Discovery of Analogical Knowledge for the Transfer of Workflow Tasks

Mirjam Minor
Goethe University Frankfurt
Business Information Systems
Frankfurt am Main, Germany
Email: minor@cs.uni-frankfurt.de

Miriam Herold
Goethe University Frankfurt
Business Information Systems
Frankfurt am Main, Germany
Email: herold@cs.uni-frankfurt.de

***Abstract*—Analogical knowledge considers functional properties of objects in contrast to literal similarity which compares the degree of featural overlap. A classical example from Gentner's structure mapping theory is "An electric battery is like a reservoir" [1]. Acquiring analogical knowledge in a computational approach is a challenging task. In this paper, we present a solution that combines learning with knowledge engineering. The proposed knowledge discovery approach uses word embeddings to learn analogy on workflow tasks. The resulting knowledge is integrated with an ontology for the purpose of workflow transfer across application domains. A case study is conducted on the two example domains 'passenger and baggage handling at the airport' and 'SAP warehouse management'. The experimental results on comparing the computational analogy with a golden standard from a knowledge engineering expert are quite promising and provide a proof-of-concept for the feasibility of the approach.**

I. Introduction

During quarantine times, digital transformation is being perceived on a growing scale and becoming a matter of concern in almost every business. Appropriate business processes are an important part of a business strategy that matches the challenges of digital transformation [2]. For instance, during a pandemic people stay more at home and order goods online instead of shopping in retail shops. The growing amount of online business is a challenging issue, especially for smaller logistics and warehouse companies, as their operations are more hands-on than well-organized. To stay competitive, they need to use business process automation and workflow technology[1]. Designing workflows is a mandatory prerequisite for this but a time-consuming and difficult task. Small companies also sometimes lack the necessary know-how in designing workflows.

AI methods have a huge potential to provide assistance for workflow designers. A large body of research exists on the reuse of workflow models by *process-oriented case-based reasoning* (PO-CBR) [4]. In PO-CBR, workflows are recorded as cases that can be reused in a similar context. Case-based reasoning is considered analogical reasoning within a particular domain [5]. Recently, analogical transfer of workflows has been investigated also between different domains [6]. If the source domain differs from the target domain, analogical knowledge is required to transfer the procedural knowledge that is stored in workflows from the source domain into the target domain. Gentner [1] defines an analogy as "an assertion that a relational structure that normally applies in one domain can be applied in another domain."

When we consider airport operators, who already use technologies to achieve digitalized processes [7], and compare their activities with the warehouse domain, the task "load luggage on transporter" as part of the passenger check-in process of an airport has a corresponding task "load on truck" in the target domain of warehouse logistics. Obviously, the task can not be reused literally in the target domain but by analogy. The workflows including the handling of bulky luggage might be useful templates for modeling workflows on the outbound of bulky goods in a warehouse. The tasks in the control flow surrounding the luggage size check, sorting the luggage in normal or bulky size and treating it accordingly, have also corresponding tasks in the warehouse domain. Hence, the analogical knowledge on tasks is extremely useful for an automated or semi-automated transfer of entire workflows.

The elicitation of analogical knowledge on workflow tasks is a challenging issue for knowledge engineers. This publication proposes a knowledge discovery approach for analogical knowledge on workflow tasks. It uses word embeddings to extract computational analogy from documents. Computational analogy denotes analogy that has been hypothesized by a computational approach such as knowledge discovery. The contributions of the publication are the following:

- a novel approach to employ machine learning for knowledge discovery using word embeddings
- an ontology-based approach to use the resulting analogical knowledge across the boundaries of application domains
- a case study with experiments on the two example domains 'passenger and baggage handling at the airport' and 'SAP warehouse management'.

The remaining paper has the following structure: In Section 2, we discuss some related work. The third section describes the analogical transfer theoretically and explains it on samples from the two examined domains. Section 4 introduces the

[1]Workflows are "the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules" [3].

discovery of similarity-based knowledge in more detail. The fifth section gives an overview on the data, experimental setup and shows the results of our experiments. Finally, in Section 6 we draw some conclusions and give an outlook on future work.

II. Related work

In the following, related work from two fields is discussed that are highly relevant for this publication. First, some work on ontology matching is considered. Second, the state of the art on learning analogical knowledge is reported.

"Ontology matching aims at finding correspondences between semantically related entities of different ontologies." [8, viii]. Ontology matching has mainly been motivated by the semantic heterogeneity problem. It occurs, for instance, in the context of the Semantic Web when ontologies from different origins are supposed to interoperate. On the one hand, ontology matching has a diffenrent goal than our problem since the desired correspondences in ontology matching are based on similar features as described by Tversky [9]. Analogy differs from literal similarity since it describes corresponding functional properties in different contexts. In contrast to ontology matching, learning analogical knowledge aims to find correspondences of functional nature. A well-known example of an analogy is "An electric battery is like a reservoir" [1]. The purpose of both is to store energy while their size, material and color differ significantly. On the other hand, bridging heterogeneity between ontologies is worth a deeper discussion in the following since some of the basic findings on ontology matching are also useful for handling analogical knowledge. Euzenat & Shvaiko [8, p. 37] identify four types of heterogeneity:

- syntactic (different ontology languages),
- terminological (variations in names),
- conceptual (difference in coverage, difference in granularity, difference in perspective), and
- semiotic/pragmatic (intended use of the entities) heterogeneity.

In a broader sense, discovering analogical knowledge can be considered bridging conceptual heterogeneity. Obviously, different application domains like those in our analogical transfer approach have a significant difference in coverage since they describe different portions of the world (cmp. the micro-theories in CYC [10] and their discussion with respect to ontology matching in Benerecetti [11]). There are commonalities in representing the results of ontology matching and the results of learning analogy. Both can be formalized as an ontology alignment. The research on ontology matching has inspired our basic, relational representation form for analogy. Further, ontology matching provides solutions for further representational issues that might become interesting in our future work, such as provenance or quality.

Very closely related work to our approach has been reported in the literature on *learning analogical knowledge*. Fam & Lepage [12] describe an approach to learn analogical knowledge on the morphology of words from a text corpus. The scope is slightly different from our work that focuses on semantical analogy of workflow tasks. Fam & Lepage's approach is based on vectors of formal features of the words like the number of occurrences of the different characters. This method is similar to our work since both use statistical language modeling. However, our approach uses a word embedding model in contrast to a vector model with rather syntactic features. Further, the authors consider computational analogy as a possible way of explaining unseen words, which is a different goal than our goal of analogical transfer.

The three following approaches are examples of learning analogical knowledge based on word embeddings like our approach. The three approaches have in common that they solve an analogical equation to fill empty cells in a proportional analogy. Determining the *solution of an analogical equation* for the proportional analogy (*A* is to *B* as *C* is to *?*) via word embeddings in vector space models has been proposed for the first time in [13]. In [14], the estimation of the extent to which a candidate answer belongs to the correct word class is considered in addition to the vector space model. In [15], analogical clusters are produced from an example seed cluster using word embeddings. Four types of analogical relations are considered, including lexicographic semantics like the hypernyms *carrots : vegetables* and *cafe : restaurant*, encyclopedic semantics like the country-language relation *andorra : catalan* , *argentina : spanish* and *australia : english* as well as two types of morphological relations namely derivational like *seasonal : seasonally*, *modest : modestly* and inflectional like *play : played*, *recover : recovered*. The latter approach has been very inspiring for our work since the lexicographic and encyclopedic analogies go across domains via the commutative property that holds in their work:

$$A : B :: C : D \Leftrightarrow \begin{cases} A : B = C : D \\ A : C = B : D \end{cases}$$

.

In contrast to proportional analogies which comprise four elements at least, our approach tries to find correspondences between two analogical elements only, i.e. to find pairs of corresponding workflow tasks where "A B is like an A" holds according to Gentner's structure mapping theory [1]. Further, our approach for task names considers multi-word expressions instead of single words.

III. Analogical transfer

The goal of our work is to elicit analogical knowledge to port procedural knowledge across domains. Prior to presenting the novel knowledge discovery approach in Section IV, we will sketch some basic ideas how the analogical transfer of pocedural knowledge can be accomplished. Further details on the transfer process are described in the literature [6].

Transfer learning (TL) addresses the question of "how the things that have been learned in one context can be re-used and adapted in related contexts" [16, p. 5]. The *source domain* D_S denotes the context in which knowledge is available at a rich, mature level. The *target domain* D_T provides a context where the knowledge is sparse. In TL for PO-CBR, the procedural knowledge to be transferred is contained in workflows. D_S comprises a large repository of workflows called WF_S. The workflow repository WF_T in the target domain is small and is to be enriched by a set of transferred workflows denoted by $WF_{T'}$.

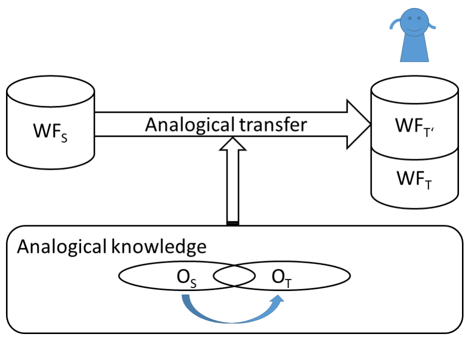

Fig. 1. Interactive analogical transfer.

As depicted in Fig. 1, the transfer process is performed by *analogical transfer* in an interactive manner. The user is a workflow designer who aims to model workflows in D_T. The analogical transfer process creates entire workflows or parts of workflows in $WF_{T'}$ from workflows in WF_S, which are approved and further developed by the user. *Analogical knowledge* provides the translation rules to transform the workflows across the domains. Analogical knowledge on workflow tasks is used to substitute tasks from D_S by tasks from D_T. According to Gentner's structure mapping theory [1], analogical transfer preserves the relations between objects. The order of tasks remains unchanged during transfer of workflows.

We assume that an ontology O is available (or can be created) as vocabulary for both domains. O covers the workflow tasks and the data items of the workflows in WF_S and WF_T. $O_S \subseteq O$ denotes the part of the ontology whose concepts belong to the source domain D_S. $O_T \subseteq O$ denotes the concepts of the target domain D_T. The analogical knowledge is represented as a correspondence $R \subseteq O_S \times O_T$ that aligns concepts from O_S to concepts in O_T by analogy:

$$R = \{(c, c') | c' \text{ is an analogy to } c\}. \quad (1)$$

For example, Fig. 2 depicts a clipping of a workflow 'Luggage handling departure' in the airport operation domain. The workflow is designed in Business Process Modeling Notation (BPMN) [17]. It describes the handling of bulky luggage previously to the departure of a passenger. If the size of the luggage has been classified as '*bulky*' during luggage control ('*Control luggage*') the standard luggage handling procedure ('*Standard luggage handling*') using a conveyor belt system is not appropriate. Instead, the luggage is moved to a special loading area ('*Move luggage to loading area*') where it is loaded on a transporter ('*Load luggage on transporter*').

This example can be ported from the airport operation domain as a source domain D_S to the warehouse domain D_T.

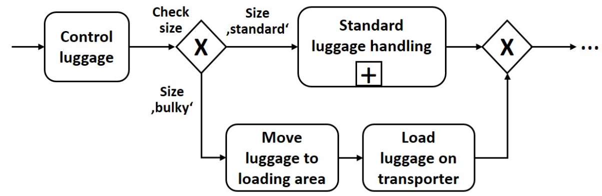

Fig. 2. Workflow 'Luggage handling departure' in BPMN.

Ideally, analogical knowledge on the four depicted tasks is represented in R, including the following pairs:

('*Control luggage*', '*Inspect storage items*')
('*Control luggage*', '*Check incoming goods*')
('*Standard luggage handling*', '*Standard packaging*')
('*Move luggage to loading area*', '*Move to picking area*')
('*Load luggage on transporter*', '*Load truck*')

Each pair is regarded a substitution rule for tasks. A possible result of using R to transfer the workflow clipping from Fig. 2 is the artificial workflow in $WF_{T'}$ depicted in Fig. 3. It addresses the inspection and handling of oversized cargo during the outbound of goods from a storage.

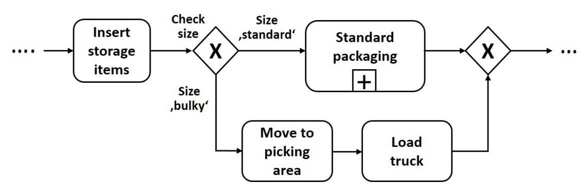

Fig. 3. Artificial workflow 'Outbound of goods' in BPMN.

IV. Similarity-based Knowledge Discovery

The proposed knowledge discovery approach aims to support the knowledge engineers in acquiring analogical knowledge on workflow tasks. Knowledge engineers may be advanced workflow designers who are users of the analogical transfer as described in the previous section.

The knowledge discovery process generates computational analogies $R_{comp} \subseteq O_S \times O_T$:

$$R_{comp} = \{(c, c') | c' \text{ is a computational analogy to } c\}. \quad (2)$$

R_{comp} is hypothesized by means of a similarity function $sim : O_S \times O_T \to \mathbb{R}$ which induces a ranking of all tasks in O_T for a source task in O_S. R_{comp} comprises the k best matching tasks for each particular source task with respect to sim. The computational analogies are presented to the knowledge engineer source task by source task. An illustrating sample is depicted in the discussion of experimental results in Fig. 11. The human expert finally decides which task pairs are retained from all potentially analogical tasks in R_{comp}.

The similarity function for two workflow tasks is based on the words that are contained in their task names:

$$sim(c, c') = \Phi sim(w_i, w'_j). \quad (3)$$

w_i is a word as part of the sequence of words $w_1, ..., w_l$ forming the task name of c, w'_j is a word in $w'_1, ..., w'_m$ of c', and Φ is an aggregation function over a local similarity function for words. The values for the local similarity $sim(w_i, w_j)$ are derived from a *word2vec* model [18] that provides vector space embeddings of words. The embeddings are result of statistical language modelling procedures.

The intuition behind this idea is the assumption, that if single words have a similar context and hence are located close to each other in a vector space, also the whole workflow tasks can be analogically aligned, even if they are from different domains. Hence, the provenance of the computational analogy between tasks is a large text corpus.

V. Experiments

In this work, we introduce word embeddings as a means for computational discovering of analogical knowledge in workflow tasks. As a proof-of-concept, we have conducted a case study experiment using data from real workflow repositories. The computational analogies R_{comp} created by our approach are evaluated in comparison to task analogies R specified by a human expert. Three variants of R_{comp} using different similarity functions based on word embeddings have been investigated. Our hypothesis for the case study is the following:

Word embeddings are a suitable means to discover analogical knowledge that aligns workflow tasks from different domains.

The detected alignments can be used for transferring workflows from the source into the target domain. The transferred knowledge can help the workflow modeller in creating new workflows or adapting the sparse pre-existing models in the target domain.

A. Experimental data

For our experiments, we use workflows from two different domains: passenger and baggage handling at the airport as D_S and warehouse management as D_T. The workflows in the airport domain WF_S are mainly based on Richter's book on baggage logistics at airports [19]. They describe passenger and baggage handling, check-in procedures, transport of baggage through the airport and loading into the aircraft. Following the textual descriptions in the book, we modelled 30 workflows with 124 different task descriptions. The base for WF_T in the warehouse domain is the SAP Best Practices Explorer [20]. This repository contains 20 BPMN workflows with 149 different tasks, describing mostly inbound and outbound of goods in/from various kind of warehouses, replenishment, scrapping, inventory and consumption of materials during production. Fig. 4 summarizes the workflow data in the experimental repositories. In our previous work [21], we showed that processes from these two domains have sufficient overlap and are suitable candidates for transfer learning.

Workflow data		
	Airport Domain	Warehouse Domain
Workflows	30	20
Tasks	124	149
Unique words	166	253

Fig. 4. Summarization of workflow data in the experimental repositories.

For the training of word vectors, we use a corpus with technical texts from various sources. In the airport domain we used texts from [19], [22], [23], [24] and [25] describing logistics at the airports, passenger movement and strategic positioning of airports. The texts in the warehouse domain [26], [27] describe mainly the SAP warehouse management system and IT-based logistics. We input the texts with the open source library Apache Tika[2] and produced one big string. This string serves as a base for the preprocessing step. Preprocessing is an important issue in data preparation and have a strong impact on the quality of results [28]. For our purposes we use a pipeline with the following order:

1) Tokenization of sentences
2) Removing of not required punctuation and symbols
3) Tokenization of words
4) Removing of stop words
5) Lemmatization

Steps 1, 3 and 4 have been implemented by means of the Python library Natural Language Toolkit (NLTK)[3]. Step 2 uses the regular expression [^a-zA-ZäöüÄÖÜß] to filter out non-character symbols of the German language. Step 5 is done using the lemmatization module of the library spacy[4]. After passing all the preprocessing steps the training corpus has the size illustrated in Fig. 5.

Training data	
Sentences	48.403
Words	674.944
Unique words	54.106

Fig. 5. Summarization of training data

In this stage of our project, we decided not to use pretrained word embeddings but train our own model. One reason is that the domain specific corpus reaches higher coverage for the test vocabulary (in the airport domain 76% and in the warehouse domain 78%). Additionally, the technical texts help in disambiguation of words. For instance the German word 'laden' stands for 'loading', which is correct in our context but used in a common language it can have a misleading meaning of a 'shop'. To obtain comparable results we preprocess the test data with the same pipeline as the training data. Especially removing stop words from the task labels and the

[2]https://tika.apache.org
[3]https://www.nltk.org/api/nltk.tokenize.html
[4]https://spacy.io/api/lemmatizer

lemmatization improves the chance to find correct alignments between the tasks from the two examined domains.

B. Experimental set-up

In the next step, we train the Word2Vec model and calculate word embeddings with the open source library Gensim [29]. The results are vectors for task words, which further can be used for the determination of cosine similarity and the Word-Mover's Distance. In our experiments we compare the results of following similarity functions:

a. Cosine similarity aggregated with Greedy algorithm
b. Cosine similarity aggregated with Kuhn-Munkres method
c. Similarity induced by Word-Mover's Distance

For the first two methods we require local cosine similarity values based on word embeddings. For the task examples 'Load the luggage on the transporter' (in German 'Gepäck auf Transporter laden') and 'Load the truck' (in German 'LKW beladen') the local similarity values between single words are depicted in Fig. 6.

	LKW	beladen
Gepäck	0.298597	0.385093
Transporter	0.610036	0.714201
laden	0.464546	0.516442

Fig. 6. Example of local cosine similarity values

Based on these local similarity values, we compute aggregated values between the task pairs with the help of two different algorithms: Greedy and Kuhn-Munkres method. Greedy algorithm provides results in a reasonable computational complexity, but it does not always find the optimal solution. It is a good approximation to Kuhn-Munkres algorithm, also called the Hungarian method, which computes results in polynomial time and always finds the optimum [30].

The Hungarian method is usually used for minimization problems (for instance obtaining minimal costs in a cost matrix) but it also can be applied for maximization. To find a maximum we first negate the values in the matrix. To make it nonnegative, we add the highest value from the initial matrix to every negative value. In the next step we proceed the Kuhn-Munkres method in the usual way and aggregate local similarity values to a global similarity between the workflow task pairs. Even though the two methods obtain similar results, we decided to implement both of them for flexibility reasons. In case of a huge amount of test data with very long task labels, the Greedy algorithm is expected to provide good approximative results in a reasonable computational complexity.

The third method, Word-Mover's Distance (WMD) is also based on word embeddings [31]. The method is used for measuring dissimilarity between two text documents. In WMD, the document distance is formulated as a linear optimization problem. The objective function calculates a minimum cumulative distance considering the term frequencies in the documents and the Euclidian distance of the according word vectors from the Word2Vec model. In our experiments, we consider workflow task names as documents and compute WMD between the task pairs. The pairs with the shortest cumulative distance are semantically similar and can indicate an analogy between the task labels. For the computation we use the method *wmdistance()* and use the negated distance values as similarity values. We use Gensim's implemenation of the WMD [32] and the library PyEMD based on [33]. For the above task pair example we depicted the Euclidean distance $||w_i - w'_j||_2$ for the words w_i from the German task name of 'Load the luggage on the transporter' and the words w'_j from the German task name of 'Load the truck' in Fig. 7.

	LKW	beladen
Gepäck	1.184401	1.108970
Transporter	0.883135	0.756041
laden	1.034847	0.983421

Fig. 7. Euclidean distance between example word vectors used for WMD

Utilizing these methods we determine the most similar task pairs from both domains and evaluate them against a golden standard. The golden standard *R* is a collection of 100 alignments between the tasks from both domains. It was created manually by a domain expert and is defined as follows:

$$R \subseteq O_S \times O_T$$

To illustrate the golden standard, Fig. 8 shows an excerpt translated in English.

Airport Domain	Warehouse Management Domain
Check the luggage manually	Inspect cargo
Load the luggage on transporer	Load truck
Load the luggage on transporter	Load handling unit
X-Ray the luggage	Inspect cargo
Store the luggage	Store the goods

Fig. 8. Excerpt from golden standard

In the evaluation, we determine how many task pairs from the golden standard are comprised under the k-best matchings and compute the accuracy *A* defined in the following equation:

$$A = \frac{|R_{comp} \cap R|}{|R|} \tag{4}$$

Like for conventional accuracy measures, the values of A remain between 0 and 1 under the assumption that the maximum number of analogical pairs in R with the same source task does not exceed the number k of best matchings recorded in R_{comp} (cmp. Section IV).

C. Experimental results

The results of our experiments are summarized in Fig. 9 and Fig. 10. Fig. 9 shows how the three similarity functions perform depending on the level of *k*. For example, when we consider the 5 best matchings, the Kuhn-Munkres algorithm and the WMD detect approximately 40% of the golden standard alignments. It seems that chosing different similarity measures has only little impact on the accuracy of the results.

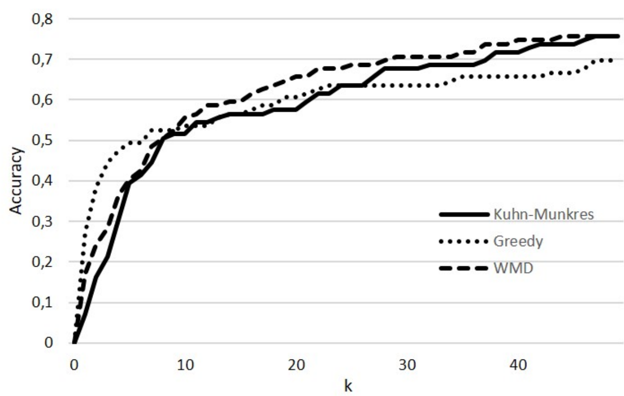

Fig. 9. Model comparison

If we look at the accuracy values for k higher than 10, the model detects only slightly more alignments from the golden standard. For reasons of readability for the knowledge engineers, we have chosen to restrict k to 10. Fig. 10 shows the accuracy in % for $k \leq 10$.

k	Greedy	Kuhn-Munkres	WMD
1	27	7	17
2	38	16	24
3	44	21	28
4	47	30	36
5	49	39	40
6	49	41	42
7	53	44	48
8	53	51	51
9	53	52	53
10	54	52	56

Fig. 10. Accuracy in %

In addition, the deeper analysis of the results shows that especially for higher similarity values the model is also able to find alignments that are not in the golden standard. But at a closer look they comprise task analogies in a broader sense and can serve as a good suggestion for the modeler who creates workflows in the target domain. Fig. 11 shows the 10 best matching suggestions for the task 'Load luggage on transporter' aggregated with Kuhn-Munkres method. The best two results are marked dark grey and are direct hits, means they are comprised in the golden standard (compare Fig. 8). The light grey marked tasks are not in the golden standard but they are analogical in a wider sense. They also could serve as a suitable suggestion for a workflow designer. The remaining task suggestions with no marking are not usable. The aggregated similarity values are based on German word embeddings, the task names are translated into English for better understanding.

10 best matchings for the task ‚Load luggage on transporter'	
Suggested analogical tasks	**Aggr. cos. sim.**
Load handling unit	0.76185
Load truck	0.58937
Putaway of products from handover point to high rack pallet buffer	0.50757
Unload incl. repacking and labeling	0.50587
Process transport unit: Check-in and Arrival at door	0.49937
Release the picking wave	0.48609
End loading	0.48047
Truck arrives at the check-in and drives to the door	0.46599
Move pallets from staging area to handover point	0.46594
Putaway of products to the high rack storage (upper levels)	0.45849

Fig. 11. 10 best matchings aggregated with Kuhn-Munkres method

Consequently, we are conviced that word embeddings are a suitable means to find computational analogies between task pairs. One of the limitations of our approach is the coverage of the test vocabulary in the training corpus. Not covered words produce null values in the measurement of local values and hence affect the aggregated results. In the future, we consider to accomodate pretrained models and expand them with our technical texts to reach higher coverage and examine the impact on the accuracy. Another issue worth considering is the specification of a threshold for aggregated similarity values. A threshold might reduce the amount of detected alignments but provide more accurate results.

VI. Conclusion

In this paper, we presented a knowledge discovery approach for analogical knowledge on tasks from different workflow application domains. The computational analogy is detected by means of vector embeddings learned from texts. It serves as a modeling assistance for knowledge engineers preparing the knowledge resources required for the analogical transfer of workflows. Some experiments have been conducted in a case study comparing the computational analogy R_{comp} with a golden standard analogy R comprising 100 pairs of analogical workflow tasks. In absolute terms, the accuracy values in the experimental results seem not very competitive. Indeed they are quite promising with respect to their intended purpose. Suggesting half of the task pairs contained in the golden standard to the knowledge engineers provides a valuable support and a significant reduction of knowledge engineering efforts. The contribution highlights the huge potential of using statistical language models such as word embeddings for interactively supported knowledge engineering.

The analogical transfer of workflows is ongoing work. In future work, we will investigate the integration of further knowledge from the ontology with the discovery of analogical knowledge on tasks. As reported in the literature

[6], generalization and specialization are suitable means for workflow transfer. Replacing workflow tasks by generalized tasks (for example “move luggage to loading area” by “move item”) provides opportunities to use analogy between tasks not only at the level of leafs in the ontology but also at higher levels within the hierarchy of concepts. The discovery of such structural analogy in O_S and O_T is really an exciting research strand. Beyond the isomorphic transformation of workflows, knowledge on abstraction and refinement could be learned.

ACKNOWLEDGMENT

This publication is a contribution to the project EVER2, funded by Deutsche Forschungsgemeinschaft (DFG) under the project number MI 1455-2-3. The authors also would like to thank Nils Gormsen for his support in the development of experiments in Section V.

REFERENCES

[1] D. Gentner, “Structure-Mapping: A Theoretical Framework for Analogy,” *Cognitive Science*, vol. 7, no. 2, pp. 155–170, Apr. 1983.

[2] T. Hess, C. Matt, A. Benlian, and F. Wiesböck, “Options for Formulating a Digital Transformation Strategy,” *MIS Q. Executive*, vol. 15, no. 2, pp. 123–139, 2016.

[3] {Workflow Management Coalition}, “Workflow Management Coalition Terminology & Glossary (Document No. WFMC-TC-1011),” 1999, last access 05-23-2007. [Online]. Available: http://www.wfmc.org/standars/docs/TC-1011_term_glossary_v3.pdf

[4] M. Minor, S. Montani, and J. A. Recio-García, “Editorial: Process-oriented Case-based Reasoning,” *Inf. Syst.*, vol. 40, pp. 103–105, Mar. 2014.

[5] S. Beller and Hans Spada, “Denken [Reasoning],” Stuttgart, pp. 114–124, 1996.

[6] M. Minor, R. Bergmann, J.-M. Müller, and A. Spät, “On the Transferability of Process-Oriented Cases,” in *Case-Based Reasoning Research and Development - 24th International Conference, ICCBR 2016, Atlanta, GA, USA, October 31 - November 2, 2016, Proceedings*, ser. Lecture Notes in Computer Science, A. K. Goel, M. B. Díaz-Agudo, and T. Roth-Berghofer, Eds., vol. 9969. Springer, 2016, pp. 281–294.

[7] I. Kovynyov, “Digital Transformation in Airport Ground Operations.” *CoRR*, vol. abs/1805.09142, 2018. [Online]. Available: http://arxiv.org/abs/1805.09142

[8] J. Euzenat and P. Shvaiko, *Ontology Matching, Second Edition.* Springer Berlin / Heidelberg, 2013.

[9] A. Tversky, “Features of similarity.” *Psychological review*, vol. 84, no. 4, p. 327, 1977, publisher: American Psychological Association.

[10] D. Lenat and R. Guha, “Building large knowledge-based systems: Representation and inference in the cyc project: D.B. Lenat and R.V. Guha,” *Artificial Intelligence*, vol. 61, no. 1, pp. 65–79, May 1993.

[11] M. Benerecetti, P. Bouquet, and C. Ghidini, “On the Dimensions of Context Dependence: Partiality, Approximation, and Perspective,” in *Modeling and Using Context*, ser. Lecture Notes in Computer Science, V. Akman, P. Bouquet, R. Thomason, and R. Young, Eds. Berlin, Heidelberg: Springer, 2001, pp. 59–72.

[12] R. Fam and Lepage, Yves, “Morphological Predictability of Unseen Words Using Computational Analogy.” in *ICCBR 2016 Workshop Proc.*, vol. 1815. CEUR-WS.org, 2017, pp. 51–60. [Online]. Available: http://ceur-ws.org/Vol-1815/paper5.pdf

[13] T. Mikolov, W.-t. Yih, and G. Zweig, “Linguistic regularities in continuous space word representations,” in *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*. Association for Computational Linguistics, 2013, pp. 746–751.

[14] A. Drozd, A. Gladkova, and S. Matsuoka, “Word Embeddings, Analogies, and Machine Learning: Beyond king - man woman = queen.” in *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*. ACL, 2016, pp. 3519–3530.

[15] Y. Hong and Y. Lepage, “Production of Large Analogical Clusters from Smaller Example Seed Clusters Using Word Embeddings.” in *Case-Based Reasoning Research and Development - 26th International Conference, ICCBR 2018, Stockholm, Sweden, July 9-12, 2018, Proceedings*, vol. LNCS 11156. Springer, 2018, pp. 548–562.

[16] D. Kudenko, “Special Issue on Transfer Learning,” *KI - Künstliche Intelligenz*, vol. 28, no. 1, pp. 5–6, Feb. 2014.

[17] G. Decker, R. Dijkman, M. Dumas, and L. García-Bañuelos, “The Business Process Modeling Notation,” in *Modern Business Process Automation*, A. H. M. Hofstede, W. M. P. Aalst, M. Adams, and N. Russell, Eds. Springer Berlin Heidelberg, Jan. 2010, pp. 347–368.

[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proc.,*, Yoshua Bengio and Yann LeCun, Eds., 2013.

[19] A. Richter, *Gepäcklogistik auf Flughäfen: Grundlagen, Systeme, Konzepte und Perspektiven*. Springer-Verlag, 2012.

[20] “Sap best practices explorer,” 2019, last visit on 07.10.2020. [Online]. Available: https://rapid.sap.com/bp/#/browse/categories/sap_s%254hana/areas/on-premise/packageversions/RDS_EWM_S4,

[21] M. Herold and M. Minor, “Ontology-based transfer learning in the airport and warehouse logistics domains,” in *Workshop Proceedings of ICCBR 2019*, Otzenhausen, Germany, 2019.

[22] B. Kaluza, H. Winkler, B. E. Rogl, and L.-u. U. K. Ãlpen-Adria- Universität Klagenfurt. Institut für Wirtschaftswissenschaften. Abteilung Produktions, *Strategische Positionierung von Flughäfen im Air-Cargo-Geschäft*. Alpen-Adria-Universität, 2006.

[23] I. Hausladen, *IT-gestützte Logistik*. Springer, 2014.

[24] H. Wagner and S. Kabel, *Mobilität 4.0 - neue Geschäftsmodelle für Produkt-und Dienstleistungsinnovationen*. Springer, 2018.

[25] M. Schultz, *Entwicklung eines individuenbasierten Modells zur Abbildung des Bewegungsverhaltens von Passagieren im Flughafenterminal*. Jörg Vogt Verlag, 2010.

[26] H. B. eV Baumgarten and H. Baumgarten, *Das Beste der Logistik*. Springer, 2008.

[27] M. B. Carter, J. Lange, F.-P. Bauer, C. Persich, and T. Dalm, “Sap extended warehouse management: Processes, functionality, and configuration,” 2010.

[28] M. Van de Kauter, G. Coorman, E. Lefever, B. Desmet, L. Macken, and V. Hoste, “Lets preprocess: The multilingual lt3 linguistic preprocessing toolkit,” *Computational Linguistics in the Netherlands journal*, vol. 3, pp. 103–120, 2013.

[29] R. Řehůřek and P. Sojka, “Software Framework for Topic Modelling with Large Corpora,” in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50, http://is.muni.cz/publication/884893/en.

[30] H. W. Kuhn, “The hungarian method for the assignment problem,” *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[31] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, “From word embeddings to document distances,” in *International conference on machine learning*, 2015, pp. 957–966.

[32] “Word movers’ distance,” last visit on 30.09.2020. [Online]. Available: https://radimrehurek.com/gensim/auto_examples/tutorials/run_wmd.html

[33] O. Pele and M. Werman, “Fast and robust earth mover’s distances,” in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 460–467.