Increasing Recall of Process Model Matching by Improved Activity Label Matching

Christopher Klinkmüller[1,2], Ingo Weber[2,3], Jan Mendling[4], Henrik Leopold[5], and André Ludwig[1]

[1] Information Systems Institute, University of Leipzig, Leipzig, Germany⋆
{klinkmueller,ludwig}@wifa.uni-leipzig.de
[2] Software Systems Research Group, NICTA, Sydney, Australia⋆⋆
ingo.weber@nicta.com.au
[3] School of Computer Science & Engineering, University of New South Wales
[4] Wirtschaftsuniversität Wien, Augasse 2-6, A-1090 Vienna, Austria
jan.mendling@wu.ac.at
[5] Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany
henrik.leopold@wiwi.hu-berlin.de

Abstract. Comparing process models and matching similar activities has recently emerged as a research area of business process management. However, the problem is fundamentally hard when considering realistic scenarios: e.g., there is a huge variety of terms and various options for the grammatical structure of activity labels exist. While prior research has established important conceptual foundations, recall values have been fairly low (around 0.26) – arguably too low to be useful in practice. In this paper, we present techniques for activity label matching which improve current results (recall of 0.44, without sacrificing precision). Furthermore, we identify categories of matching challenges to guide future research.

Keywords: BPM, process similarity, process model matching

1 Introduction

Business process models support analysis, redesign, and implementation projects in enterprises. In various situations, correspondences between different process models have to be found, e.g. when similar processes of recently merged companies have to be identified. The major challenge in such scenarios is the efficient and effective identification of same or similar activities in heterogeneous models.

Recent research has approached the problem of automatically matching activities between process models by adopting techniques from schema and ontology

⋆ The work presented in this paper was partly funded by the German Federal Ministry of Education and Research under the project LSEM (BMBF 03IPT504X) and LogiLeit (BMBF 03IPT504A).

⋆⋆ NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

matching [15, 9]. However, the few studies in this area reveal an issue with recall. This is a serious problem since process matching is usually utilized as decision support. As such it aims to show users an extensive set of potential matches from which they de-select false positives [4]. A prerequisite for applying matching in such a way is a high recall and a big share to be true matches.

This paper contributes to the area of process model matching in a twofold way. First, we present label matching techniques that aim to improve the recall without weakening precision. These techniques are evaluated using established benchmark samples, and yield statistically significant improvements. Second, we conduct a qualitative study towards identifying categories of issues that impede matching performance. Our work not only has implications for process matching research, but also for consistent process modeling altogether.

The paper is structured as follows. Section 2 summarizes prior research and section 3 introduces the techniques for improving recall. Evaluation results and a qualitative analysis are presented in section 4. Section 5 concludes the paper.

2 Prior Research on Process Model Matching

The use of heterogeneous terminology and labels with different levels of details as well as different grammatical structure are challenges, not only to process matching research, but also to practice [1]. The foundations for research in process model matching can be found in various works on process model similarity and ontology matching. Such process similarity techniques exploit different sources of information such as text [3, 7], model structure [6, 2], or execution semantics [8, 16]. Approaches on process model matching directly build on such techniques and combine them in different ways. For example, the ICoP framework defines a generic architecture for assembling and combining different matchers [15]. It, for instance, integrates the graph-based matcher from [2] and the Levenshtein distance [11]. The semantic matcher proposed in [9] relies on Markov logic networks and on an approach to derive semantic match hypotheses from model pairs. Therefore, they apply a label decomposition approach [10] to annotate each activity with action, business object, and additional fragment. Based on the semantic comparison of these components with techniques from ontology matching [5], such as the Lin metric [12], semantic match hypotheses are computed. These hypotheses then serve as input for the Markov model. Although these approaches include several similarity measures and apply complex mechanisms to compute the best matching constellation, they do not only achieve low recall values of around 0.26.

3 Activity Label Similarity

We now discuss techniques for matching activities based on their labels. Therefore, we introduce a basic process matching algorithm. Subsequently, we describe two variations of this algorithm called *Bag-of-Words* and *Label Pruning*.

Basic Process Matching Algorithm. Algorithm 1 presents our basic procedure to compute activity matches between two process models p_1, p_2. As we do not consider structural properties of process models for process matching, we simply refer to a process model as a set of activities $p \in \mathcal{P}(A)$. Furthermore, each activity is given a label which is returned by the function $\lambda : A \rightarrow \mathcal{L}$.

First, the function *createSimilarityMatrix* calculates similarity scores of all activity pairs as $sim.\lambda(a_1, a_2)$, where $a_1 \in p_1$, $a_2 \in p_2$. $sim.\lambda = 0$ implies complete dissimilarity, $sim.\alpha = 1$ means that the two words are identical, and in between are degrees of similarity. Next, the algorithm selects all activity pairs whose similarity score is above a threshold, and proposes them as matches.

Algorithm 1. Basic process matching algorithm (pseudocode)

```
map(Process p1, Process p2, double threshold) {
  SimilarityMatrix sim = createSimilarityMatrix(p1,p2);
  MatchList matches = emptyMatchList();
  while (highestScore(sim) >= threshold) {
    ActivityPair match = getPairWithHighestScore(sim);
    addMatch(matches, match);
    removeMatchFromMatrix(sim, match);
  }
  return matches;
}
```

The function $sim.\lambda$ constitutes the crucial point as it defines to which degree two activities are similar. Hence, we two variants of $sim.\lambda$ are introduced below.

Bag-of-Words. The first variant adopts the bag-of-words technique, where we treat each label as a set of words – and do not further consider the structure of the label. The rationale for neglecting label structure is that the brevity of labels makes it hard to deduce information like word forms. In this way, we aim to offer a means to find matches like "prepare online application" vs. "apply online".

In order to define the bag-of-words similarity, a *tokenize* function is introduced as $tok : \mathcal{L} \rightarrow \mathcal{P}(\mathcal{W})$, from the set of labels, $\mathcal{L}$, to the powerset of words $\mathcal{P}(\mathcal{W})$. This function splits a label into its individual words, and removes common *stop words* like "the", "if", and "to". Then, the label similarity $sim.\lambda$ is computed by comparing the tokenized words of both labels, using a word similarity function $sim.\omega : (\omega_1, \omega_2) \rightarrow [0..1]$ which has the same properties as *actsim*. Note that we evaluate concrete implementations of $sim.\omega$ in section 4. In the basic variant, $sim.\lambda_b$, we aggregate these values by determining the maximum similarity score for each word and calculating the mean over these values.

Definition 1 (Basic bag-of-words similarity). *Let p_1, p_2 be two processes, and $a_1 \in p_1$, $a_2 \in p_2$ be two activities. We define $\Omega^1 := tok(\lambda_1(a_1))$, $\Omega^2 := tok(\lambda_2(a_2))$ as tokenized lists of words contained in the labels. The basic bag-of-word similarity $sim.\lambda_b(a_1, a_2)$ is then defined as:*

$$sim.\lambda_b(a_1, a_2) := \frac{\sum_{i=1}^{|\Omega^1|} max_{j=1}^{|\Omega^2|}(sim.\omega(\omega_i^1, \omega_j^2)) + \sum_{j=1}^{|\Omega^2|} max_{i=1}^{|\Omega^1|}(sim.\omega(\omega_i^1, \omega_j^2))}{|\Omega^1| + |\Omega^2|}$$

Label Pruning. The second technique for label similarity builds on $sim.\lambda_b$, but attempts to better capture activity labels with a strong difference in specificity. This extension called $sim.\lambda_p$ prunes words from the longer label. Thus, in cases where $|\Omega^1| > |\Omega^2|$ (without loss of generality), e.g. "rank application on scale of 1 to 10" vs. "rank case", $sim.\lambda_p$ only considers $|\Omega^2|$-many words of Ω^1.

First, we introduce a generic function $pru : \mathcal{P}(\mathcal{W}) \times \mathcal{P}(\mathcal{W}) \rightarrow \mathcal{P}(\mathcal{W})$. It returns a set of words extracted from its first input: $pru(\lambda_1(a_1), \lambda_2(a_2))$ is Ω^1 iff $|\Omega^1| \leq |\Omega^2|$, or a subset of Ω^1 of size $|\Omega^2|$ otherwise. Criteria for choosing the words to prune from Ω^1 are introduced below the generic definition of $sim.\lambda_p$.

Definition 2 (Bag-of-words similarity with label pruning). *Let p_1, p_2 be two processes, $a_1 \in p_1$, $a_2 \in p_2$ two activities, and $\Omega^1 := tok(\lambda_1(a_1))$, $\Omega^2 := tok(\lambda_2(a_2))$ tokenized lists of words contained in the labels. Further, $pr_1 = pru(\Omega^1, \Omega^2)$ and $pr_2 = pru(\Omega^2, \Omega^1)$ are the pruned lists of words. The bag-of-words similarity with label pruning $sim.\lambda_p(a_1, a_2)$ is then defined as:*

$$sim.\lambda_p(a_1,a_2) := \frac{\sum\limits_{i=1}^{|\Omega^1|} max_{j=1}^{|\Omega^2|}(sim.\omega(pr_1^i, pr_2^j)) + \sum\limits_{j=1}^{|\Omega^2|} max_{i=1}^{|\Omega^1|}(sim.\omega(pr_1^i, pr_2^j))}{2 \times min(|\Omega^1|, |\Omega^2|)}$$

We consider three variants of pru. The first variant, pru_{max}, calculates the similarity scores for all word pairs, as well as the maximal score for each word in $|\Omega^1|$. $pru_{max}(\Omega^1, \Omega^2)$ returns the $|\Omega^2|$-top-scoring words from Ω^1. The second and the third variant rely on numerical statistics for the occurrence of a word (or term) t in a collection of documents $\mathcal{D}$, called *document frequency* (df). The df measure is defined as $\frac{f_t}{|\mathcal{D}|}$, where f_t is the number of documents containing t. In our context, an activity label is considered a document, but we provide two variants for determining which documents are considered part of the collection. One variant takes all activity labels of all models in the model collection as part of the document pool. This variant is called pru_{coll}. In the other variant, only the activity labels of the two models being compared form the document pool. This variant is called pru_{2p}. In both cases the $|\Omega^2|$ words from Ω^1 with the highest df are selected. Applying df, we consider words occuring more often as more important for activity matching.

4 Evaluation

In this section, we evaluate the introduced matching techniques. First, we describe the evaluation's setup including the data set and parameter sampling. Then, the results are presented with focus on precision and recall. Next, we provide a qualitative result analysis. Finally, we discuss the findings and their implications.

Setup. In order to achieve comparability, we used the data set from [9] containing a *process model collection* of nine admission processes of German universities which are publicly available[6]. The other part of the evaluation data is a *process matching standard* which was also used in [9]. It defines normative 1:1 activity matches for all 36 possible pairs in the collection.

To evaluate the quality of a matching technique, each 1:1 match found by the technique can be classified as true-positive (TP), true-negative (TN), false-positive (FP), or false-negative (FN) – with respect to the standard. Based on this classification the standard measures of *precision (P)* (TP/(TP+FP)), *recall (R)* (TP/(TP+FN)), and F_1 *measure* as harmonic mean between P and R ($2 \times P \times R/(P+R)$) can be computed for each model pair. We measure overall quality for a given technique as the mean and standard deviation of these three values over the set of process pairs.

In the evaluation, we examined different parameter configurations for the basic process matching algorithm and both label similarity scores. We sampled *threshold* over the interval [0..1] in steps of 0.05. Furthermore, we employed the following variants for $sim.\omega$:

1. Levenshtein ($sim.\omega_{lev}$): based on the Levenshtein distance [11]
2. Lin ($sim.\omega_{lin}$): a semantic notion [12] based on WordNet [14]
3. Levenshtein-Lin-Max ($sim.\omega_{max}$) the maximum of $sim.\omega_{lev}$ and $sim.\omega_{lin}$
4. Stemmed versions of the former ($sim.\omega_{s.lev}$, $sim.\omega_{s.lin}$, $sim.\omega_{s.max}$): which apply word stemming [13] and in particular the stemming algorithm in the state-of-the-art tool *MIT Java Wordnet Interface*[7] to their stems.

Results. Table 1 summarizes the evaluation results. The first two rows list the results from [9] whereby *ICoP* refers to a matching approach based on the ICoP framework and *Markov* to the one relying on Markov Logic (cf. Section 2).

The next two rows outline the results for the basic process matching algorithm in combination with the basic bag-of-words similarity. The first row shows the best parameter configuration when applying word similarity functions without stemming, while the second row presents the best stemming variant. Note, that "best" refers to the highest F_1 value obtained using the parameter sampling explained above. There are two important observations. First, the variant with stemming outperformed Markov and ICoP regarding precision (0.748), recall (0.299) and F_1 (0.363). Second, the application of stemming helped to improve the F_1 value (0.372) due to higher precision (0.808) and recall (0.304).

The last three rows represent the best results for the basic process matching algorithm in combination with each of the three pruning variant. All pruning variants yield higher F_1 measures than the best basic bag-of-words variants. The best F_1 measure (0.409) was yielded by the document frequency variant using the whole model collection (pru_{coll}). This variant also yielded the highest recall (0.450), while the variant based on the maximal similarity scores yielded the highest precision (0.735).

[6] http://www.mendling.com/Admission_Processes_BPM2012_Leopold_et_al.zip

[7] http://projects.csail.mit.edu/jwi/

Table 1. Evaluation results for variants of bag-of-words similarity

variant	precision	stddev.	recall	stddev.	F_1	stddev.	*threshold*	$sim.\omega$	prune
Markov	0.421	0.217	0.263	0.170	0.315	0.182	-	-	-
ICoP	0.506	0.309	0.255	0.282	0.294	0.253	-	-	-
$sim.\lambda_b$	0.748	0.254	0.299	0.282	0.363	0.249	0.75	*max*	-
$sim.\lambda_b$	0.808	0.241	0.304	0.281	0.372	0.247	0.75	*s.lev*	-
$sim.\lambda_p$	0.735	0.235	0.331	0.279	0.393	0.245	0.75	*s.lev*	*max*
$sim.\lambda_p$	0.468	0.253	0.450	0.256	0.409	0.179	0.70	*s.lin*	*coll*
$sim.\lambda_p$	0.689	0.259	0.356	0.287	0.407	0.242	0.80	*s.lev*	$2p$

Qualitative Analysis: Matching Challenges. To identify challenges in matching activity labels we conducted, a qualitative analysis based on data collected during the above evaluation. For the admission data set, we considered all matches found by the best configuration as well as all matches contained in the gold standard – a total of 912 matches comprising 223 true positives (TP), 381 false positives (FP) and 308 false negatives (FN). In an iterative process of manual coding and clustering, we derived a list of *matching challenge categories*. This process involved three researchers in clustering reasons and resolving different opinions in discussions. We explain the four major categories below – specificity of labels, wording, term semantics and process structure.

1. Different specificity in labels: This class refers to the degree of information provided by a label. We found a difference in the *detail of information*, a.o., when one of the activities is described in more detail than the other. There are problems with *implicit objects*, i.e. when the object of consideration is assumed to be known from the context of an activity, and thus omitted. There are also *higher-level activity* challenges, where one activity in the first process corresponds to multiple activities in the second process, or activities in both processes refer to the same higher-level activity. Finally, *action/object combinations* are challenging when one of the activities contains a list of actions or objects.
2. Other wording challenges: Challenges in this class refer to words. The *domain specificity* can be a problem. Second, *abbreviations* are sometimes used in labels. Third, the action is the same but *different conditions* might apply. Fourth, similar issues are expressed with similar words but different *sentence structure*. Fifth, one of the labels may be the *inverse* of the other.
3. Challenges from term semantics: The comparison of labels depends on the meaning of words. We identified several problems regarding the interpretation of words. A concept can be expressed by a *compound word*. A word might have *spelling errors*. There exist *semantic relations* between the concepts represented by words, like homonyms and antonyms.
4. Process structure-related challenges: Control flow characteristics may challenge activity matches. First, activities with similar labels may appear at different *control flow positions*. Second, activities may be performed by *different roles* that are not modeled. Third, processes use non-consensual *case differentiation*.

Table 2. Matching challenge classification, ordered by number of occurrences (#)

class	challenge	#	FP	FN	class	challenge	#	FP	FN
1	detail of information	463	0.35	0.51	4	different roles	75	1.00	0.00
3	compound words	412	0.59	0.26	4	case differentiation	59	0.24	0.49
1	implicit object	290	0.38	0.48	2	abbreviations	27	0.11	0.82
2	different conditions	249	0.48	0.36	2	domain specificity	25	0.40	0.56
1	higher-level activity	223	0.05	0.87	3	spelling errors	21	0.29	0.57
3	semantic relation	136	0.22	0.54	2	sentence structure	17	0.77	0.00
4	control flow position	120	1.00	0.00	2	inverse	9	0.67	0.33
1	action/object combinations	99	0.37	0.46					

The results of the analysis are summarized in Table 2. For each challenge the table shows how often it was identified (#) and the relative appearance in false positive (FP) and false negative (FN) matches – note that a match can pose multiple challenges. The most striking problems are apparently *detail of information* and *compound words*. Overall, challenges regarding the label specificity appear to constitute the biggest source of errors, while challenges related to the process structure and other wording issues seem to occur least often.

Discussion. The evaluation shows that we were able to outperform the results of the two state-of-the-art approaches from previous research by applying our label based matching techniques. Most of the gains in recall can be attributed to the general design decision to employ a bag-of-words technique. This is in contrast to prior research where the label structure is explicitly utilized [9]. Disregarding the label structure alone already yielded improvements in our evaluation, with word stemming and pruning providing further gains.

Our post-hoc analysis of false positive and false negative match proposals provides a good basis for future innovations in process model matching. Detail of information and compound words are difficult problems, in particular as their resolution has to rely on less semantic context and text structure as in general natural language processing. There are also problems that are apparently specific to process models. The identification of implicit objects and roles may offer opportunities for further improvements.

However, the validity of our results is clearly restricted by the size of the data set used in the evaluation. Linked thereto is the threat to validity that we did not distinguish between training and evaluation data. A clear separation of data for development and evaluation purposes prevents the development of techniques well suited for a certain data set. Thus, enlarging the evaluation data set is an important step to substantiate our findings in future work.

5 Conclusion

In this paper, we presented techniques for improving process activity matching. In particular, our focus is on activity labels, so as to increase recall of matches when applied to realistic process model collections. Our comparative evaluation shows that we achieved significant improvements: recall increased by around 0.2

to 0.445. Driven by this outcome, we analyzed what makes label matching hard, and categorized the challenges into 4 classes over 15 categories in total.

In future work, we plan to pursue two directions regarding the improvement of process matching: investigating additional techniques for considering process structure, both from literature and new approaches, as well as further improving label matching. To substantiate our findings we will also work on an enlarging our evaluation data set.

References

1. M. C. Branco, J. Troya, K. Czarnecki, J. M. Küster, and H. Völzer. Matching business process workflows across abstraction levels. In *MODELS 2012*.
2. R. Dijkman, M. Dumas, and L. García-Bañuelos. Graph matching algorithms for business process model similarity search. In *BPM 2009*.
3. R. Dijkman, M. Dumas, B. van Dongen, R. Käärik, and J. Mendling. Similarity of business process models: Metrics and evaluation. *Inf. Syst.*, 36(2):498–516, 2011.
4. F. Duchateau, Z. Bellahsene, and R. Coletta. A flexible approach for planning schema matching algorithms. In *COOPIS 2008*.
5. J. Euzenat and P. Shvaiko. *Ontology Matching*. Springer, 2007.
6. D. Grigori, J. C. Corrales, and M. Bouzeghoub. Behavioral Matchmaking for Service Retrieval. In *IEEE ICWS 2006*.
7. A. Koschmider and E. Blanchard. User assistance for business process model decomposition. In *IEEE RCIS 2007*.
8. M. Kunze, M. Weidlich, and M. Weske. Behavioral similarity - a proper metric. In *BPM 2011*.
9. H. Leopold, M. Niepert, M. Weidlich, J. Mendling, R. Dijkman, and H. Stuckenschmidt. Probabilistic optimization of semantic process model matching. In *BPM 2012*.
10. H. Leopold, S. Smirnov, and J. Mendling. On the refactoring of activity labels in business process models. *Inf. Syst.*, 37(5):443–459, 2012.
11. V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady.*, 10(8):707–710, 1966.
12. D. Lin. An information-theoretic definition of similarity. In *ICML 1998*.
13. J. B. Lovins. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11:22–31, 1968.
14. G. A. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.
15. M. Weidlich, R. Dijkman, and J. Mendling. The icop framework: identification of correspondences between process models. In *CAiSE 2010*.
16. H. Zha, J. Wang, L. Wen, C. Wang, and J. Sun. A workflow net similarity measure based on transition adjacency relations. *Computers in Industry*, 61(5):463 – 471.