Using Hidden Markov Models for the Accurate Linguistic Analysis of Process Model Activity Labels

Henrik Leopold[a,b,*], Han van der Aa[c], Jelmer Offenberg[d], Hajo A. Reijers[e,f]

[a]*Kühne Logistics University, Großer Grasbrook 17, 20457 Hamburg, Germany*
[b]*Hasso Plattner Institute, University of Potsdam, Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Germany*
[c]*Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany*
[d]*Vrije Universiteit Amsterdam, De Boelelaan 1081, 1081HV Amsterdam, The Netherlands*
[e]*Utrecht University, Domplein 29, 3512 JE Utrecht, The Netherlands*
[f]*Eindhoven University of Technology, PO Box 513, 5600MB Eindhoven, The Netherlands*

Abstract

Many process model analysis techniques rely on the accurate analysis of the natural language contents captured in the models' activity labels. Since these labels are typically short and diverse in terms of their grammatical style, standard natural language processing tools are not suitable to analyze them. While a dedicated technique for the analysis of process model activity labels was proposed in the past, it suffers from considerable limitations. First of all, its performance varies greatly among data sets with different characteristics and it cannot handle uncommon grammatical styles. What is more, adapting the technique requires in-depth domain knowledge. We use this paper to propose a machine learning-based technique for activity label analysis that overcomes the issues associated with this rule-based state of the art. Our technique conceptualizes activity label analysis as a tagging task based on a Hidden Markov Model. By doing so, the analysis of activity labels no longer requires the manual specification of rules. An evaluation using a collection of 15,000 activity labels demonstrates that our machine learning -based technique outperforms the state of the art in all aspects.

Keywords: Label Analysis, Process Model, Natural Language, Hidden Markov Models

[*]Corresponding author. *Phone number:* +49 40 328707-203
Email addresses: henrik.leopold@the-klu.org (Henrik Leopold), han.van.der.aa@hu-berlin.de (Han van der Aa), j.j.offenberg@student.vu.nl (Jelmer Offenberg), h.a.reijers@uu.nl (Hajo A. Reijers)

1. Introduction

The considerable number of process models in many organizations has led to the development of a wide range of automated analysis techniques. A large number of these techniques provides support to check the correctness of process models [1, 2, 3] and to determine different types of quality criteria [4, 5, 6]. Some techniques also support organizations with more advanced analyses. For example, there are techniques that automatically discover service candidates from process models [7, 8, 9] or automatically determine where process models from different organizations overlap [10, 11].

What many of these techniques have in common is that they depend on an accurate analysis of the natural language contents of process model activity labels. In this context, the automatic decomposition of activity labels into their semantic components has been proven to be particularly helpful [12]. As pointed out in [13, 6], activity labels can comprise three semantic components: an *action*, a *business object* on which the action is performed, and an *addition* that is providing further details. As an example, consider the the activity label "*Product delivery to client*", which consists of the action "*deliver*", the business object "*product*", and the addition "*to client*". These components provide valuable input to analysis techniques. Service discovery techniques, for instance, can use the information about the components to cluster activities based on their actions and business objects [9, 14]. In a similar way, techniques determining the overlap between two processes are capable of explicitly comparing actions and business objects of the different activities to reason about their similarity [15, 10, 16].

However, the automated decomposition of activity labels into their semantic components is a complex task. Due to the shortness and the varying grammatical styles of activity labels, standard natural language processing tools such as parsers and taggers do not deliver sufficiently accurate results [12]. One reason for this is that they are not able to recognize when nouns, such as "*delivery*" in "*Product delivery to client*", actually play the role of an action. Therefore, prior research has proposed a dedicated technique for decomposing process model activity labels [17]. While this technique delivers an overall satisfying performance, it also suffers from a severe limitation: It is a rule-based technique. Due to this, its performance varies considerably among data sets with different characteristics. Moreover, it is not capable to detect less frequently occurring grammatical patterns, which limits its use in practice. Finally, and perhaps most importantly, any adaptation of the technique, for example because of a different context or a different target language, requires in-depth domain knowledge.

Recognizing these limitations, we propose in this paper a machine learning-based technique for activity label analysis that overcomes the issues of existing techniques. Our technique conceptualizes activity label analysis as a tagging task based on a Hidden Markov Model (HMM) such that it no longer requires the manual specification of rules. In this way, our technique is more flexible, more stable, and also better capable to detect less frequent grammatical patterns than the state of the art. We perform a direct comparison with the rule-based approach from [17] to demonstrate that our machine learning approach outperforms the state-of-the-art technique in all aspects.

The remainder of this paper is organized as follows. Section 2 introduces the problem in more detail and highlights the challenges our approach needs to address. Section 3 specifies the grammatical structure of activity labels. Section 4 presents our machine learning-based technique for activity label analysis. Section 5 presents the evaluation of our technique. Section 6 reflects on the limitations of our technique. Section 7 discusses related work before Section 8 concludes the paper.

2. Problem Illustration

The automated analysis of natural language represents, in general, a notable challenge. The main reason for this lies in its complexity: The English language encompasses more than 300,000 words, which can be combined to proper sentences in countless ways. Yet, there are highly accurate tools available which can, for instance, detect the parts of speech of words in natural language texts [18, 19]. However, as illustrated in prior work [12], these developments cannot be directly transferred to process models. The natural language used in process models simply differs from the natural language used in in standard natural language texts. Most notably, because process model activity labels do not necessarily use verbs to convey actions.

To illustrate the use of natural language in process model activities and the challenges that

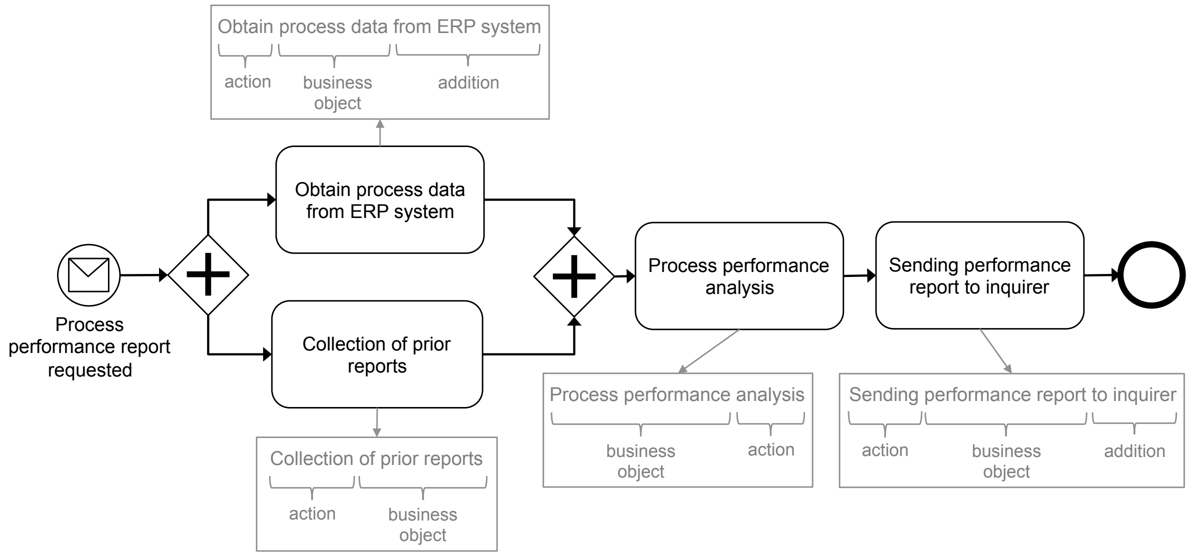

Obtain process data from ERP system
action
business object
addition
Obtain process data from ERP system
Process performance report requested
Collection of prior reports
Process performance analysis
Sending performance report to inquirer
Collection of prior reports
action
business object
Process performance analysis
business object
action
Sending performance report to inquirer
action
business object
addition

Figure 1: Process model illustrating the challenges of activity label analysis

are associated with its analysis, consider the process model in Figure 1. The depicted process is concerned with delivering a process performance report and is triggered when a respective request is received. Then, two concurrent activities take place: The process data is obtained from the ERP system and prior reports are collected. Once both activities have been completed, the process performance analysis is conducted. Finally, the performance report is sent to the inquirer.

Besides the process model, Figure 1 also provides information on the semantic components of the activities. Each activity contains an action, a business object on which the action is performed, and an optional addition fragment, which is providing further details. The process model from Figure 1 illustrates that these components can occur in different linguistic variations. For instance, in the activity "*Obtain process data from ERP system*" the action occurs as an imperative verb at the first position of the label, followed by the business object and the addition fragment. In the activity "*Sending performance report to inquirer*" the action is also positioned at the beginning, but provided as a gerund. In "*Collection of prior reports*" and "*Process performance analysis*" the actions are provided as nouns at the first and the last position of the label. The positions of the business objects differ respectively.

These exemplary activity labels illustrate why the recognition of the semantic activity components is so challenging: First, activity labels are very short and, thus, only provide little context for automated analysis techniques. Second, actions can occur as nouns, which results in considerable ambiguity. As an example, consider the activity "*Process performance analysis*". An unsophisticated algorithm could recognize the word "*process*" as an imperative verb and thus erroneously extract "*process*" as action instead of "*analysis*". This is caused by a frequently occurring phenomenon that is referred to as *zero derivation ambiguity*, which means that syntactically identical words can represent a verb as well as a noun. While a first solution for addressing these challenges has been proposed in prior research [17], the technique suffers from a number of limitations that reduce its value for an application in practice. Because the technique is governed by a specific set of rules, its performance varies among data sets with different characteristics and, therefore, fails to recognize less frequent grammatical patterns. What is more, any adaption of the technique requires in-depth domain knowledge. This motivates us to present a machine-learning technique for activity label analysis. More specifically, we employ a Hidden Markov Model to approach the recognition of semantic activity components as a *tagging task*: Our technique associates each word of a given activity label with a corresponding tag describing the semantic role of the word.

As the development of such a tagging technique requires a solid understanding of the grammatical structure of activity labels, the next section introduces the peculiarities of the natural language of activity labels in detail.

3. Grammatical Structure of Activity Labels

The grammatical structure of activity labels has been subject of various works [20, 17, 6]. The main finding is that activity labels follow regular structures, which can be described using a set of *labeling styles*. In each labeling style, the action is captured in a different way. Table 1 summarizes the seven existing activity labeling styles. It shows the structure of each style and an example. Note that the additional fragment is always optional.

In *verb-object* labels, the verb is given as an imperative verb at the beginning of the label, followed by the business object and an optional addition. As examples, consider "*Create invoice*" or "*Notify Customer*". In *action-noun* labels, the action is provided as a noun. Since there are different possibilities for doing so, there are four different sub styles. The first is the *action-noun*

Table 1: Activity labeling styles

Labeling Style	**Structure**	**Example**
Verb-Object VO	$\mathrm{A}_{\mathrm{Imperative}}$ + BO (+ ADD)	"*Create invoice*"
Action-Noun AN (NP)	BO + $\mathrm{A}_{\mathrm{Noun}}$ (+ ADD)	"*Invoice creation*"
Action-Noun AN (ING)	$\mathrm{A}_{\mathrm{Gerund}}$ + BO (+ ADD)	"*Creating invoice*"
Action-Noun AN (OF)	$\mathrm{A}_{\mathrm{Noun}}$ + "*of*" + BO (+ ADD)	"*Creation of invoice*"
Action-Noun AN (IRR)	*anomalous*	"*Invoice: Creation*"
Descriptive DES	(Role +) A_{3P} + BO (+ ADD)	"*Clerk creates invoice*"
No-Action NA	*anomalous*	"*Invoice*"

style (NP). Here, the nominalized action is provided at the end of the label as in "*Invoice creation*". In labels following the *action-noun style (ING)* the action occurs as a gerund at the beginning of the label such as in "*Creating invoice*". In labels following the *action-noun style (OF)* the preposition "*of*" is used to separate the nominalized action from the business object. As examples, consider "*Creation of invoice*" or "*Notification of customer*". All labels that contain a nominalized action but cannot be assigned to one of the three previously introduced styles are categorized as *action-noun (IRR)*. In *descriptive* activity labels, the action is a verb in the third person form. In many cases, a role is mentioned at the beginning of the label. Examples are "*Clerk creates Invoice*" or "*Customer approves order*". All labels that do not contain any action are assigned to the *no-action style*. Examples typically include single nouns such as "*Invoice*" or "*Error*".

Based on these labeling styles, we define an approach for the automatic analysis of activity labels in the next section.

4. Analyzing Activity Labels Using Hidden Markov Models

To automatically tag activity labels with their semantic components, we employ a Hidden Markov Model (HMM). In general, HMMs are used to relate a sequence of observations to a sequence of hidden states. In our setting, this means that we can use HMMs to relate a sequence of words from an activity label to a sequence of corresponding semantic components. To achieve this, HMMs determine the probability of all possible hidden state sequences and select the most likely one.[1] Given an observation sequence O, the probability of a particular hidden state sequence

[1] For further details about the technical aspects of HMMs, we refer the reader to [21].

Q, i.e., probability $P(O, Q)$, is calculated as follows:

$$P(O, Q) = P(Q) \times P(O \mid Q) \tag{1}$$

Equation 1 illustrates that $P(O, Q)$ results from multiplying two separate probabilities:

1. The probability of encountering the particular sequence of hidden states Q, given as $P(Q)$. This probability is computed as the Markovian probability that a hidden state sequence starts with a particular state at index 0, i.e., the state q_0, multiplied by the probability that each subsequent state q_i follows its preceding state q_{i-1}, i.e., $P(Q) = P(q_0) \times \prod_{i=1}^{n} P(q_i \mid q_{i-1})$, where $n = |O|$;

2. The probability of encountering that the observations from O are associated with the specific hidden states from Q, given as $P(O \mid Q)$. This probability is computed as $\prod_{i=1}^{n} P(o_i \mid q_i)$, where $P(o_i \mid q_i)$ denotes the probability that the observation o_i corresponds to the state q_i.

The aforementioned probabilities are all derived from probability matrices, which are constructed based on training data. The probabilities used to compute $P(Q)$ stem from a so-called *probability transition matrix* and the probabilities associated with $P(O \mid Q)$ from a so-called *emission probability matrix*. In the following, we elaborate on the details of our HMM for tagging activity labels. In Section 4.1, we first describe the tag set we developed. In Sections 4.2 and 4.3, we describe the transition probability matrix and the emission probability matrix and how we obtain them from training data. Finally, in Section 4.4, we discuss how we adapted our HMM to deal with the limited availability of real-world process models as training data.

4.1. Tag Set for Activity Labels

A core aspect of applying HMMs is the establishment of a *tag set*. Each tag in such a set corresponds to a particular hidden state that can be associated with an observation. For example, given the label "*Create invoice*", with the observation sequence $O = \langle create, invoice \rangle$, we require tags to indicate that "*create*" corresponds to the *verb* in a verb-object label and that "*invoice*" corresponds to the *business object* in this labeling style. The design of a suitable tag set plays an important role for the performance of our technique. If the tags are too general, the HMM will fail to recognize certain patterns. If the tag set is too specific, the HMM will require an increasing amount of training data to extract reliable probabilities. Therefore, we establish a tag set that includes tags for each possible semantic component that may be contained in labels, e.g., actions, business objects, and modifiers. Additionally, we create separate tags for each of these components per labeling style. Table 2 depicts the tag set that results from this, showing the tags for each (possible) combination of 8 *tag items* with the 7 labeling styles.

Table 2: Tag set for activity labels

Tag Item	**AN (NP)**	**AN (ING)**	**AN (OF)**	**AN (IRR)**	**VO**	**DES**	**NA**
Action	A-NP	A-ING	A-NOF	A-IRR	A-VO	A-DES	-
Business Object	BO-NP	BO-ING	BO-OF	BO-IRR	BO-VO	BO-DES	BO-NA
Addition Object	AO-NP	AO-ING	AO-OF	AO-IRR	AO-VO	AO-DES	AO-NA
Modifier	MOD-NP	MOD-ING	MOD-OF	MOD-IRR	MOD-VO	MOD-DES	-
Connective	C-NP	C-ING	C-OF	C-IRR	C-VO	C-DES	C-NA
Prepositions	P-NP	P-ING	P-OF	P-IRR	P-VO	P-DES	P-NA
Special Character	SC-NP	SC-ING	SC-OF	SC-IRR	SC-VO	SC-DES	SC-NA
Number	NUM-NP	NUM-ING	NUM-OF	NUM-IRR	NUM-VO	NUM-DES	NUM-NA

The columns correspond to the seven modeling styles denoted in Table 1, where AN refers to the four *Action-Noun* styles, respectively using *noun phrases* (NP), *gerunds* (ING), *of-based* constructs (OF), and *irregular* constructs (IRR). Furthermore, VO refers to *Verb Object*, DES to *descriptive*, and NA to *no-action* style.

As shown in the table, we define for each style respective tags for actions, business objects, and objects that are part of an additional fragment. Consequently, the tag *A-NP* denotes an action of an action-noun (np) label, the tag *A-ING* denotes an action of an action-noun (ing) label, and so forth. What is more, we introduce tags for action modifiers (e.g. "*automatic*" or "*manually*"), connectives (e.g. "*and*" or "*or*"), prepositions (e.g. "*for*" and "*in*"), special characters (e.g. "*/*" and "*&*"), and numbers. The following activity labels from the process model shown in Figure 1 illustrate the use of the tag set.

$$\underbrace{\text{Obtain}}_{\text{A-VO}}\ \underbrace{\text{process}}_{\text{BO-VO}}\ \underbrace{\text{data}}_{\text{BO-VO}}\ \underbrace{\text{from}}_{\text{P-VO}}\ \underbrace{\text{ERP}}_{\text{AO-VO}}\ \underbrace{\text{system}}_{\text{AO-VO}}$$

$$\underbrace{\text{Process}}_{\text{BO-NP}}\ \underbrace{\text{performance}}_{\text{BO-NP}}\ \underbrace{\text{analysis}}_{\text{A-NP}}$$

$$\underbrace{\text{Collection}}_{\text{A-OF}}\ \underbrace{\text{of}}_{\text{P-OF}}\ \underbrace{\text{prior}}_{\text{AO-OF}}\ \underbrace{\text{reports}}_{\text{AO-OF}}$$

Besides the general use of the tag set, these examples also highlight the reason for introducing separate tags for each style. Consider the first activity label, which follows the verb-object style. From this example, an HMM could mistakenly learn that business objects are likely to follow actions. However, in fact, this mainly applies to verb-object labels. The two examples following different action-noun style patterns illustrate this vividly. To be able to accurately capture these differences among labeling styles, we use labeling style-specific tags.

4.2. Transition Probability Matrix

The transition probability matrix captures the probabilities of moving from one state to another. In the context of tagging activity labels it holds the transition probabilities between tags. To illustrate the notion of a transition probability matrix, consider the visualization in Figure 2. It shows an exemplary, simplified transition probability matrix as a probabilistic finite automaton. The transition probabilities between the states of the automaton denote the probabilities for moving from one tag to another. According to Figure 2, verb-object labels are more likely (70%) than action-noun labels (30%). Furthermore, we see that verb-object labels are likely to have an object (80%) and that half of the verb-object labels with a business object have an addition (which consists of *PRP-VO* and *AO-VO*). For action-noun labels, we observe that they always have an action (they would be no-action labels otherwise) and that about 60% of action-noun labels have an addition.

While the probabilities shown in Figure 2 are artificial, the transition probabilities can be learned by training an HMM on a collection of manually tagged activity labels. In case the training data is fully tagged, the probabilities can be derived by simply counting [22]. If the data is only partially tagged, estimation algorithms can be used to compute the probability distributions. The most common are so-called forward-backward algorithms, such as the Baum-Welch method [21].

4.3. Emission Probability Matrix

The emission probability matrix captures the probability of an observation occurring in a certain state. In the context of tagging activity labels, the matrix indicates the likelihood of a given word to be associated with a certain tag. As an example, consider the activity label "*Process*

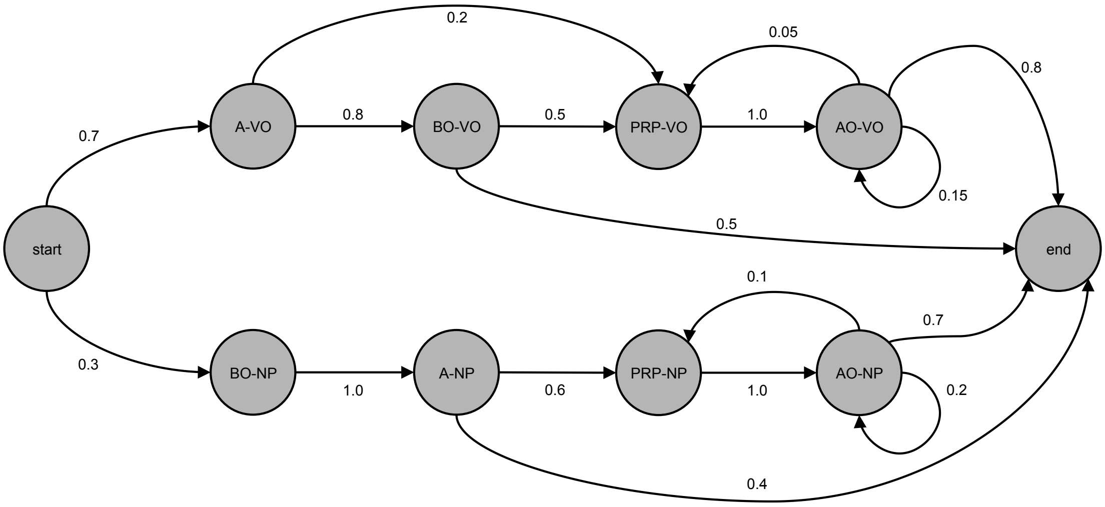


Figure 2: Simplified exemplary probability transition matrix for activity tagging

performance analysis" from Figure 1. Since the word "*process*" suffers from the zero derivation ambiguity, it may represent an action in a verb-object label or a business object in an action-noun (NP) label. Thus, we can expect the emission matrix to return probabilities greater than zero for $P(\text{"}process\text{"}, A{-}VO)$, as well as $P(\text{"}process\text{"}, BO{-}NP)$. However, suppose "*process*" is only rarely used as an action and "*process*" is frequently associated with the tag *A-NP*. Then, the HMM would conclude that the tags corresponding with an action-noun (NP) label are more likely. Note that the final decision will be based on considering both the emission and the transition probabilities. When two alternative tag sequences are equally likely, the probabilities from the emission matrix particularly contribute to the decision of the HMM. Just as the transition probability matrix, the specific emission probabilities have to be learned from manually tagged activity labels.

4.4. Adaption for Dealing with Sparse Training Data

Without adaptions, an HMM that strictly follows the computation from Equation 1 is not likely to perform well. The reason is that $P(O, Q)$ will be zero for any sequence of observations O that contains words that are unknown to the HMM, i.e. not part of the emission probability matrix. This problem can be resolved in two ways: by applying a smoothing technique and by increasing the training data.

Smoothing is an approach that is widely applied in information retrieval [23]. It essentially assigns a small probability to unseen words, such that probabilities of zero no longer occur. To illustrate the effect of smoothing, consider the activity label "*Escalate case*" and suppose the HMM has not seen the word "*case*" before. Without smoothing the probability $P(O, Q)$ would be zero for any sequence of tags. We thus have to expect a random outcome. When applying smoothing, the HMM would assign a small probability to "*case*" being associated with any of tags from our tag set. Since these probabilities would be all equal, the choice for the most likely tag sequence will not be random but be driven by the transition probabilities. Assume the HMM found that "*escalate*" is often tagged as verb-object action (A-VO). Then, the HMM identifies the tag that most likely follows A-VO and assigns it to "*case*". According to the exemplary transition probability matrix from Figure 2, "*case*" will therefore receive the tag *BO-VO*.

While smoothing helps to deal with unseen data, it can still lead to wrong conclusions. A safer way to improve the performance of an HMM therefore is to *increase the amount of training data*. Looking into the size of the data resources that are used to train accurate Natural Language Processing tools, such as the Stanford Tagger [18] and the Stanford Parser [24], we can learn that they use corpora that consist of several million words. Unfortunately, it does not appear to be feasible to obtain such an amount of data for business process models. The contents of process models are typically sensitive, which explains why there are hardly any process model collections publicly available. To deal with this problem of scarce data, we decided to train our HMM on additional, non-process related resources. More specifically, we parsed an English dictionary and extracted 4684 infinitive verbs and 7338 singular nouns. We tagged the former as actions for verb-object labels (i.e. as A-VO) and the latter as business object for every style (i.e. as *BO-A*, *BO-NP*, etc.). What is more, we derived the third person singular, the present participle, and the gerund forms from the verbs and tagged them respectively (i.e., third person singular verbs as *A-DES*, present participle verbs as *A-DES*, and gerund verbs as *A-ING*). In the experiments in the next section, we will show that this augmentation significantly contributes to the performance of our HMM.

5. Evaluation

The goal of the evaluation is to demonstrate that the machine-learning technique presented in this paper outperforms the state of the art in the form of the rule-based technique from [17]. To this end, we run both techniques on a data set that consists of three large process model collections from practice and compare their performance. Section 5.1 first introduces the data set. Section 5.2 then elaborates on the details of the evaluation setup. Sections 5.3 and 5.4 present the results. Section 5.5 discusses the effect the HMM augmentation with non-process related resources. Finally, Section 5.6 assesses the performance of the technique on previously unseen domains.

5.1. Test Data set

To test our technique we build on an extended version of the data set from [17]. In particular, we were able to add almost 700 new process models to the original data set (primarily new models from the telecommunications sector).As a result, the number of activity labels increased from 10,784 to 15,488.

The data collection is well-suited to test the capabilities of our technique since it varies with respect to multiple dimensions such as label style distribution, domain, and expertise of the modelers. It consists of three different real-world process model collections, of which 3 summarizes the main features. They include the following:

- *SAP Reference Model (SAP)*: The SAP Reference Model represents the business processes of the SAP R/3 system in its version from the year 2000 [25, pp. 145-164]. It contains 604 Event-driven Process Chains (EPCs), which are organized in 29 functional branches such as sales and accounting. This collection mainly contains action-noun labels (81%).

- *Telecommunication collection (TC)*: The TelCo Collection contains a set of 1000 ADONIS models from a large telecommunication service provider. With regard to contents, the models capture various aspects from the domain of customer service management. The major share of labels follow the verb-object style (93%).

- *Academic Collection (AC)*: The Academic Collection includes 578 process models created using the Business Process Model and Notation (BPMN). The models cover diverse domains and mainly stem from academic training. Most of the labels follow the verb-object style (74%).

Table 3: Characteristics of test data set

Property	**SAP**	**TC**	**AC**
Process models	604	1000	578
Action-Noun labels	81%	7%	10%
Verb-Object labels	11%	93%	78%
Descriptive labels	0%	0%	7%
No Action labels	8%	0%	5%
Zero-Derivation cases	42%	49%	46%
Activity labels	2,433	8,152	4,903
Average no. of activities per model	4.03	8.15	7.01
Average no. of words per label	3.5	3.51	3.65
Minimum no. of words per label	1	1	1
Maximum no. of words per label	12	16	19
Modeling language	EPC	ADONIS	BPMN

As indicated by the characteristics shown in Table 3, the model collections differ in several dimensions. Most importantly, they differ with respect to the label style distribution, the domain, and the expertise of the modelers.

The most important feature of the considered collections is the opposed *distribution of labeling styles*. While the majority of the activity labels in the SAP Reference Model follow the action-noun style, the TC and the AC collections mainly contain verb-object labels. This is an important feature since it allows us to learn about the ability of our technique to deal with different extremes. It should be noted that the distribution of the labeling styles does not lead to a considerable difference with respect to the zero-derivation cases. All collections have a comparable and considerable share of zero-derivation cases (between 42% and 49%). Another important feature of the test data set is the variety of covered *domains*. Since different domains come with different vocabularies and possibly even varying use of label patterns, the broad range of covered domains helps us the reason about the general applicability of our technique. It is important to note that in particular the SAP collection contains a large of number of highly domain-specific words that are typically not used in standard texts. Finally, the test data set differs with respect to the *expertise of the modelers*. While the SAP Reference Model and the TC collections were created in a professional environment, the AC model collection was mainly created by students. We expect that this difference in modeling expertise affects the way natural language is used in activity labels. To account for this heterogeneity in the context of our evaluation, we include both professional and non professional process model collections.

5.2. Setup

To evaluate the performance of our technique, we manually annotated all activities from our test data set with their label style and the tags from our tag set. Based on this manually annotated and tagged data set, we conducted a stratified 10-fold cross validation. The idea behind this validation approach is to randomly split the data set into 10 mutually exclusive subsets of about equal size and distribution of labeling styles [26]. The HMM is then trained on 9 of the 10 sub sets and tested on the remaining (unseen) sub set. This process is repeated 10 times such that, in the end, all data has been used for both training and testing. The advantage of this evaluation method is that it does not require us to partition the data set into training and testing data. Thus, it neither compromises the training nor the testing possibilities.

To measure and compare the performance of our technique, we evaluate two key capabilities:

- the capability to correctly recognize labeling styles;
- the capability to correctly identify semantic components.

We quantify the *recognition capability* using the metrics precision, recall, and F1-measure. In our context, the precision value is the number of correctly recognized labels of a given style divided by the total number of labels the technique associated with that style. The recall is the number of correctly recognized labels of a given style divided by the total number of labels belonging to this style. Since we aim to obtain high recall and precision values at the same time, we also compute the F1-measure, the harmonic mean of precision and recall. Note that a correct recognition of the label style does not necessarily imply the correct identification of the semantic components. Therefore, we quantify the *identification capability* by computing the share of semantic components that have been identified correctly.

Since the technique from [17] does not require any training, we simply run it on our test data set and compute the two capability metrics. For our machine learning-based technique, we compute the two capability metrics for each fold. By averaging them among all 10 folds, we obtain a realistic picture of its performance.

5.3. Results for Label Style Recognition

The aggregated recognition results for all three process model collections are provided in Table 4. The total numbers for precision, recall, and F1-measure show that our HMM-based technique clearly outperforms the rule-based technique from [17]. The F1-measure achieved by the HMM-based technique is about 3 percentage points higher than the F1-measure of the rule-based technique. This increase results from both an improved recall as well as from an improved precision. However, the numbers also reveal that the increase in precision has a slightly bigger contribution to the improved F1-measure. The consideration of precision, recall, and F1-measure for the individual labeling styles provides further insights. The numbers show that the difference between the techniques can be explained by three main factors. First, the HMM-based technique recognizes action-noun labels with a much higher precision. As discussed in [27], the precise recognition of action-noun labels is particularly challenging when most labels of a collection actually follow the verb-object style. Second, the HMM-based technique is better able to recognize descriptive labels than the rule-based technique. The latter was only able to detect about 25%. At the same time, the precision was very low (0.29). While the HMM has also difficulties with descriptive labels, the results are substantially higher. It recognizes slightly more than 70% of descriptive labels with a moderate precision of 0.63. Third, as opposed to the rule-based technique, the HMM is able to recognize no-action labels. The reason for the failure of the rule-based technique to do so is that it is almost impossible to specify a simple rule to recognize such labels. No-action labels often consist of one or two nouns. Whether these nouns relate to an action is an aspect that the HMM can learn from empirical data. A general rule, however, is unlikely to think of, let alone allow for good predictions.

Table 5 shows the detailed results for each process model collection. The numbers reveal a number of interesting insights. Considering the totals, we can see that the performance of the HMM is much more stable than the performance of the rule-based technique. The F1-measure of the HMM-based technique ranges from 0.90 to 0.97, while the performance of the rule-based

Table 4: Aggregated recognition results

		HMM-based technique			Rule-based technique		
Style	**Count**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**
Action-Noun labels	3,031	0.83	0.82	0.83	0.65	0.80	0.71
Verb-Object labels	11,674	0.96	0.95	0.96	0.86	0.96	0.89
Descriptive labels	343	0.63	0.71	0.66	0.29	0.24	0.22
No-Action labels	440	0.47	0.78	0.58	–	–	–
Weighted average		**0.93**	**0.92**	**0.93**	**0.90**	**0.91**	**0.90**

technique ranges from 0.75 to 0.96. This represents an important finding. Since the specific characteristics of a process model collection can hardly be assessed prior to its analysis, a stable performance of a technique is essential for its application in practice. Our evaluation reveals that the performance of the rule-based technique is subject to considerable variation.

Table 5: Disaggregated recognition results for each collection

		HMM-based technique			Rule-based technique		
	Style	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**
SAP	Action-Noun labels	0.96	0.90	0.93	0.53	0.86	0.66
	Verb-Object labels	0.46	0.75	0.55	0.27	0.86	0.41
	Descriptive labels	0.30	0.07	0.11	0.00	0.00	0.00
	No-Action labels	0.58	0.82	0.67	–	–	–
	Total	**0.92**	**0.89**	**0.90**	**0.74**	**0.75**	**0.75**
TC	Action-Noun labels	0.76	0.77	0.75	0.65	0.75	0.70
	Verb-Object labels	0.98	0.98	0.98	0.98	0.97	0.98
	Descriptive labels	0.93	0.82	0.84	0.06	0.20	0.09
	No-Action labels	0.40	0.40	0.40	–	–	–
	Total	**0.97**	**0.97**	**0.97**	**0.97**	**0.96**	**0.96**
AC	Action-Noun labels	0.70	0.66	0.68	0.53	0.86	0.66
	Verb-Object labels	0.98	0.95	0.96	0.96	0.99	0.97
	Descriptive labels	0.66	0.70	0.68	0.82	0.41	0.55
	No-Action labels	0.32	0.81	0.45	–	–	–
	Total	**0.93**	**0.90**	**0.92**	**0.87**	**0.91**	**0.89**

Despite the better and more stable performance of the HMM-based technique, the numbers also indicate that both techniques suffer from extreme label style distributions. For the SAP collection, which mainly contains action-noun labels, we observe a fairly low recognition performance for verb-object labels. For the TC and the AC collections, which mainly contain verb-object labels, we observe lower numbers for the recognition performance of action-noun labels. The reason for these erroneous classifications can be mainly related to cases of zero-derivation ambiguity. As examples, consider the labels "*Contact maintenance*" and "*Order checkout*". Both labels can be interpreted

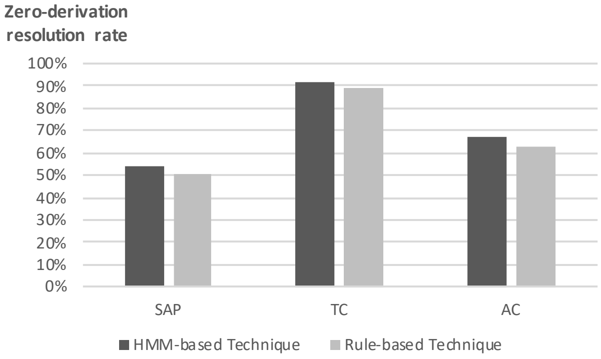


Figure 3: Comparison of zero-derivation resolution ratios

as verb-object as well as action-noun style labels. Neither the HMM-based technique nor the rule-based technique can perfectly deal with these cases. Figure 3 illustrates the zero-derivation ratios of both techniques for each collection. While we can see that the HMM-based technique is, overall, more accurately resolving zero-derivation ambiguity than the rule-based technique, we can also see how severely zero derivation impacts performance. For the SAP collection both techniques were only able to resolve about 50% of all zero-derivation cases. The results for the AC collection are better (63% and 67%), but still not satisfactory. Only the results for the TC collection are highly accurate (89% and 92%). The reason for the varying zero-derivation resolution ratios can again be explained by the labeling style distribution. The more action-noun labels are used, the harder the accurate detection, because a single word such as "*order*" is likely to be used as both verb and noun in the same collection. Despite this remaining potential for improvement, the overall recognition results can be considered highly satisfactory.

5.4. Results for Semantic Component Identification

The results for the identification of the semantic components are provided in Table 6. The numbers show the overall accuracy of the techniques for correctly recognizing the semantic components. Since the correct recognition of the label style is a necessary precondition for the correct identification of the semantic components, the numbers are equal or lower than the recognition recall values.

The results illustrate that the identification accuracy is typically very close to the recognition

Table 6: Accuracy of semantic component identification

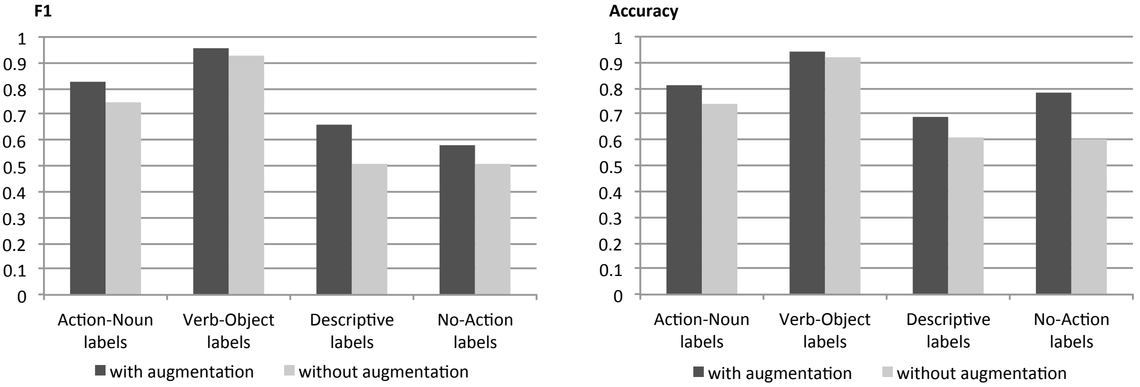
Style	**HMM-based technique**	**Rule-based technique**
Action-Noun labels	0.81	0.78
Verb-Object labels	0.94	0.95
Descriptive labels	0.69	0.23
No-Action labels	0.78	0.00
Total	**0.91**	**0.89**

recall of the respective style. This is so since the labeling style often reliably indicates the position of the different components. However, the accuracy of the HMM-based technique is about 2 percentage points higher than the accuracy of the rule-based technique. Comparing this to the advances typically presented in classical part-of-speech tagging, this needs to be considered as a substantial improvement [28]. The strengths of the HMM-based technique can be mainly observed for labels suffering from the *adjective-noun ambiguity.* As an example, consider the label "*Manual verification*". While the rule-based technique identifies "*manual*" as a business object, the HMM-based technique correctly identifies it as an action modifier. The most challenging, remaining error relates to labels with compound nouns, such as *bill of exchange* or *scope of work.* If the HMM has not seen these compounds in the training phase, it classifies the part of the preposition as additional fragment. The rule-based technique, however, is even more sensitive since it generally considers a preposition as start of the additional fragment.

5.5. Effect of Augmentation

Sparse training data is a considerable problem for any machine learning-based technique. However, process models represent a particularly sensitive data source. Hence, the augmentation of our HMM with additional non-process related resources is an important conceptual component of our technique. Figure 4 shows the effect of this augmentation on both recognition as well as identification.

Figure 4 illustrates that the augmentation significantly contributes to the overall recognition and identification performance. The numbers for the different labeling styles also show that this improvement cannot be traced back to a single labeling style. In fact, all labeling styles benefit from the augmentation. The biggest change can be observed for descriptive labels. This can be



(a) Recognition

(b) Identification

Figure 4: Effect of augmentation

explained by the rather limited occurrence of third-person verb forms in process models. Thus, the augmentation with the respective verb forms leads to a notable improvement. The smallest change can be observed for verb-object labels. It should be noted, however, that this represents a relative change. Taking the labeling style distribution of the test data set into account, a delta of three percentage points in verb-object style recognition means that due to the augmentation several hundred activity labels are correctly classified. In summary, the augmentation proves to be essential for the performance of our HMM.

5.6. Domain Dependence

Finally, given that the terminology used in process may differ considerably across domains, it is important to assess how well the proposed technique is able to deal activity labels from new domains. In other words, how domain dependent is the technique? To evaluate this, we performed evaluation experiments where we trained the HMM-based technique on two of the data sets and, subsequently, tested its performance on the third set. In this way, we can, for instance, test how well the technique performs on data from the telecommunication domain (TC collection), even though the technique is not trained on data from this domain.

Table 7 depicts the results obtained in this manner. The left-hand results (*Trained on other sets*), presents the results when the HMM technique is trained on the two other data sets, e.g., the first row indicates the performance of the technique for the SAP collection when the technique is

Table 7: Evaluation results for unseen domains

		Trained on other sets			**Trained on all sets**		
Test set	**Size**	**Prec**	**Rec**	**F1**	**Prec**	**Rec**	**F1**
SAP	2,433	0.64	0.87	0.74	0.92	0.89	0.90
TC	8,152	0.88	0.88	0.88	0.97	0.97	0.97
AC	4,903	0.85	0.85	0.85	0.93	0.90	0.92
Total	**15,488**	**0.83**	**0.87**	**0.85**	**0.95**	**0.94**	**0.94**

trained using the TC and AC collections. The right-hand of the table shows the original results obtained using k-fold cross validation over all data sets, as originally presented in detail in Table 5.

The table shows that the technique still performs rather well on activity labels from unseen domains, achieving an $F1$-score of 0.87. Still, it can also be observed that this performance is lower than the results obtained when training the technique on data from all domains ($F1$-score of 0.94). These results reveal that the technique's performance indeed reduces when applied to a previously unseen domain. However, it should be recognized that two other factors play a role here as well. First, the amount of training data is simply lower. Most notably, when applying the technique on the TC collection, we can only train the technique on 7.3k labels, whereas in the situation of 10-fold cross validation the technique is trained on 13.9k labels. Second, the distribution of the labeling styles is non-uniform across the three data sets. Primarily, the SAP collection consists of mostly AN-labels, which are comparably rare in the other two data sets. This means that for this evaluation, the HMM is trained on a data set with few training examples of this labeling style, even though it is by far the most commonly occurring style in the test set. As a result, in particular the precision achieved for this collection is considerably lower (0.64) than for the original precision (0.92).

Given these insights, it is important to recognize that, while indeed the domain-specificity of activity labels may always play a role, the latter two causes (i.e., limited training size and limited AN training data) are only present for this particular test set-up and would not be present if our technique is applied on additional data sets.

6. Limitations

Despite the satisfactory results presented in this paper, our technique is subject to a number of limitations. More specifically, there are limitations with respect to our evaluation experiments, the training of our technique, and the adaptability.

The limitations relating to our *evaluation experiments* mainly concern the test collection we used. It should be noted that it is not representative in a statistical sense. Therefore, we cannot generalize our results to any process model collection we may encounter in practice. However, we tried to select highly heterogeneous collections to cover as much diversity as possible. The evaluation experiments demonstrated that our technique can deal well this heterogeneity. It delivers a high and stable performance across all data sets. In the unlikely case of a collection exhibiting fundamentally different characteristics, our technique would still be valuable. It simply needs to be trained on such a deviating data set respectively.

The limitations with respect to the *training* of our technique are twofold. First, it might be challenging to obtain a sufficient amount of training data for our technique. As pointed out earlier, training data for process model analysis is sensitive and, therefore, not always easy to access. At the same time, our technique requires sufficient training data to perform well. Second, even when sufficient raw data is available, a training data set first needs to be manually created by respectively tagging the activity labels from the respective raw data set. This requires a certain amount of effort and insight. To solve these problems, we provide a pretrained version of our technique, which is ready for use.[2] In this way, the technique is fully applicable without requiring users to manually create a data set or to train the technique themselves. In case users still wish to retrain the technique on a different data set, we consider this a feasible task. In comparison to the technique from [17], the manual tagging of a data set is more intuitive than defining and implementing rules. Next to the pretrained technique, we also provide the non-restricted parts of our training data set via the above-mentioned link. We are convinced this can help interested users to understand how the training data set needs to be prepared.

The last limitation relates to the *adaptability* of our technique to other model elements and

[2] A download is available under http://www.henrikleopold.com/downloads/

languages. While the focus of this paper is on activity label analysis, our technique can be also applied to other process model elements, such as events and gateways. Prior work has shown that event and gateway labels suffer from less ambiguity than activity labels and are, in general, easier to analyze [6]. Thus, there are no conceptual adaptations necessary. However, it is required is to retrain our technique on a respective data set with manually annotated events and gateways. In the same way, our technique could be adapted to other languages than English. While the particular patterns of activity labels have been found to differ, the elements are the same [6]. Therefore, it is again sufficient to retrain our technique on a respective data set. While this is associated with a certain effort, it does not limit the general adaptability of our technique.

7. Related Work

The work presented in this paper relates to two major streams of research: the automated quality assurance of process models and the application of Hidden Markov Models.

Our work primarily relates to approaches for the *automated quality assurance* of process models. There are techniques available for checking formal properties of process models such as soundness [29, 30], the correctness of the dataflow [31, 32] and the satisfiability of constraints on the resource perspective [33, 34, 35]. As for linguistic aspects of process models, there are various approaches for detecting and enforcing naming conventions [6, 36, 37]. Some approaches can also check the linguistic consistency of the terms that are used for labeling model elements [38, 39, 40, 41, 42]. Many of these linguistic techniques rely on accurately recognizing the grammatical patterns of the activity labels in the first place. While the technique presented in [43] is able to achieve that, it is a rule based approach. This comes with a number of limitations, particularly with respect to performance stability. The solution presented in this paper overcomes these limitations by conceptualizing activity label analysis as a tagging task based on Hidden Markov Models.

Hidden Markov Models are probabilistic models that are used to annotate sequences of *observations* with *hidden states.* The contexts in which HMMs can be applied are diverse, including bioinformatics [44, 45], electrical engineering [46], and natural language processing. In the latter domain, which also encompasses our application context, a variety of use cases are addressed using

HMMs, such as speech [47] and handwriting recognition [48], part of speech tagging [49], machine translation [50], and information extraction [22, 51]. The strength of HMMs in the context of these applications, as well as in the context of the technique presented in this paper, is that they combine emission and transition probabilities. In this way, HMMs are able to reliably recognize patterns in sequences, including those involving previously unseen terms. Our use case distinguishes itself from the aforementioned, generic use cases, such as part of speech tagging and machine translation, primarily because activity labels are short text snippets, which are often not semantically complete and can be highly ambiguous.

8. Conclusion

In this paper, we addressed the problem of automatically analyzing the natural language of process model activity labels. Recognizing the limitations of existing techniques, we proposed a machine learning-based technique that builds on a Hidden Markov Model. Our approach conceptualizes activity label analysis as a tagging task and is able to recognize the labeling style and the semantic components of an activity label. Evaluation experiments with more than 15,000 activity labels demonstrated that our technique is highly accurate and outperforms the existing rule-based technique for activity label analysis [17]. What is more, our technique has shown to deliver a more stable performance and appears to be better suited for recognizing labels that follow less frequent labeling styles. In contrast to the existing rule-based technique, it is also able to detect no-action labels.

While this paper focused on activity label analysis, it is important to note that our technique can be also applied to other process model elements, such events and gateways, and even other languages. Prior work has shown that the grammatical elements and, therefore the tag set provided in this paper, are applicable among different process model elements and languages [6]. Hence, the adaptation of our technique does not require any conceptual adaptations. Our technique simply needs to be retrained on a respective data set.

In future work, we aim to further improve our technique. In particular, we strive for developing a solution for the complex zero-derivation cases. To this end, we plan to look into more sophisticated learning mechanisms and to exploit other external knowledge sources such as Wikis.

Bibliography

[1] W. M. Van Der Aalst, Workflow verification: Finding control-flow errors using petri-net-based techniques, in: Business Process Management, Springer, 2000, pp. 161–183.

[2] I. Weber, J. Hoffmann, J. Mendling, Beyond soundness: on the verification of semantic business process models, Distributed and Parallel Databases 27 (3) (2010) 271–343.

[3] N. Sidorova, C. Stahl, N. Trčka, Soundness verification for conceptual workflow nets with data: Early detection of errors with the most precision possible, Information Systems 36 (7) (2011) 1026–1043.

[4] H. Leopold, J. Mendling, O. Gunther, Learning from quality issues of bpmn models from industry, IEEE Software 33 (4) (2015) 26–33.

[5] B. Weber, M. Reichert, J. Mendling, H. A. Reijers, Refactoring large process model repositories, Computers in Industry 62 (5) (2011) 467–486.

[6] H. Leopold, R.-H. Eid-Sabbagh, J. Mendling, L. G. Azevedo, F. A. Baião, Detection of naming convention violations in process models for different languages, Decision Support Systems 56 (2013) 310–325.

[7] D. Paulraj, S. Swamynathan, M. Madhaiyan, Process model-based atomic service discovery and composition of composite semantic web services using web ontology language for services (owl-s), Enterprise Information Systems 6 (4) (2012) 445–471.

[8] R. Yousef, M. Odeh, D. Coward, A. Sharieh, Bpaontosoa: A generic framework to derive software service oriented models from business process architectures, in: Applications of Digital Information and Web Technologies, 2009. ICADIWT'09. Second International Conference on the, IEEE, 2009, pp. 50–55.

[9] H. Leopold, F. Pittke, J. Mendling, Automatic service derivation from business process model repositories via semantic technology, Journal of Systems and Software 108 (2015) 134–147.

[10] H. Leopold, M. Niepert, M. Weidlich, J. Mendling, R. Dijkman, H. Stuckenschmidt, Probabilistic optimization of semantic process model matching, in: International Conference on Business Process Management, Springer, 2012, pp. 319–334.

[11] M. Weidlich, R. Dijkman, J. Mendling, The ICoP framework: Identification of correspondences between process models, in: Advanced Information Systems Engineering, Springer, 2010, pp. 483–498.

[12] H. Leopold, Natural language in business process models, Springer, 2013.

[13] J. Mendling, H. A. Reijers, J. Recker, Activity Labeling in Process Modeling: Empirical Insights and Recommendations, Information Systems 35 (4) (2010) 467–482.

[14] K. Klose, R. Knackstedt, D. Beverungen, Identification of Services - A Stakeholder-Based Approach to SOA Development and its Application in the Area of Production Planning, University of St. Gallen, 2007.

[15] L. Makni, N. Z. Haddar, H. Ben-Abdallah, Business process model matching: An approach based on semantics

and structure, in: e-Business and Telecommunications (ICETE), 2015 12th International Joint Conference on, Vol. 2, SCITEPRESS, 2015, pp. 64–71.

[16] M. Weidlich, T. Sagi, H. Leopold, A. Gal, J. Mendling, Predicting the quality of process model matching, in: Business Process Management, Springer, 2013, pp. 203–210.

[17] H. Leopold, S. Smirnov, J. Mendling, On the refactoring of activity labels in business process models, Information Systems 37 (5) (2012) 443–459.

[18] K. Toutanova, C. D. Manning, Enriching the knowledge sources used in a maximum entropy part-of-speech tagger, in: Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, 200, pp. 63–70.

[19] K. Toutanova, D. Klein, C. Manning, Y. Singer, Feature-rich part-of-speech tagging with a cyclic dependency network, in: HLT-NAACL, 2003, pp. 252–259.

[20] J. Mendling, H. A. Reijers, J. Recker, Activity labeling in process modeling: Empirical insights and recommendations, Information Systems 35 (4) (2010) 467–482.

[21] L. R. Rabiner, A tutorial on hidden markov models and selected applications in speech recognition, Proceedings of the IEEE 77 (2) (1989) 257–286.

[22] Y.-Y. Wang, L. Deng, A. Acero, Spoken language understanding, Signal Processing Magazine, IEEE 22 (5) (2005) 16–31.

[23] D. Jurafsky, J. H. Martin, Speech & language processing, Pearson Education India, 2000.

[24] D. Klein, C. D. Manning, Accurate Unlexicalized Parsing, 41st Meeting of the Association for Computational Linguistics (2003) 423–430.

[25] G. Keller, T. Teufel, SAP(R) R/3 Process Oriented Implementation: Iterative Process Prototyping, Addison-Wesley, 1998.

[26] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Ijcai, Vol. 14, 1995, pp. 1137–1145.

[27] H. Leopold, S. Smirnov, J. Mendling, Refactoring of process model activity labels, in: Natural Language Processing and Information Systems, Springer, 2010, pp. 268–276.

[28] C. D. Manning, Part-of-speech tagging from 97% to 100%: is it time for some linguistics?, in: International Conference on Intelligent Text Processing and Computational Linguistics, Springer, 2011, pp. 171–189.

[29] W. M. P. van der Aalst, A. Hirnschall, H. Verbeek, An Alternative Way to Analyze Workflow Graphs, in: Proceedings of the 14th international conference on Advanced Information Systems Engineering, Vol. 2348 of LNCS, Springer, 2002, pp. 535–552.

[30] D. Fahland, C. Favre, J. Koehler, N. Lohmann, H. Völzer, K. Wolf, Analysis on Demand: Instantaneous Soundness Checking of Industrial Business Process Models, Data & Knowledge Engineering 70 (5) (2011) 448–466.

[31] S. Sun, J. Zhao, J. Nunamaker, O. Liu Sheng, Formulating the Data-Flow Perspective for Business Process

Management, Information Systems Research 17 (4) (2006) 374–391.

[32] I. Weber, J. Hoffmann, J. Mendling, Beyond Soundness: on the Verification of Semantic Business Process Models, Distributed and Parallel Databases 27 (3) (2010) 271–343.

[33] E. Bertino, E. Ferrari, V. Atluri, The Specification and Enforcement of Authorization Constraints in Workflow Management Systems, ACM Transactions on Information and System Security 2 (1) (1999) 65–104.

[34] M. Strembeck, J. Mendling, Modeling Process-related RBAC Models with Extended UML Activity Models, Information and Software Technology 53 (5) (2011) 456–483.

[35] J. Crampton, H. Khambhammettu, Delegation and Satisfiability in Workflow Systems, in: SACMAT 2008, ACM, New York, NY, USA, 2008, pp. 31–40.

[36] J. Becker, P. Delfmann, S. Herwig, L. Lis, A. Stein, Towards Increased Comparability of Conceptual Models - Enforcing Naming Conventions through Domain Thesauri and Linguistic Grammars, in: ECIS 2009, 2009.

[37] J. Becker, P. Delfmann, S. Herwig, L. Lis, A. Stein, Formalizing linguistic conventions for conceptual models, in: Conceptual Modeling - ER 2009, LNCS, Springer Berlin Heidelberg, 2009, pp. 70–83.

[38] A. Koschmider, T. Hornung, A. Oberweis, Recommendation-based editor for business process modeling, Data & Knowledge Engineering 70 (6) (2011) 483 – 503.

[39] A. Koschmider, E. Blanchard, User assistance for business process model decomposition, in: Proceedings of the 1st IEEE International Conference on Research Challenges in Information Science, 2007, pp. 445–454.

[40] C. Francescomarino, P. Tonella, Supporting ontology-based semantic annotation of business processes with automated suggestions, in: Enterprise, Business-Process and Information Systems Modeling, Vol. 29 of LNBIP, Springer Berlin Heidelberg, 2009, pp. 211–223.

[41] B. van der Vos, J. A. Gulla, R. van de Riet, Verification of conceptual models based on linguistic knowledge, Data & Knowledge Engineering 21 (2) (1997) 147 – 163.

[42] F. Pittke, H. Leopold, J. Mendling, Automatic detection and resolution of lexical ambiguity in process models, IEEE Transactions on Software Engineering 41 (6) (2015) 526–544.

[43] H. Leopold, S. Smirnov, J. Mendling, On the refactoring of activity labels in business process models, Information Systems 37 (5) (2012) 443–459.

[44] A. Krogh, B. Larsson, G. Von Heijne, E. L. Sonnhammer, Predicting transmembrane protein topology with a hidden markov model: application to complete genomes, Journal of molecular biology 305 (3) (2001) 567–580.

[45] M. Stanke, S. Waack, Gene prediction with a hidden markov model and a new intron submodel, Bioinformatics 19 (suppl_2) (2003) ii215–ii225.

[46] L. Satish, B. Gururaj, Use of hidden markov models for partial discharge pattern classification, IEEE transactions on electrical insulation 28 (2) (1993) 172–182.

[47] L. Bahl, P. Brown, P. De Souza, R. Mercer, Maximum mutual information estimation of hidden markov model parameters for speech recognition, in: Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'86., Vol. 11, IEEE, 1986, pp. 49–52.

[48] A. Kundu, Y. He, P. Bahl, Recognition of handwritten word: first and second order hidden markov model based approach, Pattern recognition 22 (3) (1989) 283–297.

[49] C. D. Manning, P. Raghavan, H. Schütze, Introduction to information retrieval, Vol. 1, Cambridge university press Cambridge, 2008.

[50] S. Vogel, H. Ney, C. Tillmann, Hmm-based word alignment in statistical translation, in: Proceedings of the 16th conference on Computational linguistics-Volume 2, Association for Computational Linguistics, 1996, pp. 836–841.

[51] H. van der Aa, H. Leopold, A. del Río-Ortega, M. Resinas, H. A. Reijers, Transforming unstructured natural language descriptions into measurable process performance indicators using hidden markov models, Information Systems 71 (2017) 27–39.