



HAL
open science

Knowledge Fusion of Manufacturing Operations Data Using Representation Learning

Martin Ringsquandl, Steffen Lamparter, Raffaello Lepratti, Peer Kröger

► **To cite this version:**

Martin Ringsquandl, Steffen Lamparter, Raffaello Lepratti, Peer Kröger. Knowledge Fusion of Manufacturing Operations Data Using Representation Learning. IFIP International Conference on Advances in Production Management Systems (APMS), Sep 2017, Hamburg, Germany. pp.302-310, 10.1007/978-3-319-66926-7_35 . hal-01707253

HAL Id: hal-01707253

<https://inria.hal.science/hal-01707253>

Submitted on 12 Feb 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Knowledge Fusion of Manufacturing Operations Data using Representation Learning

Martin Ringsquandl¹, Steffen Lamparter² Raffaello Lepratti³, and Peer Kröger¹

¹ Ludwig-Maximilians Universität, Munich, Germany,
martin.ringsquandl@gmail.com

² Siemens Ag, Corporate Technology, Munich, Germany

³ Siemens Industry Software S.r.l., Digital Factory, Genoa, Italy

Abstract. Due to increasingly required flexibility in manufacturing systems, adaptation of monitoring and control to changing context such as reconfiguration of devices becomes more important. Referring to the usage of structured information on the Web, digital twin models of manufacturing data can be seen as knowledge graphs that constantly need to be aligned with the physical environment. With a growing number of smart devices participating in production processes, handling these alignments manually is no longer feasible. Yet, the growing availability of data coming from operations (e.g. process events) and contextual sources (e.g. equipment configurations) enables machine learning to synchronize data models with physical reality. Common knowledge graph learning approaches, however, are not designed to deal with both, static and time-dependent data.

In order to overcome this, we introduce a representation learning model that shows promising results for the synchronization of semantics from existing manufacturing knowledge graphs and operational data.

Keywords: Representation Learning, Digital Twin, Knowledge Fusion

1 Introduction

The ubiquitous availability of data empowers manufacturing companies to embrace advanced data analytic technologies that allow to monitor, predict, and optimize manufacturing operations. Still, ensuring semantic interoperability within hardware-software integrated cyber-physical systems (CPS) and management applications requires extensive manual data modeling effort, thus introducing and maintaining these technologies is challenging for manufacturers [7]. For example, today, deploying a new device for machine condition monitoring at a shop floor means manual effort to model this device and all of its signals throughout several software applications (e.g. SCADA, MES). Otherwise, physical reality is not correctly reflected in existing models and there is no semantic interoperability between applications.

Recently, descriptive data models have been revitalized as part of a digital representation of physical systems, the so-called *digital twin*, which allows systems

to discover, inherit, evaluate and share information across different sub-systems [2]. From a data modeling perspective, structured information of digital twins can be represented as knowledge graph (KG), where relations and entities follow well-defined vocabularies and semantics.

Knowledge graphs are commonly understood as publicly-accessible Linked Data resources – prominent examples are Wikidata⁴ and WordNet⁵. Similarly, Manufacturing Execution Systems (MES) and engineering platforms that are built upon sizable relational databases can be seen as domain-specific knowledge graphs, when lifted to a semantic schema [5]. Such a *manufacturing knowledge graph* should be able to automatically acquire updated information based on different operational data sources (e.g. SCADA, PLCs, etc.), even if these data sources are not aware of their semantics.

Continuing the machine monitoring example: By observing data coming from the newly added device (e.g. events) the KG should automatically recognize the type of device, its location, or its capabilities and therefore allow other applications to adapt to this updated context.

Machine Learning in KGs has emerged recently with the goal to enable automated integration of new facts into KGs without manual modeling efforts [9]. When multiple data sources are used to extract information, the problem further extends to so-called *knowledge fusion* [3]. The same problems apply to models in manufacturing systems that need to be in-sync with physical reality reflected by multiple operational data sources [4]. In this paper, we present an approach to support fusion of information coming from operational data sources with manufacturing KGs by learning latent representations of entities. The goal is to offer automated recommendations on how to integrate unknown entities into the existing structure of the KG and thus keeping the digital twin in-sync without manual modeling effort. Ultimately, this is beneficial to monitoring and management applications that rely on a immediately aligned digital representation of the manufacturing system.

2 Motivation Scenario

In this section we present an example scenario that motivates the application of machine learning (knowledge fusion) to manufacturing KGs in conjunction with operational data sources.

Consider an automated production line at a discrete manufacturing facility, consisting of multiple production units that can be configured to produce several variants of a product. The manufacturing KG (e.g. provided by an MES) of this production line gives information about device topology and processes executed by each of the production units, whereas a SCADA system observes sequences of events during operation. As shown in Figure 1, at the bottom, sequences of events are continuously generated and aligned to entities in the manufacturing KG. Entities and their relations are denoted as triples (**head-entity, relation,**

⁴ wikidata.org

⁵ wordnet.princeton.edu

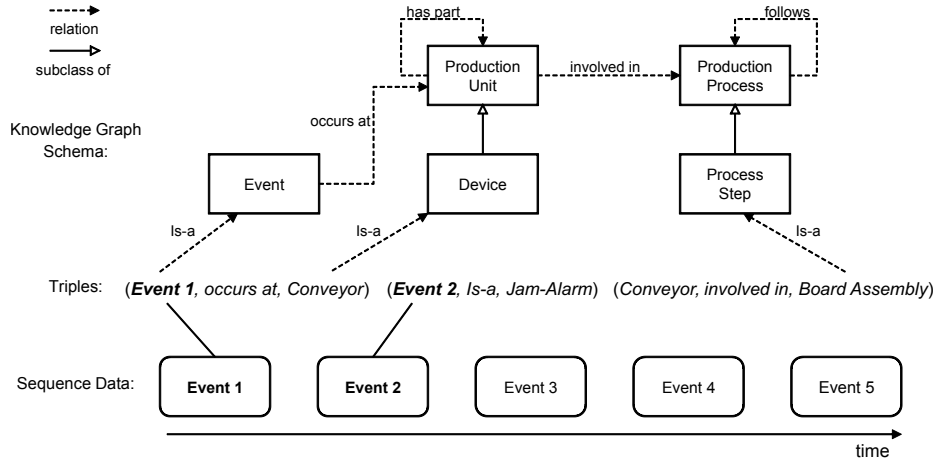


Fig. 1: Sequence data entities aligned to triples in the knowledge graph

tail-entity), in the middle of the Figure. The schema (classes and relations) of the KG is shown on top of the entities using a simplified class diagram notation. For example, the triple (*Event 1*, *occurs at*, *Conveyor*) in the KG states that entity *Event 1* occurs at entity *Conveyor*. Additionally the conveyor entity is modeled as device that is involved in the board assembly process.

Assuming a new device is deployed to the production line to monitor temperature measurements of the conveyor. As production resumes, events of this new device are continuously observed, but they are lacking semantic alignment to the existing KG. Figure 2 shows a new sequence of events, where unknown entities in the triples are denoted with question marks. Here, the class of the unaligned event *Event 2* and its source (device) are unknown, (*Event 2*, *is-a*, ?), respectively (*Event 2*, *occurs at*, ?).

However, the distribution of events in the sequence data should give an indication about which device is most likely to be hold responsible (in this case the conveyor). Since other conveyor events are assumed to co-occur in similar fashion as the new monitoring events, this information can be exploited to re-engineer semantics. Presuming one could obtain a vector representation of all involved entities (events, devices, etc.), it would be possible to calculate a *similarity* between *Event 1* and *Event 2* that would allow to infer that both are related to the conveyor entity in the KG. The representation learning approach in the following is motivated by learning latent entity embeddings that reflect such similarity.

3 Problem Statement

In this section, we formally define the problem of learning joint representations of entities in KGs and operations data of manufacturing systems.

A **knowledge graph**, denoted as \mathcal{KG} is a directed graph with labeled edges.

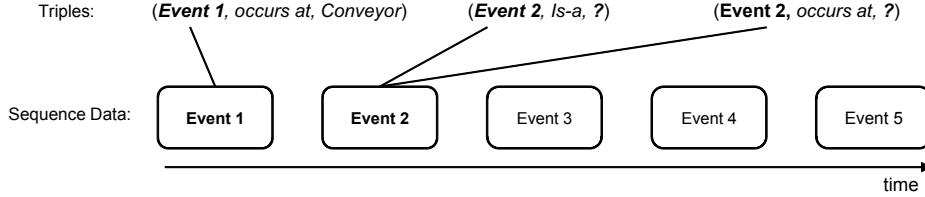


Fig. 2: Observing new events with unknown semantics

Each edge is represented in form of *triples* (h, r, t) that indicates the existence of a relationship between the head entity h and the tail entity t by the labeled relation r . Head and tail are contained in the set of entities $h, t \in \mathcal{E}$ and each relation respectively in the set of relations $r \in \mathcal{R}$.

A **sequence data set**, denoted as $\mathcal{D} = \{(x_1, \dots, x_i, \dots, x_m)_j^T\}$, is a set of sequences, where each sequence consists of an ordered set of event entities x_i . The length m of each sequence can vary depending on a sequence window size. It is implied that there exists a mapping of event entities x_i to entities in \mathcal{E} , i.e. event entities are also represented in the KG.

Knowledge Graph Embeddings Given \mathcal{KG} , the problem of learning **knowledge graph embeddings** is to encode all entities in \mathcal{E} and relations in \mathcal{R} in a continuous low-dimensional vector space, i.e. $\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$ and relation $\mathbf{r} \in \mathbb{R}^d$. In order to learn useful representations, a meaningful distance measure has to be employed, e.g. in the original TransE model [1], $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$. This means that *translating* entity h with relation r should end up close at its tail entity t in the latent d -dimensional space. It has been shown that these *translation embeddings* can be effectively learned by using a ranking loss with the intuition that $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ should be close for true triples and far apart for false/unknown ones. Formally, the learning objective is formulated as minimizing a margin-based ranking loss:

$$\mathcal{L}_{KG} = \sum_{(h,r,t) \in \mathcal{KG}} \sum_{(h',r,t') \in S'_{h,r,t}} \max(0, 1 + \text{dist}(\mathbf{h} + \mathbf{r}, \mathbf{t}) - \text{dist}(\mathbf{h}' + \mathbf{r}, \mathbf{t}')) \quad (1)$$

where $\text{dist}(\cdot)$ is some distance function (e.g. Euclidian) and $S'_{h,r,t}$ is a set of negative samples, i.e. artificially constructed false triples by replacing h or t with a random entity. This loss is minimized when the translation of correct triples is closer than that of unknown ones by a constant margin, here 1.

Sequential Data Embeddings Given \mathcal{D} , the problem of learning **sequential embeddings** of entities x_i is similar to knowledge graphs, i.e. encode all entities in the same low-dimensional vector space, $\mathbf{x}_i \in \mathbb{R}^d$, where semantically similar entities should end up close to each other in this latent space. Learning this kind of embeddings follows the distributional semantics hypothesis which states that similar entities occur in similar context. This has been one of the key ideas in the field of Natural Language Processing (NLP), since these embeddings tend to exhibit natural relations between words (e.g. capture synonymous meanings) [6].

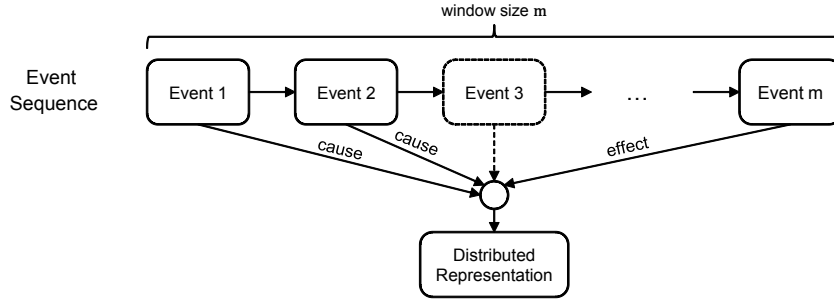


Fig. 3: Representations of event entities are learned from surrounding context

Distributed representations are obtained by assuming that similarity between entities in the data can be modeled with a distribution, formally $P(x_i|W)$, i.e. the occurrence of entity x_i depends on and can be predicted from its surrounding window events W . Figure 3 displays how *Event 3* can be modeled from its surrounding events in a sliding time window of length m through the event sequences. It is assumed that events having similar causes and effects share similar semantics.

Mathematically, the probability distribution of predicting target entity x_i from its surrounding entities can be expressed by a categorical distribution, e.g. the Softmax function:

$$P(x_i|W_i) = \frac{\exp S(\mathbf{x}_i, \mathbf{W}_i)}{\sum_{j \neq i} \exp S(\mathbf{x}_j, \mathbf{W}_i)} \quad (2)$$

, where \mathbf{x}_i is the vector representation of entity x_i and $S(\cdot)$ is some similarity function between entities and their surrounding window entities represented as matrix \mathbf{W}_i . The objective function in terms of loss is given by the negative log likelihood:

$$\mathcal{L}_{Seq} = - \sum_i^n \log(P(e_i|W_i)) \quad (3)$$

Joint Embeddings As the goal of this approach is to jointly model entities in the knowledge graph as well as in the sequential data, we propose a joint learning model that is trained by simply adding both loss terms:

$$\mathcal{L}_{Joint} = \mathcal{L}_{Seq} + \mathcal{L}_{KG} \quad (4)$$

Minimizing the joint loss \mathcal{L}_{Joint} should result in solid embeddings of both, entities in the knowledge graph and the sequence data set. In reality, joint loss minimization is approximated using a state-of-the-art stochastic gradient descent optimizer. The key idea here is that entity embeddings are shared across both tasks and therefore the outcome should reflect co-occurrence of sequential data as well as the structure of the knowledge graph. The architecture of the joint embedding approach is shown in Figure 4, where the $|\mathcal{E}|$ -by- d matrix of

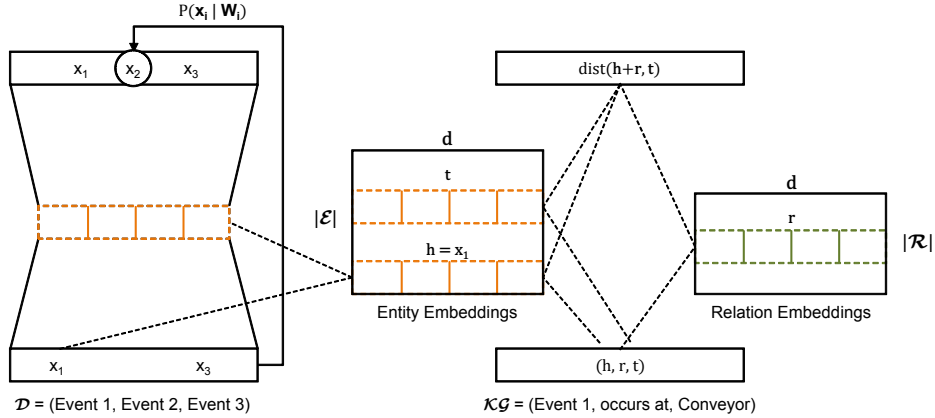


Fig. 4: Architecture of the joint embedding learning model

entity embeddings is located in the center. These embeddings are shared with the prediction model of entities in the sequential data on the left-hand side and the knowledge graph embedding model on the right-hand side. Note that in the depicted example this shared aspect is highlighted with *Event 1* having the same embedding (representation) in both models, i.e. $\mathbf{h} = \mathbf{x}_1$. The $|\mathcal{R}|$ -by- d matrix of relation embeddings on the right-hand side is solely used for the knowledge graph embeddings as it only influences distance calculation between triples.

4 Prototype Evaluation

We evaluated this approach on a real-world manufacturing KG data set coming from an automated assembly line. The event sequences are taken from a SCADA-level Alarms & Events database, whereas the initial KG was extracted from several spreadsheet files and CAD models. The final KG ended up with a size of about 3,700 triples about processes, equipments, and events, whereas the sequential data consisted of 57 thousand events occurrences. A prototypical implementation of the representation learning was implemented using the *TensorFlowTM* library. For performance evaluation, the usual criteria are (cf. [1]):

- Mean Rank: The average predicted rank of the head or tail entity that would have been the correct one (1 indicating perfect rank)
- Hits Top-10: The fraction of predicted ranks that were in the top 10

We compare two models, *KG* (knowledge graph embeddings only) and *KG + Seq* (joint embeddings). In Figure 5, the performance of *KG* and *KG + Seq* are visualized during model training on a hold-out (unseen) test data set of incomplete triples, e.g. (*Conveyor*, *involved in*, ?). It can be seen that the joint model performs better in terms of lower mean rank and higher hits top-10 percentage.

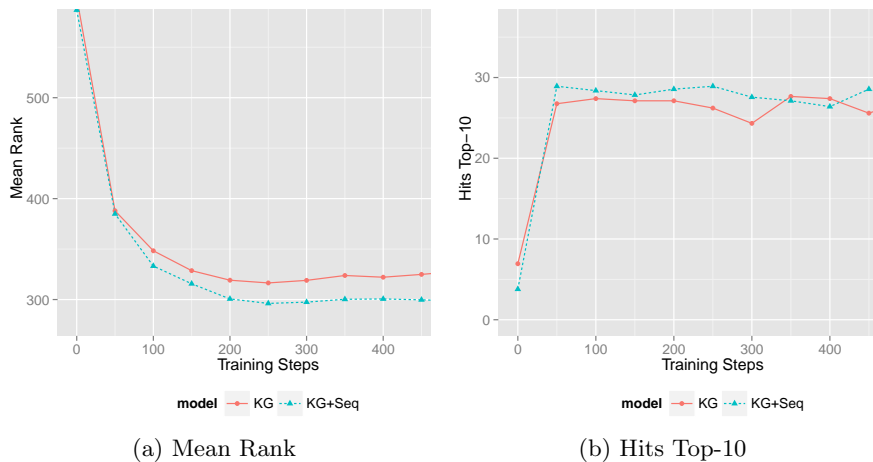


Fig. 5: Evaluation on hold-out test data (unknown triples) during training

Table 1: Models and data sets with and without sequential data

Model	$ \mathcal{KG} $	$ \mathcal{D} $	Test size	Mean Rank	Hits Top-10
KG	3.7k	-	3%	316.46	27.66
KG + Seq	3.7k	57k	3%	296.16	28.92

5 Related Work

We divide related work into two categories, limited to applications and techniques that are close to the one in this work.

Model Learning in Manufacturing Machine learning has been used to discover influencing factors of manufacturing processes[14]. Other works of adapting to changing context have studied monitoring processing times in flexible production systems [11, 10] and more high-level architecture proposals for context extraction and self-adaption of production systems [12]. However, the authors do not specify a concrete methodology on how to extract context knowledge and align it with existing models.

Learning of Knowledge Graph Embeddings Existing learning methods for KGs such as [1] and [9] have been extended to include many-to-many relationships [8] and to incorporate textual information to improve entity representation learning. Recently, word co-occurrences as sequential data were used in KG completion tasks [13]. In contrast to our approach, these works are focused on large-scale knowledge graphs containing noisy information.

6 Conclusion

An approach for automated recommendations for the alignment of semantics coming from operational data and manufacturing KGs was presented. Our model allows to predict missing relations introduced from changes in physical environments and unaligned event semantics, which can be detected and integrated into a global knowledge graph schema, thus lowering manual modeling effort. The joint representation of entities shows promising performance, which is vital for transition to fully automated synchronization, ensuring correct operation of monitoring and other management applications such as scheduling.

References

1. Bordes, A., Usunier, N., Weston, J., Yakhnenko, O.: Translating Embeddings for Modeling Multi-Relational Data. *Advances in NIPS* 26, 2787–2795 (2013)
2. Datta, S.: Emergence of Digital Twins. *arXiv:1610.06467* (2016)
3. Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., Zhang, W.: Knowledge vault: a web-scale approach to probabilistic knowledge fusion. *Proc. of the 20th ACM SIGKDD'14* pp. 601–610 (2014)
4. Hirmer, P., Breitenb, U., Cristina, A.: Automating the Provisioning and Configuration of Devices in the Internet of Things (9), 28–43 (2017)
5. Kharlamov, E., Grau, B.C., Jimenez-Ruiz, E., Lamparter, S., Mehdi, G., Ringsquandl, M., Nenov, Y., Grimm, S., Roshchin, M., Horrocks, I.: Capturing industrial information models with ontologies and constraints. In: 15th Int. Sem. Web Conf. vol. 9982 LNCS, pp. 325–343 (2016)
6. Le, Q., Mikolov, T.: Distributed Representations of Sentences and Documents. *International Conference on Machine Learning - ICML 2014* 32, 1188–1196 (2014)
7. Lechevalier, D., Narayanan, A., Rachuri, S.: Towards a domain-specific framework for predictive analytics in manufacturing. *Proceedings - 2014 IEEE International Conference on Big Data, IEEE Big Data 2014* pp. 987–995 (2015)
8. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning Entity and Relation Embeddings for Knowledge Graph Completion. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence Learning* pp. 2181–2187 (2015)
9. Nickel, M., Murphy, K., Tresp, V., Gabrilovich, E.: A Review of Relational Machine Learning for Knowledge Graph. *Proceedings of the IEEE* (28), 1–23 (2015)
10. Ringsquandl, M., Lamparter, S., Brandt, S.P., Lepratti, R.: Semantic-guided Feature Selection for Industrial Automation Systems. In: *Proc. of the 14th International Semantic Web Conference*. Springer (2015)
11. Ringsquandl, M., Lamparter, S., Lepratti, R.: Estimating Processing Times within Context-aware Manufacturing Systems. In: *Proc. of the 2015 IFAC Symp. on Inf. Contr. Manuf. (INCOM 2015)* (2015)
12. Scholze, S., Stokic, D., Barata, J., Decker, C.: Context extraction for self-learning production systems. *IEEE Int. Conf. on Ind. Inform. (INDIN)* pp. 809–814 (2012)
13. Wang, Z., Li, J.Z.J.: Text-Enhanced Representation Learning for Knowledge Graph. *Ijcai* pp. 1293–1299 (2016)
14. Wuest, T., Irgens, C., Thoben, K.d.: An approach to monitoring quality in manufacturing using supervised machine learning on product state data. *Journal on Intelligent Manufacturing* 25, 1167–1180 (2014)