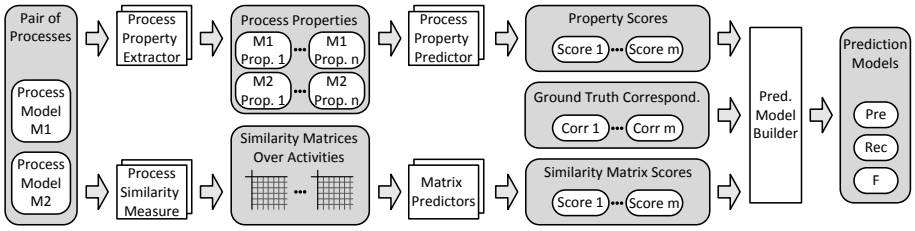

Fig. 2. Overview of the prediction architecture.

The prediction scores are combined into a prediction model, predicting a certain quality measure, such as precision, recall, or their harmonic mean, the F-score. To this end, a stepwise (multi-valued) regression is performed over the whole set of predictors in the presence of a ground truth, i.e., a set of correspondences that are known to hold true. Depending on the origin of the prediction scores, the prediction model generalises in different dimensions, e.g., it is relative to process properties (if only scores per process pair are considered) or to process properties and characteristics of similarity measures.

Below, we instantiate this architecture with strategies for prediction based on process properties (Section 3.1) and process similarity measures (Section 3.2). Then, Section 3.3 discusses how prediction is used to answer the question of how to identify matching tasks for which matchers yield high quality results.

3.1 Prediction based on Process Properties

Below, we present several examples of property predictors, each taking a certain process property and implementing an evaluation measure.

Avg Length of Labels (ALL). An example of a property referring to the textual syntax is the average length of activity labels. It can be expected to yield insights on the suitability of textual similarity measures. With avg_1 and avg_2 as the average lengths of activity labels in two processes, we define two predictors.

- ALLR is the relative difference between the properties, i.e., the prediction score is $p_{ALLR} = 1$ if $\max\{avg_1, avg_2\} = 0$ and $p_{ALLR} = 1 - (|avg_1 - avg_2| / \max\{avg_1, avg_2\})$ otherwise.
- ALLA is the arithmetic average of the smoothed absolute deviation of the property values from a label length l (e.g., $l = 20$), i.e., $p_{ALLA} = (1 - \max\{0, 1 - 0.01(avg_1 - 20)^2\})/2 + (1 - \max\{0, 1 - 0.01(avg_2 - 20)^2\})/2$.

Number of Labels with Action in a Lexical Database (NLALD). To take semantic textual features into account, we consider the number of activity labels for which the action can be found in a lexical database, e.g. WordNet [9] for English. Therefore, all activity labels are annotated with their semantic components using the approach presented by Leopold et al. [10]. Then, a lookup in a lexical database is performed. If the lookup is successful in many cases, we expect good results of textual similarity measures. With act_1 and act_2 as the numbers of activities of two process models and $actAction_1$ and $actAction_2$ as the number of

these activities with labels for which the lookup succeeded, a predictor uses the arithmetic average of the ratio of labels for which actions have been found as an evaluation function, i.e., $p_{NLALD} = (actAction_1/act_1)/2 + (actAction_2/act_2)/2$.

Number of Nodes in Cycles (NNC). Focussing on structural features of a process model, we consider the number of nodes in control flow cycles. A large difference in the values of this property for two process models hints at different control flow structures and, thus, lower chances of achieving a good match result. Our predictor, thus, applies a relative comparison. With $inCycle_1$ and $inCycle_2$ as the number of nodes in two process models that are part of a control flow cycle, the prediction score is $p_{NNC} = 1$ if $\max\{inCycle_1, inCycle_2\} = 0$ and $p_{NNC} = 1 - (|inCycle_1 - inCycle_2| / \max(inCycle_1, inCycle_2))$ otherwise.

Depth of the RPST (DRPST). Another structural property of a process model is the depth of a decomposition tree, e.g., defined by the Refined Process Structure Tree (RPST) [11]. The depth of this tree provides a means to assess the complexity of the control flow structure. Complex control flow structures in either model as well as large differences in this complexity can be expected to have a negative impact on process model matching. With $depth_1$ and $depth_2$ as the depths of the RPSTs of two process models, we define two predictors.

- DRPSTR measures the relative difference, i.e., $p_{DRPSTR} = 1 - (|depth_1 - depth_2| / \max\{depth_1, depth_2\})$.
- DRPSTA measures the arithmetic average of a smoothed absolute depth, i.e., $p_{DRPSTA} = (1 - \max\{0, 1 - 0.02(depth_1 - 1)^2\})/2 + (1 - \max\{0, 1 - 0.02(depth_2 - 1)^2\})/2$.

Size of the Concurrency Relation (SCR). The size of the concurrency relation is an example for a behavioural property. The concurrency relation contains all pairs of activities that may be enabled concurrently in some reachable state of the process, cf., [12]. Since its size provides insights on behavioural complexity, high absolute values and large relative differences for two process models may lower the result quality of process model matchers. Let $conc_1$ and $conc_2$ be the sizes of the concurrency relation of two process models with act_1 and act_2 as the number of activities, respectively. Then, we define two predictors.

- SCRR measures the relative difference, i.e., $p_{SCRR} = 1$ if $\max\{conc_1, conc_2\} = 0$ and $p_{SCRR} = 1 - (|conc_1 - conc_2| / \max\{conc_1, conc_2\})$ otherwise.
- SCRA measures the arithmetic average of the ratio of concurrent activities, i.e., $p_{SCRA} = (conc_1/act_1^2)/2 + (conc_2/act_2^2)/2$.

3.2 Prediction based on Similarity Measures

Most matchers use textual similarity measures for deriving match candidates and consider structural and behavioural features for selecting correspondences. Hence, below, we focus on textual similarity measures and, for each of them, apply one of the matrix predictors (Section 2) to obtain a prediction score.

Optimal String Edit Distance over Activity Labels (OSEDAL). A first syntactical similarity measure is the optimal string edit distance over tokenised activity labels. First, string preprocessing techniques, such as stop word removal and stemming, are applied to all terms. For the remaining, preprocessed terms

$\{t_1, \ldots, t_n\}$ and $\{t'_1, \ldots, t'_m\}$ of two activity labels, we seek an optimal term alignment $\sim \subseteq \{t_1, \ldots, t_n\} \times \{t'_1, \ldots, t'_m\}$ that (1) relates one term t to at most one term t' and (2) maximises the string edit distance similarity $seds(t, t') = lev(t, t')/\max\{|t|, |t'|\}$ (with lev as the Levenshtein Distance and $|\cdot|$ as the term length) over the aligned pairs. Then, the similarity measure for the two activities is defined as $s_{OSEDAL} = \sum_{t \sim t'} seds(t, t')/\max\{n, m\}$.

Virtual Docs Distance Set (VDDS). As another syntactical similarity measure, we define a measure that first groups activities in either process and then assesses their similarity based on virtual documents. We follow the heuristics of the Distance Doc Searcher of the ICoP framework [4] for grouping activities: Given a base activity, we group activities based on their distance in the flow graph according to predefined patterns, e.g., a join (activities from which the base activity can be reached within a certain distance). For these groups, we derive virtual documents as the union of terms of all activity labels. For two virtual documents d and d', a vector space is created and the Cosine similarity $cos(d, d')$ is used to assess their similarity, cf., [13]. For each pair of activities a_1 and a_2, the similarity score is defined as $s_{VDDS} = \max\{cos(d_1, d'_1), \ldots, cos(d_n, d'_m)\}$ with $d_1, \ldots, d_n$ and $d'_1, \ldots, d'_m$ as documents representing groups that include a_1 and a_2, respectively.

Number of (Common) Semantic Components of Activity Labels (N(C)SCAL). Turning to semantic features, we compute a similarity based on the common semantic components of activity labels. Again, we annotate activity labels to obtain their semantic components a^{ac} (the action), a^{bo} (the object), and a^{ad} (an additional part) for the activity label a [10]. Let $comp_a \subseteq \{ac, bo, ad\}$ denote the type of the components found for activity a. Then, we define two similarity measures:

- Similarity in the number of semantic components $comp_1$ and $comp_2$ for two activities a_1 and a_2 is considered by the measure defined as $s_{NSCAL} = 1 - |((|\, comp_{a_1} \,| - |\, comp_{a_2} \,|)/3)|$.
- Similarity in the types of semantic components is considered by an adapted measure, defined as $s_{NCSCAL} = 1 - (|\, comp_{a_1} \cap comp_{a_2} \,|/3)$.

Lin Distance between Activity Labels (LDAL). A fine-grained similarity measure for semantic features, is the semantic distance among the activity labels. Given the semantic components of two labels, we assess their semantic distance with the Lin metric [14] (denoted by lin). For two activities a_1 and a_2 with a_1^{ac}, a_1^{bo}, a_1^{ad}, and a_2^{ac}, a_2^{bo}, a_2^{ad} as their semantic components, we define the similarity score as $s_{LDAL} = (lin(a_1^{ac}, a_2^{ac}) + lin(a_1^{bo}, a_2^{bo}) + lin(a_1^{ad}, a_2^{ad}))$.

3.3 Assessing the Confidence in Match Results

Predictions are used to determine the confidence that is associated with a match result. Because of their high result variability, existing matchers are rarely applicable in a setting that requires unsupervised matching, e.g., similarity search in process model repositories. Further, even if applied as a semi-automated technique, a process expert has to review all match results including those that have high quality and could directly have been processed further, as well as

those that have poor quality and should thus be neglected. Using the prediction architecture, we address this issue following a machine learning approach:

(1) In a training phase, prediction models are created for a sample of process models for which some ground truth is available.
(2) In an application phase, the prediction models are applied to further matching tasks, i.e., pairs of process models, for which the ground truth is not available.

The score obtained by a prediction model is interpreted as the confidence in the match result. Prediction models should generalise for different process model matchers. They are defined relative only to process properties and characteristics of some basic similarity measures, such that the predicted score, the confidence in the match result, holds independent of any concrete matcher.

Since prediction models are geared towards a quality criterion, they can be selected for the envisioned setting of process model matching. For instance, a recall-oriented prediction qualifies for computation of match confidence when conducting process model clone detection, in order not to miss potential clones for manual evaluation. For matching in automated similarity search over process models, in turn, precision-oriented prediction may be a better option.

4 Related Work

Recently, various approaches addressed the problem of process model matching [1,3,4,5]. These works typically combine a measure for textual similarity applied for first line matching with a measure for structural or behavioural similarity that guides the second line matching. Our prediction architecture considers both and, thus, can be seen as a first step towards integrating matchers that have different strengths for certain types of matching tasks.

As for basic similarity measures, we focussed on those commonly used for process model matching. Yet, additional measures may be considered. For instance, a large number of string distance metrics, as reviewed by Cohen et al. [15], has been presented for assessing syntactic, textual similarity. Besides, a large body of structural and behavioural similarity measures are available for process models, e.g., [16,17,18]. Recent surveys of these techniques have been presented by Dijkman et al. [19] and Becker and Laue [20]. These measures can be integrated in our architecture to broaden the basis of prediction.

5 Conclusion

The presented approach addresses the issue of variability in the results obtained by process model matchers. We showed how prediction models for the quality of match results are created based on predictors that refer to process properties and characteristics of similarity measures. Such prediction models allow for assigning a confidence value to a match result.

In future work, we aim at conducting prediction on a more fine-granular level, i.e., for individual correspondences. Also, we want to exploit prediction not only for post-matching analysis, but also for improving the actual matching.

References

1. Branco, M.C., Troya, J., Czarnecki, K., Küster, J.M., Völzer, H.: Matching business process workflows across abstraction levels. In: MoDELS. Volume 7590 of LNCS., Springer (2012) 626–641
2. Ekanayake, C.C., Dumas, M., García-Bañuelos, L., Rosa, M.L., ter Hofstede, A.H.M.: Approximate clone detection in repositories of business process models. [21] 302–318
3. Dijkman, R.M., Dumas, M., García-Bañuelos, L., Käärik, R.: Aligning business process models. In: EDOC, IEEE Computer Society (2009) 45–53
4. Weidlich, M., Dijkman, R.M., Mendling, J.: The ICoP framework: Identification of correspondences between process models. In: CAiSE. Volume 6051 of LNCS., Springer (2010) 483–498
5. Leopold, H., Niepert, M., Weidlich, M., Mendling, J., Dijkman, R.M., Stuckenschmidt, H.: Probabilistic optimization of semantic process model matching. [21] 319–334
6. Gal, A.: Uncertain Schema Matching. Morgan & Claypool Publishers (2011)
7. Bellahsene, Z., Bonifati, A., Rahm, E., eds.: Schema Matching and Mapping. Springer (2011)
8. Sagi, T., Gal, A.: Schema matching prediction with applications to data source discovery and dynamic ensembling. Technical Report IE/IS-2013-02, Technion (March 2013) `http://ie.technion.ac.il/tech_reports/1364134687_Prediction_v7.pdf`.
9. Miller, G.A.: WordNet: A lexical database for English. Commun. ACM **38**(11) (1995) 39–41
10. Leopold, H., Smirnov, S., Mendling, J.: On the refactoring of activity labels in business process models. Inf. Syst. **37**(5) (2012) 443–459
11. Vanhatalo, J., Völzer, H., Koehler, J.: The refined process structure tree. Data Knowl. Eng. **68**(9) (2009) 793–818
12. Kovalyov, A., Esparza, J.: A polynomial algorithm to compute the concurrency relation of free-choice signal transition graphs. In: WODES, Edinburgh, Scotland, UK, IEE Society (1996)
13. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM **18**(11) (1975) 613–620
14. Lin, D.: An information-theoretic definition of similarity. In: In Proceedings of the 15th International Conference on Machine Learning, Morgan Kaufmann (1998) 296–304
15. Cohen, W.W., Ravikumar, P.D., Fienberg, S.E.: A comparison of string distance metrics for name-matching tasks. In: IIWeb. (2003) 73–78
16. Wombacher, A., Li, C.: Alternative approaches for workflow similarity. In: IEEE SCC, IEEE Computer Society (2010) 337–345
17. Corrales, J.C., Grigori, D., Bouzeghoub, M.: BPEL processes matchmaking for service discovery. In: OTM. Volume 4275 of LNCS., Springer (2006) 237–254
18. Kunze, M., Weidlich, M., Weske, M.: Behavioral similarity - a proper metric. In: BPM. Volume 6896 of LNCS., Springer (2011) 166–181
19. Dijkman, R.M., Dumas, M., van Dongen, B.F., Käärik, R., Mendling, J.: Similarity of business process models: Metrics and evaluation. Inf. Syst. **36**(2) (2011) 498–516
20. Becker, M., Laue, R.: A comparative survey of business process similarity measures. Computers in Industry **63**(2) (2012) 148–167
21. Barros, A.P., Gal, A., Kindler, E., eds.: Business Process Management - 10th International Conference, BPM 2012, Tallinn, Estonia, September 3-6, 2012. Proceedings. Volume 7481 of LNCS., Springer (2012)