Matching Business Process Models Using Positional Passage-based Language Models

Matthias Weidlich[1], Eitam Sheetrit[1], Moisés C. Branco[2], and Avigdor Gal[1]

[1] Technion - Israel Institute of Technology, Technion City, 32000 Haifa, Israel
{weidlich,eitams}@tx.technion.ac.il, avigal@ie.technion.ac.il
[2] Generative Software Development Laboratory, University of Waterloo, Canada
mcbranco@gsd.uwaterloo.ca

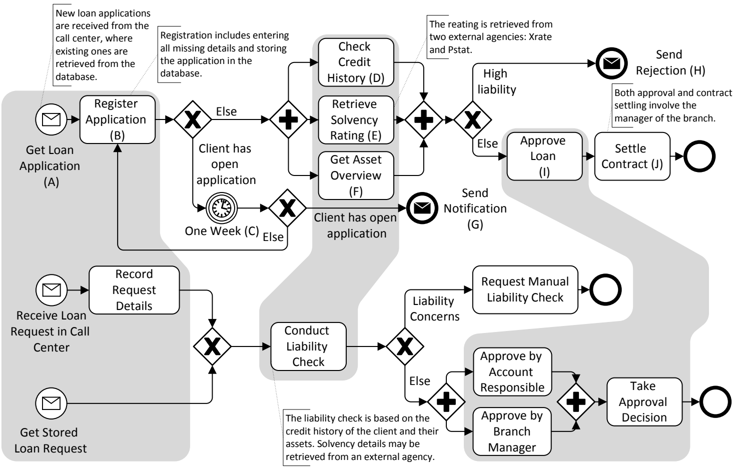
Abstract. Business operations are often documented by business process models. Use cases such as system validation and process harmonization require the identification of correspondences between activities, which is supported by matching techniques that cope with textual heterogeneity and differences in model granularity. In this paper, we present a matching technique that is tailored towards models featuring textual descriptions of activities. We exploit these descriptions using ideas from language modelling. Experiments with real-world process models reveal that our technique increases recall by up to factor five, largely without compromising precision, compared to existing approaches.

1 Introduction

Business process models were established for managing the lifecycle of a business process [1]. Many use cases require a comparison of process models, among them validation of a technical implementation of a business process against a business-centred specification [2], process harmonization [3], and effective search [4].

Comparison of process models involves *matching*, the construction of correspondences between activities. Such correspondences are highlighted in Fig. 1 for two models of a loan request process, defined in the Business Process Model and Notation (BPMN) [5]. The example illustrates the two major challenges of process model matching, namely textual heterogeneity and differences in model granularity. The latter leads to the definition of correspondences between sets of activities instead of single activities.

Recently, several approaches that support process model matching have been presented [2,6,7,8], inspired by techniques from schema matching [9]. These works rely on activity labels and structural or behavioural features of process models. However, they largely neglect the fact that process models are often used as documentation artefacts for which additional textual descriptions are available. Organisations provide descriptions for activities and maintain glossaries that explain the terms used in activity labels. In Fig. 1, for instance, the annotations for activity *'Conduct Liability Check'* indeed support the highlighted correspondence by referring to keywords such as *'credit history'* and *'solvency'*. Exploiting this information can improve over matching that is based only on activity labels.



New loan applications are received from the call center, where existing ones are retrieved from the database.

Registration includes entering all missing details and storing the application in the database.

The reating is retrieved from two external agencies: Xrate and Pstat.

Both approval and contract settling involve the manager of the branch.

The liability check is based on the credit history of the client and their assets. Solvency details may be retrieved from an external agency.

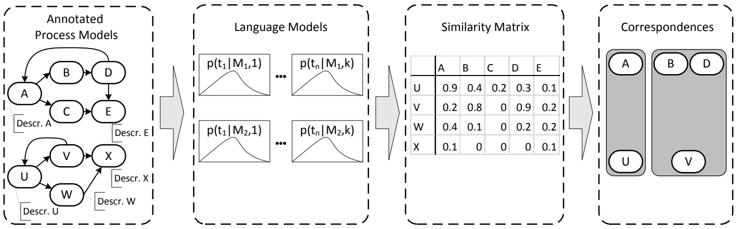
Fig. 1. An example for process model matching, correspondences are highlighted.

In this paper, we present an approach to leverage textual descriptions for matching. We look at the problem from an Information Retrieval (IR) perspective. More precisely, we combine two different streams of work on probabilistic language modelling. First, we adopt passage-based modelling such that activities are passages of a document representing a process model. Second, we consider structural features of process models by positional language modelling. Our contribution is the definition of a novel language model as well as its application for process model matching. Common IR techniques are geared towards matching a single query with a collection of documents and, thus, are not applicable in our context. Hence, we also discuss how to judge the similarity of activities and derive correspondences. Our evaluation with industry process models shows that our approach can outperform existing techniques by up to a factor of five in recall, largely without compromising precision.

The rest of the paper is structured as follows. The next section provides background on language modelling. Section 3 presents our matching approach based on a positional passage-based language model. Section 4 presents an experimental evaluation. Section 5 reviews related work, before Section 6 concludes the paper.

2 Language Models for Information Retrieval

Information Retrieval (IR) extract *relevant*, often textual, information from a corpus [10] by comparing a *query* to a collection of *documents*. Recently, language models have been successfully applied in IR. In essence, they characterise a language by assigning a probability to the occurrence of a term [11]. To answer a query in IR, one first derives a language model for each document. Then, the likelihood that the query has been generated by the same language model is estimated, which yields a ranking of documents.



	A	B	C	D	E
U	0.9	0.4	0.2	0.3	0.1
V	0.2	0.8	0	0.9	0.2
W	0.4	0.1	0	0.2	0.2
X	0.1	0	0	0	0.1

Fig. 2. Overview of the language model-based process model matching.

To illustrate the basic idea, let $\mathcal{T}$ be a set of terms and $\mathcal{B}(\mathcal{T})$ the set of all multisets over terms. Let $d \in \mathcal{B}(\mathcal{T})$ be a document with $d(t)$ as the number of occurrences of term t in d. A simple language model is a probability distribution over terms, which is based on the number of occurrences of a term in a document:

$$p(t|d) = \frac{d(t)}{\sum_{t' \in \mathcal{T}} d(t')}. \tag{1}$$

Equation 1 is independent of the importance of terms given a *corpus*, a set of documents. To countervail this effect, language models are smoothed by adding a certain probability mass to all terms that occur frequently in the corpus [12].

Our work adopts *positional* language models that define a document as a sequence of terms with a probability for a term at a document position [12]. Term proximity is integrated by propagation: term occurrences are propagated to neighbouring positions. Our approach is also inspired by *passage-based* models [13]. These models build on parts of a document identified, e.g., by section headers. As such, a passage-based model captures the probability of a term in such a passage.

3 Language Model based Matching

The basic steps of our approach to process matching using language models are illustrated in Fig. 2. Below, we provide further details on the underlying concepts.

Positional Passage-based Language Model. Our approach is based on a novel positional, passage-based language model. A process model is represented as a document, activities are passages of that document, and their proximity in the model is taken into account using passage positions.

For a process model P, we create a document $d = \langle T_1, \ldots, T_n \rangle$ as a sequence of length $n \in \mathbb{N}$ of passages, where each passage is a set of terms $d(i) = T \subseteq \mathcal{T}$, $1 \leq i \leq n$. The set $d(i) = T$ comprises all terms that occur in the label or description of the process model activity at position i. The length of d is denoted by $|d|$. We denote by $\mathcal{D}$ a set of processes, represented as documents.

Our model is built on a cardinality function $c : (\mathcal{T} \times \mathcal{D} \times \mathbb{N}) \rightarrow \{0, 1\}$, such that $c(t, d, i) = 1$ if $t \in T = d(i)$ (term t occurs in the i-th passage of d) and $c(t, d, i) = 0$ otherwise. To realize term propagation to close-by positions, a proximity-based density function $k : (\mathbb{N} \times \mathbb{N}) \rightarrow [0, 1]$ is used to assign a

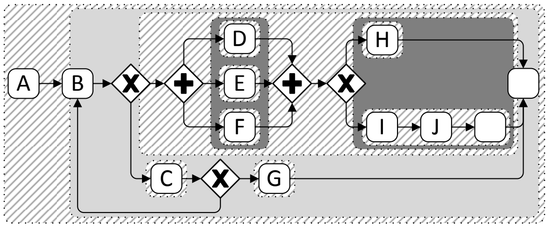


Fig. 3. Structure of the upper model from Fig. 1, normalized by an artificial end node. An example order of activities would be $A, B, E, F, D, H, I, J, C, G$.

discounting factor to pairs of positions. Then, $k(i, j)$ represents how much of the occurrence of a term at position j is propagated to position i. Lv and Zhai proposed several proximity-based kernel density functions [12]. We rely on the Gaussian Kernel $k^g(i, j) = e^{(-(i-j)^2)/(2\sigma^2)}$, defined with a spread parameter $\sigma \in \mathbb{R}^+$. Adapting function c with term propagation, we obtain a function $c' : (\mathcal{T} \times \mathcal{D} \times \mathbb{N}) \rightarrow [0, 1]$, such that $c'(t, d, i) = \sum_{j=1}^{n} c(t, d, j) \cdot k^g(i, j)$. Then, our positional, passage-based language model $p(t|d, i)$ captures the probability of term t occurring in the i-th passage of document d.

$$p(t|d, i) = \frac{c'(t, d, i)}{\sum_{t' \in \mathcal{T}} c'(t', d, i)}. \tag{2}$$

To consider importance of terms, we apply smoothing to the language model by treating each process model as a separate corpus. Then, with the corpus language model $p(t|d)$ being defined according to Equation 1, the adapted language model is defined as follows ($\mu \in \mathbb{R}$, $\mu > 0$, is a weighting factor):

$$p_\mu(t|d, i) = \frac{c'(t, d, i) + \mu \cdot p(t|d)}{\sum_{t' \in \mathcal{T}} c'(t', d, i) + \mu}. \tag{3}$$

To define how the order of passages in the document represents the order of activities in a process, we leverage the Refined Process Structure Tree (RPST) [14], a structural decomposition of a process model. The RPST is a hierarchy of non-overlapping fragments with single entry and single exit nodes. A flow arc is a *trivial* fragment; a sequence of nodes and flow arcs is a *polygon* (highlighted with striped background in Fig. 3); a fragment with multiple independent branches between the entry and exit node is a *bond* (dark solid background); other fragment structures are *rigids* (light solid background). The idea for ordering the activities is to proceed fragment-wise, starting from the root of RPST:

- If a trivial fragment has an activity as exit node, we insert the activity into the order sequence. All other trivial fragments are ignored.
- For a polygon fragments, we traverse the child fragments following the sequential order in the fragment.
- For bond fragments, we traverse the child fragments in an arbitrary order.

- For rigid fragments, we traverse child fragments as follows: starting with the entry node, we perform a depth-first traversal until we reach a node with more than one predecessor. We continue if all of these predecessors that are not reachable from the node itself (via a cycle of flows) have been visited. If not, we backtrack to the first node with multiple successors, for which not all successors have been covered and choose one of these successors randomly.

Similarity Assessment. Using the language models, we measure the similarity for document positions and, thus, activities of the process models, with the Jensen-Shannon divergence (JSD) [15]. Let $p_\mu(t|d,i)$ and $p_\mu(t|d',j)$ be the smoothed language models of two process model documents. Then, the probabilistic divergence of position i in d with position j in d' is:

$$\begin{aligned} jsd(d,d',i,j) &= \frac{1}{2}\sum_{t\in\mathcal{T}} p_\mu(t|d,i)\lg\frac{p_\mu(t|d,i)}{p^+(t)} + \frac{1}{2}\sum_{t\in\mathcal{T}} p_\mu(t|d',j)\lg\frac{p_\mu(t|d',j)}{p^+(t)} \\ \text{with}\quad p^+(t) &= \frac{1}{2}(p_\mu(t|d,i) + p_\mu(t|d',j)) \end{aligned} \tag{4}$$

When using the binary logarithm, the JSD is bound to the unit interval $[0,1]$, so that $sim(d,d',i,j) = 1 - jsd(d,d',i,j)$ can be used as a similarity measure.

Derivation of Correspondences. Finally, we derive correspondences from a similarity matrix over activities, which is known as second line matching [16]. Different strategies may be followed, guided by similarity values and ensuring that selected correspondences adhere to certain constraints. In our experiments, we rely on two strategies, i.e., *dominants* and *top-k*, see [16]. The former selects pairs of activities that share the maximum similarity value in their row and column in the similarity matrix. The latter selects for each activity in one model, the k activities of the other process that have the highest similarity values.

4 Evaluation

We first discuss the setup of our evaluation, before turning to the results discussion.

Setup. Our evaluation uses three real-world model collections that were also used in recent evaluations of techniques for process model matching.

BNB. This set was used by Branco et al. [2] and consists of models from the Bank of Northeast of Brazil (BNB). We selected a sample of three model pairs, all of them are in Portuguese and have few activity descriptions.

Electronics Company (Elec). This set was used by Weidlich et al. [7]. It includes three pairs of models taken from a merger in a large electronics manufacturing company, in English and with detailed descriptions of activities.

Municipalities (Muni). This collection, 17 pairs of models in Dutch with short activity descriptions, stems from municipalities in the Netherlands. Based on previous matching results [7], we separate 12 pairs representing easy matching tasks (Muni (easy)) and five pairs that are more challenging (Muni (hard)).

The used models have between 11 and 81 activities (31 on average). The gold standard was established by process analysts in the respective fields. Overall, it includes 560 matches between activity pairs.

Table 1. Results for different spreads for term propagation.

Spread	BNB		Elec		Muni (easy)		Muni (hard)	
	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
0	0.18	0.50	0.12	0.29	0.18	0.73	0.07	0.22
1	0.21	0.65	0.14	0.31	0.19	0.78	0.12	0.39
2	0.22	0.65	0.10	0.23	0.16	0.67	0.12	0.35

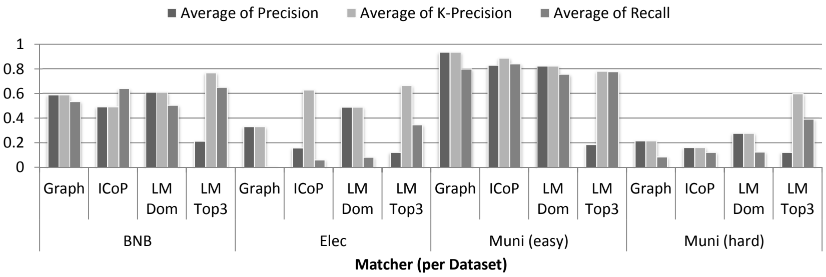


Fig. 4. Comparison to state-of-the-art matchers for process models.

To define evaluation measures, activity pairs in the gold standard are denoted by $\mathcal{C}_h$, those identified by a matcher are denoted by $\mathcal{C}_m$. Besides precision and recall, we also evaluate the extent to which our approach supports the reconciliation of the match result by an expert, which calls for high recall to reduce post-matching effort [17]. To this end, we define *k-precision* that captures in how many cases the top-k pairs proposed for an activity include a correct pair.

Precision is the fraction of selected pairs that are correct, $p = |\mathcal{C}_m \cap \mathcal{C}_h|/|\mathcal{C}_m|$.

k-Precision extends precision to top-k lists, where a match is a top-k list where a correct pair is found: $k - p = |\{(m_1, m_2) \in \mathcal{C}_m \mid \exists\ (hm_1, hm_2) \in (\mathcal{C}_h \cap \mathcal{C}_m) : m_1 = hm_1 \vee m_2 = hm_2\}|/|\mathcal{C}_m|$.

Recall is the fraction of correct pairs that is selected, $r = |\mathcal{C}_m \cap \mathcal{C}_h|/|\mathcal{C}_h|$.

To compare with the state-of-the-art in process model matching, we consider two matchers. A baseline for matching based on activity labels is a graph-edit-distance based matcher (*Graph*) [6]. Second, we use a matcher of the ICoP framework [7] (*ICoP*) that uses a vector space scoring of virtual documents derived for activities and is one of the few matchers that consider activity descriptions.

Results. We first investigate the influence of the spread parameter on the match results. Table 1 (obtained with top-3 selection of correspondences) shows that a spread of 1 yields the best results in most cases. Spreads larger than 2 lower the results since a high spread blurs the characterisation of a passage.

Figure 4 compares the results obtained with existing matchers (*Graph* and *ICoP*) and two matchers based on language models, using the dominance (*LM DOM*) or top-3 (*LM Top3*) strategy for selecting correspondences. Matcher *LM DOM* yields mixed results, slightly improving over the conservative *Graph* matcher in most cases. For datasets well-addressed by existing matchers (*BNB* and *Muni (easy)*), *LM Top3* does not improve the results. However, it achieves large improvements in recall for the challenging datasets *Elec* and *Muni (hard)*, increasing recall up to factor of 5. Although yielding low precision, the k-precision values indicate that for two-thirds of the activities in all datasets, *LM Top3* detects at least one correct corresponding pair. For instance, for dataset *Elec*, this value is comparable to the *ICoP* matcher, which is also geared towards recall. Yet, *LM Top3* identifies 5 times as many correct activity pairs for this dataset.

We conclude that language models provide a new angle for process model matching, leading to improvements for datasets for which existing tools provide poor results. These improvements come at the expense of low precision, so that the presented technique shall be applied for semi-automated matching. The high k-precision values indicate that language model-based matching is indeed suited for this setting. Reflecting on threats to validity, we note that some models had to be remodelled to achieve a common representation. Also, our models may not be representative for all scenarios of process model matching. However, our experiments covered models of three domains and in three languages, so that we expect the observations to generalize for other scenarios as well.

5 Related Work

Recent approaches to process model matching combine techniques for textual comparison of activity labels and measures for structural similarity of process model graphs. The ICoP framework [7] defines a generic architecture for this type of matchers. Following this idea, for instance, the string edit distance for activity labels has been combined with a similarity measure based on the graph-edit-distance [6], which corresponds to the *Graph* matcher in our evaluation. Other work uses the Dice Coefficient with bigrams for textual comparison of activity labels and exploits a parse tree of the process models to guide the matching [2]. Besides syntactical measures, activity labels have been compared based on semantic annotations that are derived by part of speech (POS) tagging. In [8], for instance, POS tagging of activity labels is used for deriving match hypotheses for probabilistic inference of correspondences.

Process models are often used as documentation artefacts and additional textual information is available for matching. To date, this idea was only followed by Weidlich et al. [7], applying a vector space based scoring for virtual documents derived for activities (the *ICoP* matcher in our evaluation). In this work, we took a different approach and defined a language model that allows for integrating structural details of the process model. In comparison to this previous work, our new approach leads to large improvements in recall and k-precision.

6 Conclusions

In this work, we proposed a process matching technique based on a novel combination of positional and passage-based language models. We view a process model as a document, where activity descriptions are ordered passages. We showed how these models are the basis of similarity estimation for activities and selection of correspondences. Our evaluation shows that the presented approach is geared towards high recall, increasing it up to a factor of 5 and identifying about a third of the correct activity pairs. While average precision is low, k-precision values above 60% indicate that the correct activity pairs can be extracted by an expert with reasonable effort, thereby supporting semi-automated matching.

In future work, we intend to focus on the large differences in the results obtained for certain datasets. Here, seeking techniques for predicting the quality of match results is a promising research direction.

References

1. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.: Fundamentals of Business Process Management. Springer (2012)
2. 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
3. Weidlich, M., Mendling, J., Weske, M.: A foundational approach for managing process variability. In: CAiSE. Volume 6741 of LNCS., Springer (2011) 267–282
4. Kunze, M., Weidlich, M., Weske, M.: Behavioral similarity - a proper metric. In: BPM. Volume 6896 of LNCS., Springer (2011) 166–181
5. OMG: Business Process Model and Notation (BPMN) Version 2.0. (2011)
6. Dijkman, R.M., Dumas, M., García-Bañuelos, L., Käärik, R.: Aligning business process models. In: EDOC, IEEE Computer Society (2009) 45–53
7. 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
8. Leopold, H., Niepert, M., Weidlich, M., Mendling, J., Dijkman, R.M., Stuckenschmidt, H.: Probabilistic optimization of semantic process model matching. In: BPM. Volume 7481 of LNCS., Springer (2012) 319–334
9. Bellahsene, Z., Bonifati, A., Rahm, E., eds.: Schema Matching and Mapping. Springer (2011)
10. Croft, W.B., Metzler, D., Strohman, T.: Search Engines - Information Retrieval in Practice. Pearson Education (2009)
11. Song, F., Croft, W.B.: A general language model for information retrieval. In: CIKM, ACM (1999) 316–321
12. Lv, Y., Zhai, C.: Positional language models for information retrieval. In: SIGIR, ACM (2009) 299–306
13. Liu, X., Croft, W.B.: Passage retrieval based on language models. In: CIKM, ACM (2002) 375–382
14. Vanhatalo, J., Völzer, H., Koehler, J.: The refined process structure tree. Data Knowl. Eng. **68**(9) (2009) 793–818
15. Lin, J.: Divergence measures based on the shannon entropy. IEEE Transactions on Information Theory **37**(1) (1991) 145–151
16. Gal, A., Sagi, T.: Tuning the ensemble selection process of schema matchers. Inf. Syst. **35**(8) (2010) 845–859
17. Duchateau, F., Bellahsene, Z., Coletta, R.: Matching and alignment: What is the cost of user post-match effort? - (short paper). In: OTM. Volume 7044 of LNCS., Springer (2011) 421–428