

# Analyzing longitudinal Data in Knowledge Graphs utilizing shrinking pseudo-triangles

Jens Dörpinghaus\*, Vera Weil<sup>†</sup>, Johanna Binnewitt<sup>‡</sup>

\* Federal Institute for Vocational Education and Training (BIBB), Bonn, Germany,  
 Email: jens.doerpinghaus@bibb.de, <https://orcid.org/0000-0003-0245-7752>

<sup>†</sup> Department of Mathematics and Computer Science, University of Cologne, Germany,  
 Email: weil@cs.uni-koeln.de

<sup>‡</sup> Federal Institute for Vocational Education and Training (BIBB), Bonn, Germany,  
 Email: johanna.binnewitt@bibb.de

**Abstract**—This paper aims to analyze longitudinal data, serial data related to different time points, in knowledge graphs. Knowledge graphs play a central role for linking different data. While multiple layers for data from different sources are considered, there is only very limited research on longitudinal data in knowledge graphs. However, knowledge graphs are widely used in big data integration, especially for connecting data from different domains. Few studies have investigated the questions how multiple layers and time points within graphs impact methods and algorithms developed for single-purpose networks. This manuscript investigates the impact of a modeling of longitudinal data in multiple layers on retrieval algorithms. In particular, (a) we propose a first draft of a generic model for longitudinal data in multi-layer knowledge graphs, (b) we develop an experimental environment to evaluate a generic retrieval algorithm on random graphs inspired by computational social sciences. We present a knowledge graph generated on German job advertisements comprising data from different sources, both structured and unstructured, on data between 2011 and 2021. The data is linked using text mining and natural language processing methods. We further (c) present two different shrinking techniques for structured and unstructured layers in knowledge based on graph structures like triangles and pseudo-triangles. The presented approach (d) shows that on the one hand, the initial research questions, on the other hand the graph structures and topology have a great impact on the structures and efficiency for additional data stored. Although the experimental analysis of random graphs allows us to make some basic observations we will (e) make suggestions for additional research on particular graph structures that have a great impact on the analysis of knowledge graph structures.

## I. INTRODUCTION

**K**NOWLEDGE graphs have been shown to play an important role in recent knowledge mining and discovery, for example in the fields of computational social sciences, digital humanities, life sciences or bioinformatics. They also include single purpose networks (like social networks), but mostly they contain also additional information and data, see for example [1], [2], [3]. Thus, a knowledge graph can be seen as a multi-layer graph comprising different data layers, for example social data, spatial data, etc. In addition, scientists study network patterns and structures, for example paths, communities or other patterns within the data structure, see for example [4]. Very few studies have investigated the questions how multiple layers within graphs impact methods and algorithms

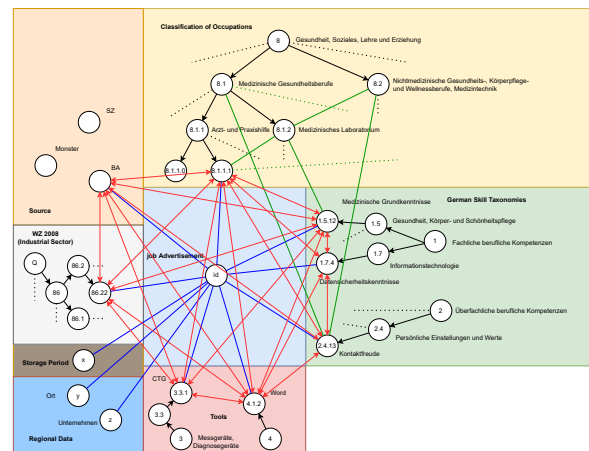


Fig. 1. Knowledge graph representation of German job ads, see Figure 1. In this case the red edges build the pseudo-triangles explaining the context of the job ad.

developed for single-purpose networks, see [5]. In addition, it is possible to store and analyze longitudinal data in knowledge graphs. This is an important topic in medical informatics, for example when working with longitudinal patient records. For example, in [6], the authors use a temporal query language on clinical knowledge graphs. Other authors like [7] use longitudinal patient records within a medical knowledge graph for predictive models. Longitudinal knowledge graphs are related to the versioning of knowledge graphs and other graph-based structures like ontologies. Although research started early on versioning RDF knowledge bases, see [8], only little research has been done on this field. Some attention was paid to the field of the evolution of data structures within information management, see for example [9], [10], [11], and the decentralized collaborative work on knowledge resources, see [12], [13]. Other researchers were interested in parallel world frameworks to analyze scenarios in knowledge graphs, see [14]. Nevertheless, a generic framework for modeling longitudinal data in knowledge graphs is still missing.

In this paper, we focus on an example use case from computational social sciences. In labor market research, ex-

tracting skill requirements from job advertisements (short: job ads) becomes a feasible approach to observe which skills are in demand by employers [15], [16], [17], [18]. Job ads are one way for a company to recruit new employees. Beside general information about the hiring company and the working conditions, they document the current skill needs on the labor market. For a longitudinal view on how skill demands develop, it is necessary to build a model which is capable of not only utilizing these data within a knowledge graph with contextual data, but also for efficient analysis.

We present a detailed overview of the knowledge graph representation in Figure 1. We give a detailed formal definition and overview in the next section.

The main research question of this paper is: How can we model longitudinal knowledge graphs on job ads for an efficient analysis of the development of skills while preserving all contextual data? In order to answer that question, this manuscript investigates both the impact of shrinking triangles in multiple layers and the runtime needed for this approach.

This paper is divided into five sections. The first section gives a brief overview of the research question, state of the art and related work. The second section describes the preliminaries and background. We will in particular introduce knowledge graphs and describe the knowledge graph models on job ads. In the third section, we present the experimental setting and the methods used. The fourth section is dedicated to experimental results and their evaluation. Our conclusions are drawn in the final section.

## II. PRELIMINARIES

The term *knowledge graph* (sometimes also called a *semantic network*) is not clearly defined, see [19]. In [20], several definitions are compared, but the only formal definition was related to RDF graphs which does not cover labeled property graphs. However, a *knowledge graph* is a systematic way to connect information and data to knowledge.

**Definition 1** (Knowledge Graph). *We define a knowledge graph as mixed graph  $G = (E, R)$  with entities  $e \in E = \{E_1, \dots, E_n\}$  coming from formal structures  $E_i$ , like ontologies.*

By using formal structures within the graph, we are implicitly using the model of a labeled property graph, see [21] and [22]. Here, nodes and edges form a heterogeneous set. Nodes and edges can be identified by using a single label or multiple labels, using a mapping  $\lambda : V \cup E \rightarrow \Sigma$ , where  $\Sigma$  denotes a set of labels. We need to mention that both concepts are equivalent, since graph databases use the concept of labeled property graphs.

*Context* is a widely discussed topic in text mining and knowledge extraction since it is an important factor in determining the correct semantic sense of unstructured text. In [23], Nenkova and McKeown discuss the influence of context on text summarization. Ambiguity is an issue for both common language words and those in scientific context. The challenge in this field is not only to extract such context data, but

TABLE I  
DIFFERENT LAYERS WITHIN THE KNOWLEDGE GRAPHS OF GERMAN JOB ADS.

	Content	Size	Structure
$E_1$	Job Ads	ca. 600,000 per year	unstructured
$E_2$	Classification of Occupations	ca. 18,700	Taxonomy
$E_3$	Sources	1	unstructured
$E_4$	German Skill Taxonomy (AMS)	va. 600	Taxonomy
$E_5$	Tools	ca. 350,000	Taxonomy
$E_6$	Industrial Sectors	ca. 1,814	Taxonomy
$E_7$	Storage Period		unstructured
$E_8$	Regional Data		unstructured

also to be able to store this data for further natural language processing (NLP), like querying and discovery approaches, see for example [4].

In general, for a node  $n \in V$ , the neighborhood  $N(n)$  contains all relevant contextual information. But usually information is best understood using information-triangles. Thus every two nodes  $v, w \in N(n)$  form an implicit triangle  $v, n, w$  and when adding an additional edge  $(v, w)$  this forms a triangle  $K_3$ , see Figure 1 for an illustration. Here, the additional edges that form pseudo-triangles are red.

**Definition II.1** (Pseudo-Triangle). *Let  $G = (E, R)$  be a knowledge graph and let  $n, v, w \in G$  be three nodes in  $G$ . Moreover, let  $n \in E_i$  for some  $i$  and let  $v, w \in N(n)$ . Then  $n, v, w$  form a pseudo-triangle in  $G$ .*

The knowledge graph on German job ads is build upon different corpora of job ads from multiple sources. In this paper, we will focus on a corpus from the German Federal Employment Agency. The corpus that we use to extract skills and tools contains approximately 600,000 job ads per year that were advertised from 2011 to 2021. In Table I we present the different knowledge graph layers.

Thus, combing Figure 1 and Table I, it appears that we are working on a knowledge graph  $G = (V, E)$  with eight different layers, thus  $G = E_1 \cup E_2 \cup \dots \cup E_8 \cup T_1 \cup I_1$  with the given data subsets  $E_1, \dots, E_8$  and the text mining results  $T_1$  and other data integrated in  $I_1$ . We will now discuss how this knowledge graph can be connected with data from different years to build a longitudinal knowledge graph representation.

To test the efficiency of the analysis, we will focus on a very generic question: Given a structured layer which is not constantly changing (e.g. a taxonomy), how do the results on unstructured data (in our case: the job ads) evolve with respect to another structured layer (e.g. another taxonomy, for example tools or skills)? In other words: How can we efficiently retrieve data from a structured layer  $E_s$  ordered by another structured layer  $E_i$  when both are connected over time by different sets of unstructured data? For the sake of simplicity, we will define  $E_s = E_4$  as skills retrieved by text mining and  $E_i$  as  $E_2$  given by the classifications of occupations. Both sets are connected by the job ads.

## III. METHOD

We can extend the knowledge graph with the information for one particular time point  $t$ , in our case for a year:  $G^t = E_1^t \cup E_2^t \cup \dots \cup E_8^t \cup T_1^t \cup I_1^t$ . With this, we can build a generic graph model comprising multiple times  $T = \{t_1, \dots, t_m\}$ :  $G_T = G^{t_1} \cup G^{t_2} \cup C_{t_1, t_2} \cup G^{t_2} \cup G^{t_3} \cup C_{t_2, t_3} \cup \dots \cup G^{t_{m-1}} \cup G^{t_m} \cup C_{t_{m-1}, t_m}$

In this case,  $C_{t_i, t_k}$  comprises all edge relations from  $G^{t_i}$  to  $G^{t_j}$ . This contains relations like `isEqual` if two entities are equivalent or `isSuccessor` if an entity in  $G^{t_j}$  is the successor of an deprecated element in  $G^{t_i}$ .

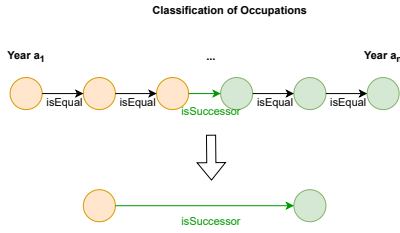


Fig. 2. Merging maximal paths  $P = p_1, \dots, p_m$  containing equal data in multiple years.

Thus, a first step to shrink the volume of the knowledge graph is to merge the multiple existence of elements in multiple years. Thus, we search for maximal paths  $P = p_1, \dots, p_m$  in  $G_T$  where  $p_i \in E_j \forall p_i \in P$ . The edges between  $p_i$  and  $p_{i+1}$  are either `isEqual` or `isSuccessor` edges, see Figure 2. Thus for every edge  $(p_i, p_{i+1}) \in C_{t_j, t_k}$  we can either merge  $p_i$  and  $p_{i+1}$  if they are the same (`isEqual`) or leave the `isSuccessor` edges. In our case we are in particular working on  $E_2$ , the classification of occupations.

This can be done with depth-first search, see Algorithm 1, because we explicitly only use the directed subgraph induced by  $R = G_T [C_{t_1, t_2} \cup C_{t_2, t_3} \cup \dots \cup C_{t_{m-1}, t_m}]$ . The worst-case behavior is in  $O(E(R) + V(R))$  and since every node  $p_i$  has at most  $\Delta(E_2)$  neighbors in  $E_2$  and at most  $N(E_1)$  neighbors in  $E_1$  the time complexity of merging the nodes is  $O(\Delta(E_2) + N(E_1))$ . Thus, the runtime of this step is linear,  $O(n)$  in  $G_T$ . We denote the graph after step 1 with  $G_T^1$ .

**Algorithm 1** STEP-1

**Require:** Knowledge Graph  $G_T$  with layer (tree)  $E_x^t$  and mappings  $C_{t_i, t_{i+1}}$  for all  $t \in T = \{t_1, \dots, t_m\}$

**Ensure:** Shrunked  $G_T^1$

$$M = N(\bigcup_{i=1, \dots, m} E_x^i) \cup E(\bigcup_{i=1, \dots, m-1} C_{t_i, t_{i+1}})$$

$$2: V = N(\bigcup_{i=1, \dots, m} E_x^i)$$

$$P = \emptyset$$

4: **for** every  $v \in E_x^i$  with  $v \in V \forall i \in \{1, \dots, m\}$  **do**

$p = \text{DFS}(M, v)$

6:  $\text{del}(V, y) \forall y \in p$

$P.\text{add}(p)$

8: **end for**

**return**  $G_T^1 = \bigcup_{i=1, \dots, \text{len}(P)} (p_i \in P)$

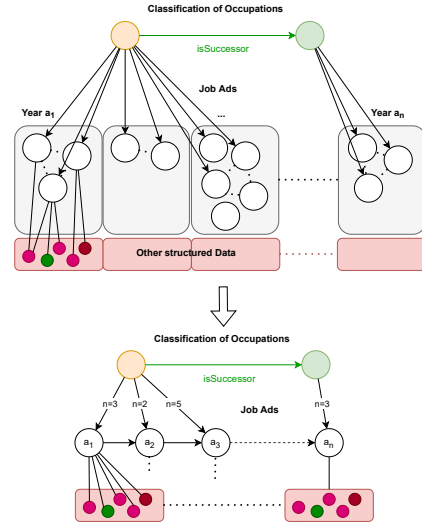


Fig. 3. Merging all job ads for a given year preserving the further links to other structured data.

In general we can make the following observations:

- Since the number of jobs in the classification schema does not increase dramatically, we can assume  $N(E_2^t) \approx N(E_2^{t+1})$ .
- Thus, even though a number of  $e_1$  jobs may either be deprecated or are added as new items, the size of  $E_2^1$  in  $G_T^1$  is  $\max_i N(E_2^i) + e_1 \in O(\max_i N(E_2^i))$ .

In a next step, we can merge all job ads for a given year preserving the further links to other structured data. Thus for every time point  $t$  and every  $v^t \in E_2^t$  we merge all nodes in  $N = \{n^t \mid n^t \in N(v^t) \text{ and } n^t \in E_1^t\}$  to a meta node  $a_t$  and add an edge  $(v^t, a_t)$  with weight  $|N|$ , see Figure 3. These form pseudo-triangles in  $E_1$ .

The runtime of this step is in  $O(tN(E_1)N(E_2))$  and thus is quadratic,  $O(n^2)$  in  $G_T$ , see Algorithm 2. We denote the changed graph after step 2 with  $G_T^2$  and the new shrunked nodes in  $E_1$  with  $E_1^1$ .

Before continuing with a possible third step of shrinking graph structures, we should consider the theoretical results. Given the question, how the description of skills in job ads evolves in job classifications over the years, in the initial graph  $G_T$  we need to consider the following steps:

- Consider the evolution of any classifications  $v$  for all times  $T = \{t_1, \dots, t_m\}$ , runtime  $O(m \max_i V(E_2^i))$ .
- Consider all skills for any job ad in  $N(v^t)$  for all times, runtime  $O(V(E_1^t)V(E_4^t))$ .

Thus, the average runtime is in  $O(n^3)$ . For the graph with shrunked pseudo-triangles this reduces to linear runtime:

- Consider any shrunked path  $p$  in  $E_2^1$ , runtime  $O(\max_i N(E_2^i))$ .
- Consider all skills for all times, runtime  $O(m)$ .

With this third step we can reduce the data complexity, but while in step 1 we do not lose any relevant data, in step 2 we lose the information about specific job ads while preserving

**Algorithm 2** STEP-2

---

**Require:** Knowledge Graph  $G_T$  with an unstructured layer  $E_x^t$  and a target layer  $E_y^t$  containing paths in  $P$  and mappings  $C_{t_i, t_{i+1}}$  for all  $t \in T = \{t_1, \dots, t_m\}$  and

**Ensure:** Shrunked  $G_T^2$

$V = \emptyset$

2:  $E'_x = \emptyset \forall t \in \{1, \dots, m\}$

$P = \square$

4: **for** every  $p \in P$  in  $E_y^t$  **do**

**for** every  $p_i \in P = \{p_1, \dots, p_z\}$  **do**

6: **for** every  $t \in \{1, \dots, m\}$  **do**

$V' = N(p_i) \cap E_x^t$

8: **add**  $v'$  to  $E'_x$

change all edges  $(\hat{v}, u) \forall (\hat{v}, u) \in V'$  and  $u \notin E_x$ ,  
 $u \notin E_y$  to  $(v', u)$

10: **end for**

12: **end for**

**return**  $G_T^2 = (G_T \setminus E_x) \cup E'_x$

---

the information for a complete time set. Thus we can make the following observations:

- Structured data like taxonomies and ontologies can be shrunked without any data loss.
- Shrinking unstructured data in triangles or pseudo-triangles always goes along with data loss of particular data points while accumulated information might be preserved.
- Thus, it highly depends on the initial research question which layers and information can be shrunked to improve the runtime of algorithms.

For the given research question, steps 1-2 are the maximum reduction of the initial graph if considering the change for years.

#### IV. EXPERIMENTAL RESULTS

Our testing environment contains a random graph with  $m$  time points containing several graph layers. First, we have a random tree  $E_2^1$  with 18,700 nodes and two probabilities  $p_p$  and  $p_d$  denoting a rate of a changing predecessor or a deleted node. These probabilities lead to  $m$  copies of  $E_2^1$  and their mapping from one time point to the next as described in the previous section.

Second, we generate  $m$  times 600,000 random nodes in  $E_1^1, \dots, E_1^m$  with equal distributed mappings to  $E_2^1, \dots, E_2^m$  as described in the last section. In addition, these nodes receive random edges to 600 descriptive elements. Thus, our experimental setting is highly related to our real-world environment describes in the second section.

We used 50 instances to evaluate the runtime and performance of the algorithms presented in the last section. In Table II and Figure 4 we show the runtime of the two optimization steps. In general, we can see that both steps in average take 0.6 seconds.

TABLE II  
RUNTIME OF STEP 1 AND STEP 2 IN SECONDS.

	Step 1	Step 2
Min	0.06	0.09
Max	0.79	0.24
Avg	0.45	0.13

TABLE III  
RUNTIME OF RETRIEVAL ALGORITHM ON  $G_T$  AND  $G_T^2$  IN SECONDS.

	$G_T$	$G_T^2$
Min	0.17	0.01
Max	1.01	0.02
Avg	0.58	0.01

In Table III and Figure 5 we show the runtime of the retrieval described in the last section. We can see that the runtime of both optimization steps is nearly the same as one retrieval on the initial graph  $G_T$ . This is not surprising, since the steps are quite similar. The retrieval on the optimized graph  $G_T^2$  is much faster and at latest with the second run of a retrieval algorithm, we see a good improvement of runtime.

#### V. DISCUSSION AND OUTLOOK

This paper investigates the impact of longitudinal data in knowledge graphs. Knowledge graphs play a central role for linking different data. While multiple layers for data from different sources are considered, there is only very limited research on longitudinal data in knowledge graphs. We presented an experimental environment to evaluate one generic retrieval heuristic given different – both structured and unstructured – data layers. The result clearly shows that the graph structures and topology has a great impact on the efficient retrieval of additional data stored. The initial very generic question was: Given a structured layer which is not constantly changing (e.g. a taxonomy), how do the results on unstructured data (in our case: the job ads) evolve with respect to another structured layer (e.g. another taxonomy, for example tools or skills) evolve? In other words: How can we efficiently retrieve data from a structured layer  $E_s$  ordered by another structured layer  $E_i$  when both are connected over time by different sets of unstructured data? We specified three example layers to illustrate our optimization approach on (pseudo-)triangles and to evaluate the efficiency.

In particular, we propose a first draft of a generic model for longitudinal data in multi-layer knowledge graphs. This approach stores copies of the knowledge graph on multiple time points and the mapping between nodes in one and a following time point. Since some optimization can be done without losing data, e.g. step 1, we propose further research on a generic longitudinal data model to use these approaches when building the knowledge graph. Second, we develop an experimental environment to evaluate a generic retrieval algorithms on random graphs inspired by computational social sciences. This example was highly influenced by the boundary conditions given by the real-world problem, a knowledge

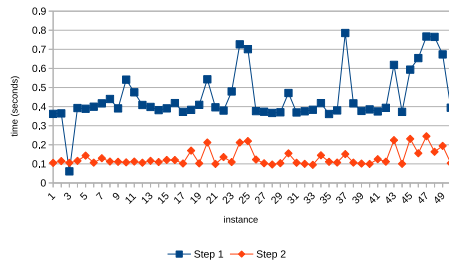


Fig. 4. Resulting runtimes of step 1 and step 2 in seconds.

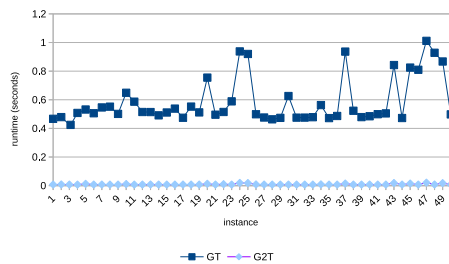


Fig. 5. Resulting runtimes of retrieval algorithm on  $G_T$  and  $G_T^2$  in seconds.

graph generated on German job advertisements comprising data from different sources, both structured and unstructured, on data between 2011 and 2021. The data is linked using text mining and natural language processing methods. In general, we present two different shrinking approaches for structured (step 1) and unstructured (step 2) layers in knowledge graphs based on graph structures like triangles and pseudo-triangles. Here, more research needs to follow. While we have argued that these approaches are generic and can be used for any content, further attention for triangles and pseudo-triangles is needed. They form a crucial factor both for understanding the data context and for efficient retrieval of these data.

The presented approach shows that on the one hand the initial research questions (what are the layers to shrink) and on the other hand the graph structures and topology have a great impact on the structures and efficiency for additional data stored. The experimental results show promising results, but further research is necessary to build a generic, time-efficient representation of longitudinal data in knowledge graphs.

## REFERENCES

- [1] D. Suárez, J. M. Díaz-Puente, and M. Bettoni, “Risks identification and management related to rural innovation projects through social networks analysis: A case study in Spain,” *Land*, vol. 10, no. 6, p. 613, 2021.
- [2] L. M. Berhan, A. L. Adams, W. L. McKether, and R. Kumar, “Board 14: Social networks analysis of African American engineering students at a PWI and an HBCU—a comparative study,” in *2019 ASEE Annual Conference & Exposition*, 2019.
- [3] C. Rollinger, “Amicitia sanctissime colenda,” *Freundschaft und soziale Netzwerke in der Späten Republik*, 2014.
- [4] J. Dörpinghaus and A. Stefan, “Knowledge extraction and applications utilizing context data in knowledge graphs,” in *2019 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2019, pp. 265–272.
- [5] G. Rossetti, S. Citraro, and L. Milli, “Conformity: A path-aware homophily measure for node-attributed networks,” *IEEE Intelligent Systems*, vol. 36, no. 1, pp. 25–34, 2021.
- [6] A. Callahan, V. Polony, J. D. Posada, J. M. Banda, S. Gombar, and N. H. Shah, “Ace: the advanced cohort engine for searching longitudinal patient records,” *Journal of the American Medical Informatics Association*, vol. 28, no. 7, pp. 1468–1479, 2021.
- [7] X. Xu, X. Xu, Y. Sun, X. Liu, X. Li, G. Xie, and F. Wang, “Predictive modeling of clinical events with mutual enhancement between longitudinal patient records and medical knowledge graph,” in *2021 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2021, pp. 777–786.
- [8] S. Auer and H. Herre, “A versioning and evolution framework for RDF knowledge bases,” in *International Andrei Ershov Memorial Conference on Perspectives of System Informatics*. Springer, 2006, pp. 55–69.
- [9] F. Zablith, G. Antoniou, M. d’Aquin, G. Flouris, H. Kondylakis, E. Motta, D. Plexousakis, and M. Sabou, “Ontology evolution: a process-centric survey,” *The knowledge engineering review*, vol. 30, no. 1, pp. 45–75, 2015.
- [10] M. Javed, Y. M. Abgaz, and C. Pahl, “Ontology change management and identification of change patterns,” *Journal on Data Semantics*, vol. 2, no. 2, pp. 119–143, 2013.
- [11] Y. Roussakis, I. Chrysakis, K. Stefanidis, G. Flouris, and Y. Stavarakas, “A flexible framework for understanding the dynamics of evolving RDF datasets,” in *International Semantic Web Conference*. Springer, 2015, pp. 495–512.
- [12] N. Arndt, P. Naumann, N. Radtke, M. Martin, and E. Marx, “Decentralized collaborative knowledge management using git,” *Journal of Web Semantics*, vol. 54, pp. 29–47, 2019.
- [13] S. Cardoso, C. Reynaud-Delaître, M. Da Silveira, Y.-C. Lin, A. Gross, E. Rahm, and C. Pruski, “Evolving semantic annotations through multiple versions of controlled medical terminologies,” *Health and Technology*, vol. 8, no. 5, pp. 361–376, 2018.
- [14] A. Eibeck, A. Chadzyski, M. Q. Lim, K. Aditya, L. Ong, A. Devanand, G. Karmakar, S. Mosbach, R. Lau, I. A. Karimi *et al.*, “A parallel world framework for scenario analysis in knowledge graphs,” *Data-Centric Engineering*, vol. 1, 2020.
- [15] M. Stops, A.-C. Bächmann, R. Glassner, M. Janser, B. Matthes, L.-J. Metzger, C. Müller, and J. Seitz, “Machbarkeitsstudie kompetenz-kompass: Teilprojekt 2: Beobachtung von kompetenzanforderungen in Stellenangeboten.” [Online]. Available: <https://www.bmas.de/DE/Service/Publikationen/Forschungsberichte/fb-553-machbarkeitsstudie-kompetenz-kompass.html>
- [16] Bertelsmann Stiftung and Burning Glass Technologies, “Digitalization in the German labor market: Analyzing demand for digital skills in job vacancies.”
- [17] S. Köhne-Finster, I. Leppelmeier, R. Helmrich, D. Deden, A. Geduldig, B. Güntürk-Kuhl, P. Martin, C. Neuber-Pohl, M. Schandock, R. S. Schreiber, and M. Tiemann, *Berufsbildung 4.0 - Fachkräftequalifikationen und Kompetenzen für die digitalisierte Arbeit von morgen: Säule 3: Monitoring- und Projektionssystem zu Qualifizierungsnotwendigkeiten für die Berufsbildung 4.0*, 1st ed., ser. Wissenschaftliche Diskussionspapiere. Leverkusen: Verlag Barbara Budrich, 2020, vol. Heft 214.
- [18] A. Bhola, K. Halder, A. Prasad, and M.-Y. Kan, “Retrieving skills from job descriptions: A language model based extreme multi-label classification framework,” in *Proceedings of the 28th International Conference on Computational Linguistics*, D. Scott, N. Bel, and C. Zong, Eds. Stroudsburg, PA, USA: International Committee on Computational Linguistics, 2020, pp. 5832–5842.
- [19] D. Fensel, U. Şimşek, K. Angele, E. Huaman, E. Kärle, O. Panasiuk, I. Toma, J. Umbrich, and A. Wahler, *Introduction: What Is a Knowledge Graph?* Cham: Springer International Publishing, 2020, pp. 1–10. [Online]. Available: [https://doi.org/10.1007/978-3-030-37439-6\\_1](https://doi.org/10.1007/978-3-030-37439-6_1)
- [20] L. Ehrlinger and W. Wöb, “Towards a definition of knowledge graphs,” *SEMANTICS (Posters, Demos, SuCCeSS)*, vol. , no. 48, 2016.
- [21] M. A. Rodriguez and P. Neubauer, “The graph traversal pattern,” in *Graph data management: Techniques and applications*. IGI Global, 2012, pp. 29–46.
- [22] —, “Constructions from dots and lines,” *Bulletin of the American Society for Information Science and Technology*, vol. 36, no. 6, pp. 35–41, 2010.
- [23] C. C. Aggarwal and C. Zhai, “An introduction to text mining,” in *Mining text data*. Springer, 2012, pp. 1–10.