Spotting terminology deficiencies in process model repositories

Fabian Pittke[1,3], Henrik Leopold[1], and Jan Mendling[2]

[1] Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany
henrik.leopold@wiwi.hu-berlin.de
[2] WU Vienna, Augasse 2-6, A-1090 Vienna, Austria
jan.mendling@wu.ac.at
[3] SRH University Berlin, Ernst-Reuter-Platz 10, 10587 Berlin, Germany
fabian.pittke@srh-uni-berlin.de

Abstract. Thinking in business processes and using process models for their documentation has become common practice in companies. In many cases this documentation encompasses more than thousands of models. One of the key challenges is the realization of consistency of the used terminology. Especially, the usage of synonym and homonym words is one of the most severe problems for terminological consistency. Therefore, this paper presents an automatic approach for identifying synonym and homonym words in model repositories. We challenged the approach against three model collection from practice that are assumed to have different levels of terminological consistency. The evaluation shows that the approach is capable to fulfill these goals and to identify meaningful synonym and homonym candidates for the following resolution.

Key words: Identification of Synonyms, Identification of Homonyms, Business Process Models

1 Introduction

Business process models have become an integral of general documentation available in enterprises, often covering thousands of models. A key challenge for such large-scale modeling initiatives is to achieve consistency and comparability [1] of the models, which are created by different process analysts or by the various business practitioners themselves. Without appropriate measures, models are often inconsistent in terms of their layout, their level of detail, their labeling styles, or their terminology [1, 2, 3].

The issue of inconsistent terminology has been acknowledged in various areas of conceptual modeling. It relates to the semantic level of a model and the meaning associated with elements and names of elements [4]. The usage of synonyms and homonyms is one of the most tangible symptoms of inconsistent terminology [5, 6, 7], for example when both words *invoice* and *bill* are used. Proposals like creating a domain thesaurus [8] or technical term modeling [9] are well justified to fix such terminology issue before starting to model. However,

modelers might find it too restrictive to check terms while modeling and tools might not be able to enforce term usage. Also, such restrictions do not directly help in cleaning up an existing model collection.

Against this background, we approach the problem of synonyms and homonyms from a non-restrictive perspective. We assume that modelers have complete control on how to assign names to model elements, which is indeed inline with how tools generally support model creation. In such a setting, automatic analysis capabilities are required to inspect a model repository for potential terminological problems. The contribution of this research is a technique for the automatic detection of synonyms and homonyms in a model repository. This technique is meant to be used either by repository managers in an offline model or by process analysts in an online mode for spotting issues and providing recommendation for solving them. The capabilities of this technique are evaluated for three process model collections from practice.

This paper proceeds as follows. Section 2 illustrates the terminology problem and provides an overview of its theoretical background. Section 3 defines the concepts of our automatic detection technique. Section 4 presents the results of applying our technique for three process model collections from practice. Section 5 relates our contribution to other research in the area of conceptual modeling. Finally, Section 6 concludes the paper and gives an outlook on future research.

2 Problem Illustration

Different aspects of process model quality have been addressed in prior research. For instance, structural and behavioural problems can be automatically identified and resolved using verification techniques [10, 11, 12]. The natural language content of process models has only been considered to a limited extent. For instance, available refactoring techniques rework the grammatical structure of an activity label and extract the action and the business object [3]. However, process models in practice often exhibit weaknesses in words of terminology. The particular challenge of terminological deficiencies relates to the fact that one syntactical word can have multiple meanings (homonymy) and that the same meaning can be represented by the different syntactical words (synonymy). In linguistic literature this phenomenon of meaning is discussed in the field of *lexical semantics* [13].

The impact of word meaning ambiguity for process model quality is illustrated in Figure 1, showing two process models constructed by different modelers. Scenario A describes the typical job application process of a company, i.e. a person that is interested in a certain job position applies for it and sends his or her application to the company. Scenario B depicts a software development process. It starts with a pre-analysis of application requirements, before relevant requirements are identified and evaluated. Thereafter, the application is designed, implemented, tested and installed.

As highlighted in Figure 1 by the right shades, we observe that the word *application* is used in two different contexts leading to different meanings. In sce-

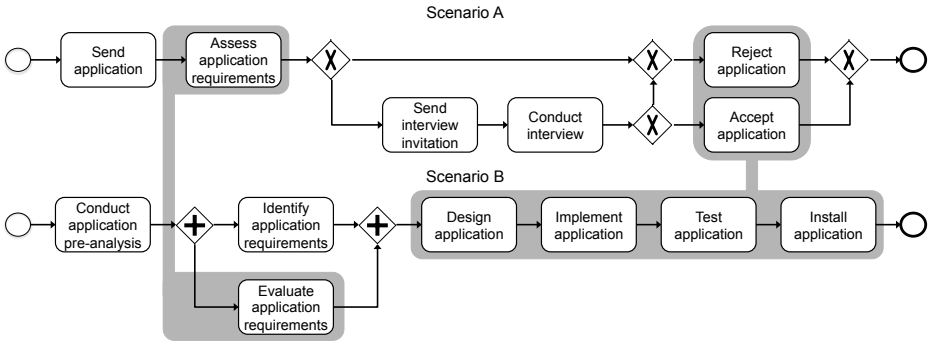

Fig. 1. Example of business process models with synonyms and homonyms

nario A, the word *application* clearly refers to a written request for employment, while scenario B uses the word in the context of a software program. Thus, from a linguistic point of view, the word *application* is a homonym, from which the according meaning has to be derived by the process modeler or stakeholder. The labels *Assess application requirements* from scenario A and *Evaluate application requirements* from scenario B further highlight the problem of homonyms. Both activities instruct employees to work on a business object represented by the syntactically identical word *application requirements*. Accordingly, process stakeholders cannot necessarily distinguish whether the requirements for a software application or requirements for a job position have to be evaluated. Again, the homonymous usage of the word application complicates the understanding of the process models and decreases the terminological clarity of these labels.

Highlighted by the left shades in Figure 1, the labels *Assess application requirements* and *Evaluate application requirements* both give the instruction to perform an evaluation task on the business object at hand. Furthermore, we notice that the actions *to assess* and *to evaluate* are used in the sense of *estimating the quality or significance* of the application requirements. This implies that the meaning of both actions is the same, stating that both actions are synonym words. In this case, one could think of replacing one word with such a word that is more comprehensible to process stakeholders. In consequence, the usage of synonymousalso impedes the understandability of process models.

To identify such problems in business process models, we face the challenge of automatically determining the meaning of words. As illustrated in the example above, that can be very challenging. The word *application requirements* is used in two completely different contexts and there is only little information available to correctly recognize this fact. Since process models only contain short linguistic fragments which in many cases do not even represent proper sentences, it is not possible to directly use standard disambiguation technology from linguistics [14, 15, 16]. Techniques for word meaning disambiguation usually employ statistical methods working on the syntactic structure of the underlying sentences. Hence, if this syntactic structure of sentences is missing or cannot be recognized, these techniques cannot be applied. In order to still identify synonyms and homonyms

in process models, alternative strategies have to be pursued. However, given these examples, it is apparent that the identification of synonyms and homonyms is an important step for improving the overall quality of a model collection.

3 Conceptual Approach

This section introduces the conceptual approach for the identification of synonyms and homonyms in process models. We first summarize preliminaries before illustrating the steps to identify synonyms or homonyms respectively.

3.1 Preliminaries

We start with the definition of the objects subject to this research, i.e. the activity labels of process models in general. According to [17], process model activities may follow different label styles, meaning that the same information can be conveyed in different ways. As an example consider the label *Notify customer* containing the action *to notify* as a verb and the business object *customer* as a noun. The same information could be also provided with the label *Customer notification*, where the action *to notify* is represented by the noun *notification*. However, as the technique defined in [3] is capable of aligning different labeling styles and automatically identifying action and business object of a label, we abstract from these problems in the remainder of the paper. As a result, we can directly work with the actions and business objects of the considered activities.

We start with the set of activities $\mathcal{A}$ derived from a model collection. For each activity $a \in \mathcal{A}$ we consider one action a_{action} and one business object a_{bo}. Due to simplicity, we abstract from multiple actions and business objects here. Further, we define $words_{action}$ as the set of actions of a process model collection, i.e. $words_{action} = \bigcup_{a \in \mathcal{A}} a_{action}$. The set $words_{bo}$ comprises all business objects, i.e $words_{bo} = \bigcup_{a \in \mathcal{A}} a_{bo}$. Finally, the union of both sets $\mathcal{W} = words_{action} \cup words_{bo}$ defines the set of all words of a given collection. In Figure 1, there are *application*, *application requirements* or *interview invitation* as $words_{bo}$. The actions *to send*, *to assess* or *to identify* are accordingly part of $words_{action}$.

Additionally, let $\mathcal{M}$ be the set of all meanings and $m_i \in \mathcal{M}(i = 1, ..., n)$ all meanings that one syntactical word $w \in \mathcal{W}$ is associated with. In order to capture the relation between meanings and a syntactical word, we define the relation *Word Meaning* $\mathcal{WM}$ that matches one word with its corresponding meanings:

$$\mathcal{WM} \subseteq \mathcal{W} \times \mathcal{M} \tag{1}$$

As an example, consider the word *to send*. According to the lexical database WordNet [18], that word has eight different meanings. One of this meanings is described as *cause to be directed or transmitted to another place*. For this particular meaning of the word *to send*, synonyms such as *to mail* or *to post* can be found. However, if *to send* is used in the sense of *sending something over the airwaves*, the words *to broadcast* or *to air* would be suitable synonyms.

This example makes clear that lexical relations such as synonymy must be identified based on the word meaning and cannot be simply defined by looking at a syntactical sequence of characters.

Accordingly, the synonym and the homonym relation is defined. A *synonym relation* is a symmetric relation between two words $w_1, w_2 \in \mathcal{W}$, where both words represent the same meaning $m \in \mathcal{M}$. Formally, the synonym relation can be defined as follows:

$$Syn = \{(w_1, w_2) | \exists_{m \in \mathcal{M}} : (w_1, m) \in \mathcal{WM} \wedge (w_2, m) \in \mathcal{WM}\} \tag{2}$$

The *homonym relation* is a relation of one words and its meanings, where this word is syntactically equal, but represents two different meanings $m_1, m_2 \in \mathcal{M}$. The relation can formally be described as follows:

$$Hom = \{w \in \mathcal{W} | \exists_{m_1, m_2 \in \mathcal{M}} : (w, m_1) \in \mathcal{WM} \wedge (w, m_2) \in \mathcal{WM}\} \tag{3}$$

Considering the process models in Figure 1 we note that both process models use the business object *application*. Yet, we observe that the business object differs in its meanings. In scenario A, the word *application* is used in the context of a written request for employment. In scenario B, *application* refers to a software application. Hence, the context could actually reveal that we face two different meanings m_1 and m_2 for a syntactically identical word. Therefore, definition 3 classifies the word *application* as a homonym.

Aiming for the automated identification of such phenomena, it is crucial to appropriately determine the meanings which can be associated with a given word. For identifying all possible meanings of a word, we can use the lexical databases such as WordNet [18]. WordNet organizes words in sets of synonyms, so called Synsets. Each Synset represents a meaning of a given word. One word can occur in multiple Synsets. Among others, Wordnet also defines direct lexical relations between words, such as synonymy or meronymy (*part of* relationship). Homonymy is not directly associated with a word, but can be assessed based on the number of Synsets associated with it.

In general, the meaning of a word depends upon the context of its use. In natural language texts, context is given by one or more sentences or a paragraph respectively. Hence, linguistic approaches often make use of a large context from which the meaning of a word can be derived. In process models, however, context information is very sparse. The symbols and the small pieces of texts are not sufficient for standard natural language techniques to derive the meaning of a given word. In order to be still able to select a likely meaning, we make use of the words co-occurring in the label. For an activity label such as *Process application*, this is the action *to process* and the business object *application*. Although this is limited information, the action *process* can already help to reduce the potential number of meanings which can be associated with *application*.

Accordingly, we introduce the notion of context in process models. First, we define the context for an activity a containing the word w. As previously defined,

the word w can either represent an action or a business object. The function *labelContext* returns the corresponding action, if w represents a business object and returns the business object if w is an action:

$$labelContext(w, a) = \begin{cases} a_{action} & \text{if } w \in \mathcal{W}_{\lfloor \wr} \\ a_{bo} & \text{if } w \in \mathcal{W}_{\dashv \rfloor \sqcup \rangle \wr \backslash} \end{cases} \tag{4}$$

Second, we define the context of a word w in a process model that consists of a specific set of activities $\mathcal{A}$. The function *modelContext* returns the set of words which contextualize the input word w for a set of activities $\mathcal{A}$ typically stemming from the whole repository:

$$modelContext(w, \mathcal{A}) = \{labelContext(w, a) | a \in \mathcal{A}\} \tag{5}$$

3.2 Identification of Synonyms

Considering the definition from Equation 2, the identification of synonym words can be reduced to the identification of word pairs that share a common meaning. However, the problem is that this procedure must be operationalized for process models. The pairwise comparison of words is not efficient for large process model collections as the consideration of all possible word combinations would result in a huge search space. Further, it is necessary to adequately determine the meaning of the considered words. Otherwise, if the context of a word is not considered, the result set would suffer from a low precision.

To clarify how we address these challenges in the proposed approach, consider the example models from Figure 1. To reduce the overall search space and guarantee that a considered word pair is semantically related, the approach only investigates word pairs with a common *context word*. That means that two actions are only considered as synonymy candidates if they are both applied to the same business object. In Figure 1 this applies to the actions in *Assess application requirements* and *Evaluate application requirements* as they both share the common business object *application requirements*. In order to verify the assumption of synonymy, WordNet can be used to check if these actions share a common meaning. In case of a common context and a common meaning in WordNet, two words are consequently considered as synonyms.

Algorithm 1 illustrates the required steps formally. The algorithm starts with the initialization of the result set. Afterwards, the algorithm determines the set of words *modelWords* from the collection given a context word w (lines 3–4). For each pair $w_1, w_2 \in modelWords$, the meaning is identified and stored in separate sets (lines 5–8). If at least one meanings is identified that is shared by both words, these words are stored in the result set (lines 9–10). The algorithm terminates with the return of the set of potential synonym candidates (line 11).

3.3 Identification of Homonyms

As outlined before, the identification of homonyms relates to the problem to find words in a process model collection having different meanings in diverging

Algorithm 1: Identification of Synonyms in a Process Repository

```
1: identifySynonyms(Activities 𝒜)
2: Set synonymCandidates = new List();
3: for all w ∈ 𝒲 do
4:    Set modelWords = getModelContext(w, 𝒜);
5:    for all w1 ∈ modelWords do
6:       for all w2 ∈ modelWords do
7:          Set s1 = wordnet.getSynsets(w1);
8:          Set s2 = wordnet.getSynsets(w2);
9:          if s1 ∩ s2 ≠ ∅ then
10:            synonymCandidates.add(w1,w2);
11: return synonymCandidates;
```

contexts. Following the formal definition of a homonym, that means that we have to identify words which have at least two different meanings. However, as many words have slightly varying meanings, this would be neither efficient nor effective. The challenge here is to identify words which do not only have different meanings, but are also used with different meanings.

In order to address this problem, we apply the function *modelContext* on a given word to obtain the context from all process models containing the target word. Using the SenseRelate approach [19] the context can be used to disambiguate the target word by finding the most fitting meanings. Yet, this does not properly identify homonyms, since one word can have different meanings that are semantically very close.

As example consider the word *insurance*. According to WordNet an insurance can be a promise of reimbursement in the case of loss, a written contract or certificate of insurance, or the protection against future loss. Obviously, these meanings are closely related with the result that *insurance* cannot be considered as potential homonym. Therefore, a refinement is needed that supports the distinction between more or less ambiguous homonyms. This is done by the *Semantic Homogeneity* indicator SH. It takes several information as an input, i.e. the target word w, the Synsets of the target word and a weight for each Synset. This weight is calculated from SenseRelate that determines the average similarity scores of the target word to the respective Synset given a context. We denote the weight of a Synset as w_i. The SH indicator is then calculated as the weighted sum of similarity scores from the target word to each word of the respective Synset, denoted as $sim(w, S_i)$ (see Equation 6). The indicator ranges from 0 to 1. A score closer to 0 indicates that the word is used with different meanings and therefore is likely considered as homonym, while higher scores up to 1 classify a word as non-homonym.

$$SH(w, \mathcal{S}) = \sum_{i=1}^{n} w_i \frac{sim(w, S_i)}{|S_i|} (\forall S_i \in \mathcal{S}) \tag{6}$$

Algorithm 2: Identification of Homonyms in a Process Repository

```
1: identifyHomonyms(Activities 𝒜)
2: Map homonyms = new Map();
3: for all w ∈ 𝒲 do
4:    List wordContext = getModelWords(w, 𝒜);
5:    Map weightedSynsets = SenseRelate.getWeightsForSynsets(w, wordContext);
6:    if weightedSynsets.size() > 1 then
7:       float SH = calculateSemanticHomogeneity(w,weightedSynsets);
8:       homonyms.add(w, SH, weightedSynsets);
9: return semanticCloseness;
```

Algorithm 2 formalizes the homonym identification approach. The algorithm starts with the initialization of the result map that stores the semantic closeness score as well as the set of most probable Synsets for each word (line 2). For each word, we extract all context words and calculate the weight for each Synset using SenseRelate. The result is stored in a map (lines 3–5). If it turns out that a word only has one Synset, we can exclude this word to be a homonym according to the definition. Otherwise, we calculate the semantic homogeneity (line 6–7). Finally, the semantic closeness is stored in the result map along with the word and the weighted Synsets(lines 8). The algorithm terminates by returning the set of potential homonym candidates (line 9).

3.4 Resolving Terminological Issues

The presented identification techniques provide sets of words that are used as synonyms or homonyms in the analyzed model collection. Both analysis results are considered as terminological issues that have to be resolved in the next step. Accordingly, repository managers can use the identified cases to manually resolve these issues. For synonym words the repository manager can replace a considered word with the more specific word. In case of our example, the repository manager might replace the action *to assess* with the action *to evaluate* since the latter one is more specific according to WordNet. This replacement can potentially been conducted automatically. For homonym resolution the repository manager might add additional words to disambiguate the identified homonyms. For our example, one might change the business object *application* to *job application* in scenario A and to *software application* in scenario B. As a result of applying these resolution techniques, the specificity and the terminological quality of the repository will be increased and the understandability can be improved.

4 Evaluation

This section shows the evaluation of the presented identification techniques with process model repositories from practice. Our evaluation includes three repositories from practice. Section 4.1 provides detailed information on each

Characteristic	SAP	TelCo	Academic Initiative
Models	604	286	597
Labels	2433	3155	4958
Unique actions	322	557	1200
Unique Business objects	885	1959	3132

Table 1. Details of evaluation sample

repository. Section 4.2 presents the results for the application of our techniques. We further provide examples of synonyms or homonyms that were identified.

4.1 Model Repository Demographics

In order to demonstrate the applicability of the presented techniques, we employ three different model collections from practice. We selected collections differing in the expected degree of terminological standardization. Since there is no gold standard available, we test whether our techniques are capable of identifying the assumed degree of homonymy and synonymy in the collections. Accordingly, we expect to find more synonyms and homonyms in non professionally maintained repositories containing models from a large number of modelers. Table 1 provides an overview of characteristics of each repository.

The SAP Reference Model contains 604 Event-Driven Process Chains organized in 29 different functional branches [20]. Examples are procurement, sales or financial accounting. The model collection includes 2433 activity labels with 322 unique actions and 885 unique business objects. Since the Reference Model was designed as a recommendation for the industry using a standard terminology, we expect a small number of homonyms and synonyms.

The process model collection from an international telecommunication company, which we will refer to as TelCo, comprises 286 separate process models with 3155 activities. We identified 557 distinct actions and 3132 business objects to test our approach on. We assume TelCo to be less strictly standardized as it uses terminology for telecommunication industry. However, there is no central glossary of terms available as in the case of the SAP models.

The model collection from the BPM Academic Initiative (AI)[1] comprises tens of thousands of process models. The models are formalized in various modeling languages and size. Our subset includes 597 process models, mostly in BPMN notation from a wide range of industrial and academic institutions. The collection subset encompasses 4958 activity labels in total, where 1200 actions and 3132 business objects are distinct. Since the collection targets no specific industry and is rather uncontrolled, the number of synonyms and homonyms is expected to be the highest among all repositories.

[1] http://bpmai.org

Characteristic	SAP	TelCo	Academic Initiative
Actions covered in WordNet	263 (81,7%)	283 (50,1%)	526 (43,8%)
Business Objects covered in WordNet	160 (18,1%)	226 (11,54%)	448 (14,3%)
Synonym actions	2	32	53
Synonym Business objects	2	12	102
Synonyms total	4	44	155

Table 2. Results of Synonym Identification

4.2 Evaluation Results

This section presents the identification results for all three model repositories. According to the assumptions for each model repository, we check whether the identification technique is capable to identify the overall tendency of the respective collection. Thus, we show the absolute numbers of identified synonyms and homonyms for collections, before providing a list of commonly identified synonyms or homonyms for each collection.

Table 2 depicts the number of identified synonyms in the model collections. Since we rely on WordNet, we can only identify words that are part of the database. For SAP, 263 actions and 160 business objects are covered. Yet, most of the business objects, such as *transfer time sheet*, *customer scheduling agreement* or *product structure management* are not included due to their specificity. Applying the synonym identification approach we identify 4 cases in which actions or business objects are used as synonyms. We conclude that the SAP collection has a fixed set of terms that are used to describe business processes, where semantically identical functions are also labeled equally in a clear and distinctive way. This clearly corresponds to the assumption we stated in the last section.

For the TelCo collection, WordNet covers 283 action and 226 business objects Thus, the approach identified 32 distinct cases of interchangeable actions and 12 cases of business objects. Compared to SAP, we face a higher number of synonyms in total. We can assume that this industry uses a fixed set of terms that is also reflected in their process models. Additionally, the low number reflects the quality and the maturity of process models and the repository managers. All in all, the results correspond to the assumption from Section 4.1.

The AI collection contained 526 actions and 448 business objects that could be analyzed, resulting in the identification of 53 synonym actions and 102 synonymous business objects. Compared to SAP, we observe that these business processes are not modeled distinctively. Since many models stem from academic training and from different application areas, the degree of ambiguity is considerably higher. This is reflected by the high number of synonym word pairs. Again, the assumption for AI is confirmed.

Figure 2 illustrates the quantitative results for homonym detection. Using the indicator of *Semantic Homogeneity* we show the numbers of homonym candidates for different ranges. As outlined before, lower SH values suggest a homonym , whereas higher values indicate an unambiguous word.

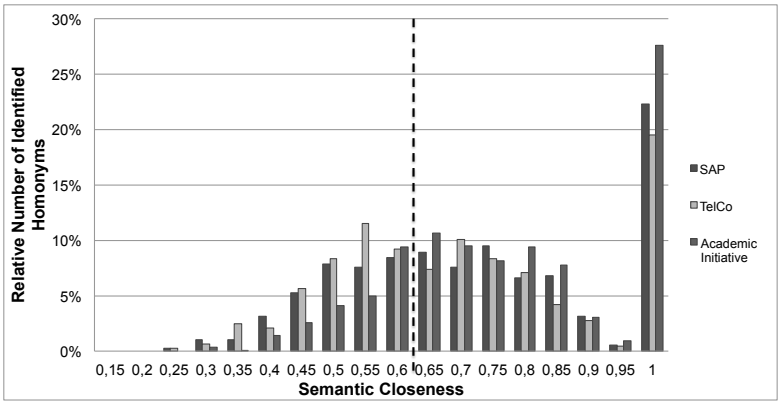

Fig. 2. Results of Homonym Identification

For the SAP collection, the techniques analyses 381 of 423 words in total. The loss of 42 words results from labels consisting of an action without any business object, i.e. having no context. This is similar for the other two collections (TelCo: 477 of 509; AI: 924 of 974). We observe that the number of homonyms is steadily increasing until 0.6. Then, the number remains constant and decreases for higher than 0.8. We also observe about 23% of words having a semantic closeness of 1.0 that denotes unambiguity. We observe a similar trend for the TelCo collection. Equal to SAP, we observe nearly 20% of words that are distinctive. For the AI collection, we observe an increase until 0.65 and a immediate decrease until 0.95. For an SH score of 1 the relative number is significantly higher for all three collections. By far, the AI collection outperforms the other (27%), followed by SAP (23%) and TelCo (19%).

Having a closer look at the relative numbers of SAP and TelCo, we observe that TelCo exceeds SAP in the range from 0.2 to 0.6. For higher ranges, we observe the opposite. As a result, we conclude that SAP has a higher terminological quality for process models than TelCo which clearly corresponds to the assumption in Section 4.1. Yet, for AI, we observe a diverging behavior. AI appears to be far more precise compared to SAP. The average number of ambiguous words is smaller below 0.6, whereas the average number of unambiguous words is higher above 0.6 and especially for a score of 1. Thus, the assumption stating that the AI is more ambiguous than SAP has to be rejected for homonyms. An explanation for this phenomenon might be that most of the unambiguous words, such as *transfer time sheet* or *customer scheduling agreement*, are not part of WordNet. Yet, we observe that for instance the word *transfer time sheet* is a meronym of the word *time sheet* that already is unambiguous. In consequence, we could conclude that the whole word is also unambiguous.

To conclude our evaluation, we provide qualitative results of our identification techniques. First, we give an extract from the prominent examples of synonyms spread among all model collections. The results are depicted in Table 3. We frequently identified pairs associated with the action *to send*. Interchangeable

Rank	Synonym Actions	Frequency	Synonym Business Objects	Frequency
1	(place, send)	35	(client, customer)	14
2	(send, ship)	20	(bill, invoice)	12
3	(issue, release)	8	(data, information)	4
4	(send, transmit)	6	(inventory, stock)	3
5	(post, send)	6	(case, event)	2

Table 3. List of most frequent Synonyms

words are *to ship* or *to transport*, if physical goods have to be delivered somewhere. In case messages have to be send, one might prefer actions such as *to transmit* or *to post*. The most prominent examples for synonym business objects are the pairs *client* and *customer* as well as *bill* and *invoice*.

Second, we turn to homonym actions and business objects as shown in Tables 4 and 5. The tables show the homonym words along with their SH score and their frequency in the samples. We also added an extract of four possible meanings at the utmost along with the weight.

By far, the action *to expire* has the lowest SH score. Although it appears only four times in the sample and although it only has three different meanings, the meaning *to pass form life* is dominating in the context. The proper meaning in business context is on the last position. In contrast, the action *to specify* has 7 different meanings and a similar SH score. Yet, the meaning we would consider in a business context only comes on the fourth position. We also listed the action *to time* as example with SH score equal to 1. Although this action has 5 different meanings, we observe that the different meanings are homogeneous with respect to assign a fixed time for an event in the future.

For the business objects, the approach identified *go* as highly ambiguous ($SH = 0.38$) which is also reflected by the four different meanings of this word. Interestingly, the meaning with the highest weight is from the area of board games followed by a designer name for MDMA. It also seems that none of the meanings fits into a business context. Another interesting example is the business object *issue*. The word has 11 different meanings and appears that meaning 2 and 3 are prominent in business process context. It underlines the ambiguity of the word and the need for a unambiguous word such as supply or topic/subject as an alternative. The tables finishes with the word *communication* as a fine example for an unambiguous word with three different meanings.

5 Related Work

The research relates to three major streams of research: approaches for automatically assuring process model quality, applications of natural language processing techniques in process models, and the field of lexical disambiguation.

Automatic quality assurance for process models has been intensively studied for structural properties. For example, soundness as a structural property of the state space can be efficiently checked using Petri-Net concepts [21, 22].

Word	SH	Frequency	Possible Meaning	Weight
expire	0.31	4	pass from physical life	0.66
			expel air	0.19
			lose validity	0.14
specify	0.33	7	be specific about	0.26
			determine the essential quality	0.25
			select something for a specific purpose	0.13
			specify as a condition or in an agreement	0.12
...	...	...	...	...
time	1	2	measure the time of an event	0.28
			adjust so that a force is applied and an action occurs at the desired time	0.23
			regulate or set the time of	0.16
			assign a time for an activity or event	0.16

Table 4. List of Homonym Actions

Word	SH	Frequency	Possible Meaning	Weight
go	0,38	6	a board game for two players who place counters on a grid	0,36
			street names for methylenedioxymethamphetamine	0,25
			a usually brief attempt	0,21
			a time for working (after which you will be relieved by someone)	0,17
issue	0,39	6	a phenomenon that follows some previous phenomenon	0,18
			act of providing an item	0,12
			some situation or event that is thought about	0,11
			the immediate descendants of a person	0,10
...	...	...	...	...
communication	1	2	the activity of communicating	0,43
			something that is communicated by people	0,41
			a connection allowing access between persons	0,14

Table 5. List of Homonym Business Objects

The degree of structuredness can be improved by using automatic refactoring techniques [23, 24]. Also the content of activity labels can be automatically refactored based on label style parsing techniques which are based on natural language processing [3, 25]. Our technique complements these approaches with an automatic technique for detecting terminological issues.

Our technique also relates to the application of natural language techniques in conceptual modeling more generally. Examples are automatic service identification [26, 27], the identification of semantically equivalent activities in different

models [28, 29], and the discovery of process patterns [30]. Our technique might serve as a preprocessing before application these approaches.

The problem of disambiguation is one of the key problems of computational linguistics, with the approach by Sanderson being one of the most prominent ones [14]. However, short language fragments remains a current challenge [15, 16]. Although we do not provide a general solution for this linguistic challenge, we operationalized and successfully address the issue in the context of process models.

6 Conclusion

We presented a novel approach for the automatic identification of synonym and homonym terms in process model repositories. The approach exploits the meaning of terms based on the process model context as well as language processing techniques to identify homonyms and synonyms. Our techniques have been implemented prototypically and evaluated with more than 1400 process models from three different repositories from practice. The evaluation demonstrates its capability to spot and quantify terminological issues in these repositories.

In future research, we first plan to improve the approach by extending the search space of words and overcoming the limitations of WordNet. The goal is to cover more actions and business objects and identify more terminological issues. Second, we aim to extend the technique to resolve the identified synonyms and homonyms. We identified the context of a term especially in process models as a critical component and thus want to integrate additional business knowledge incorporated in specific domain ontologies or text corpora. We consider the integration of such knowledge-intensive technologies as a promising step to support the identification as well as the resolution of synonyms and homonyms.

References

1. Becker, J., Rosemann, M., Uthmann, C.: Guidelines of Business Process Modeling. In: BPM 2000. Volume 1806 of LNCS. Springer (2000) 30–49
2. Davis, R.: ARIS design platform: advanced process modelling and administration. Springer (2008)
3. Leopold, H., Smirnov, S., Mendling, J.: On the refactoring of activity labels in business process models. Information Systems **37**(5) (2012) 443 – 459
4. Thalheim, B.: Syntax, semantics and pragmatics of conceptual modelling. In: NLDB 2012. Volume 7337 of LNCS., Springer (2012) 1–10
5. Dean, D., Lee, J., Orwig, R., Vogel, D.: Technological support for group process modeling. Journal of Management Information Systems (1994) 43–63
6. Rosemann, M., Muehlen, M.: Evaluation of workflow management systems-a meta model approach. Australian Journal of Information Systems **6** (1998) 103–116
7. Rolland, C.: L'e-lyee: coupling l'ecritoire and lyeeall. Information & Software Technology **44**(3) (2002) 185–194
8. Becker, J., Delfmann, P., Herwig, S., Lis, L., Stein, A.: Formalizing linguistic conventions for conceptual models. In: Conceptual Modeling - ER 2009. Volume 5829 of LNCS., Springer (2009) 70–83

9. Becker, J., Kugeler, M., Rosemann, M.: Process management: a guide for the design of business processes. Springer Verlag (2003)
10. Fahland, D., Favre, C., Jobstmann, B., Koehler, J., Lohmann, N., Völzer, H., Wolf, K.: Instantaneous soundness checking of industrial business process models. In: BPM 2009. Volume 5701 of LNCS., Springer (2009) 278–293
11. van der Aalst, W., de Beer, H., van Dongen, B.: Process mining and verification of properties: An approach based on temporal logic. In: OTM Conferences (1). Volume 3760 of LNCS. Springer (2005) 130–147
12. van der Aalst, W.M.P., de Medeiros, A.K.A.: Process mining and security: Detecting anomalous process executions and checking process conformance. Electron. Notes Theor. Comput. Sci. **121** (February 2005) 3–21
13. Cruse, A.: Meaning in Language: An Introduction to Semantics and Pragmatics. Oxford University Press (2004)
14. Sanderson, M.: Word sense disambiguation and information retrieval. In: ACM SIGIR 1994. (1994) 142–151
15. Stokoe, C., et al.: Word sense disambiguation in information retrieval revisited. In: ACM SIGIR. (2003) 159–166
16. Resnik, P.: Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. JAIR **11** (1999) 95–130
17. Mendling, J., Reijers, H.A., Recker, J.: Activity Labeling in Process Modeling: Empirical Insights and Recommendations. Inf. Sys. **35**(4) (2010) 467–482
18. Miller, G.: WordNet: a Lexical Database for English. CACM **38**(11) (1995) 39–41
19. Patwardhan, S., Banerjee, S., Pedersen, T.: Senserelate::targetword: a generalized framework for word sense disambiguation. In: ACL 2005. (2005) 73–76
20. Keller, G., Teufel, T.: SAP(R) R/3 Process Oriented Implementation: Iterative Process Prototyping. Addison-Wesley (1998)
21. Fahland, D., Favre, C., Koehler, J., Lohmann, N., Völzer, H., Wolf, K.: Analysis on Demand: Instantaneous Soundness Checking of Industrial Business Process Models. Data & Knowledge Engineering **70**(5) (2011) 448–466
22. van der Aalst, W.M.P., Hirnschall, A., Verbeek, H.M.W.: An alternative way to analyze workflow graphs. In: CAiSE 2002. LNCS 2348 (2002) 535–552
23. Weber, B., Reichert, M., Mendling, J., Reijers, H.A.: Refactoring large process model repositories. Computers in Industry **62**(5) (2011) 467–486
24. Polyvyanyy, A., García-Bañuelos, L., Dumas, M.: Structuring Acyclic Process Models. In: BPM 2010. Volume 6336 of LNCS. Springer (2010) 276–293
25. Becker, J., Delfmann, P., Herwig, S., Lis, L., Stein, A.: Towards Increased Comparability of Conceptual Models - Enforcing Naming Conventions through Domain Thesauri and Linguistic Grammars. In: ECIS 2009. (2009) 2231–2242
26. Leopold, H., Mendling, J.: Automatic derivation of service candidates from business process model repositories. In: BIS 2012. LNBIP 117 (2012) 84–95
27. Knackstedt, R., Kuropka, D., Müller, O.: An ontology-based service discovery approach for the provisioning of product-service bundles. In: ECIS 2008. (2008)
28. Becker, J., Breuker, D., Delfmann, P., Dietrich, H.A., Steinhorst, M.: Identifying business process activity mappings by optimizing behavioral similarity. In: AMCIS 2012. (2012)
29. Leopold, H., Niepert, M., Weidlich, M., Mendling, J., Dijkman, R., Stuckenschmidt, H.: Probabilistic optimization of semantic process model matching. In: BPM 2012. (2012) 319–334
30. Gacitua-Decar, V., Pahl, C.: Automatic business process pattern matching for enterprise services design. Services II (2009) 111–118