# Extending PLASMA for Industrial Semantic Modeling

Alexander Paulus[1,*,†], Andreas Burgdorf[1,†], Tobias Meisen[1] and André Pomp[1,†]

[1]*Institute for Technologies and Management of Digital Transformation, University of Wuppertal, Wuppertal, Germany*

Abstract

Automated semantic modeling does not produce rich semantic models that include contextual information often required for data interpretation. Additionally, the necessary refinement of semantic models by human modelers is still a challenging task, especially for domain experts such as engineers. This paper shows how PLASMA, a semantic modeling system, integrates automation approaches for semantic modeling. Using two existing approaches, PLASMA is able to assist domain experts during semantic labeling and refinement.

**Keywords**

data space, semantic modeling, data management, recommendation engine

## 1. Introduction

In the ever-evolving landscape of technology and manufacturing, Industry 4.0 has emerged as a revolutionary concept that is reshaping the way industries operate and interact with the digital world. Central to the successful realization of Industry 4.0 is the effective management, integration, and utilization of vast and heterogeneous industrial datasets generated by interconnected devices, processes, and entities. Ontology-based data platforms [1, 2, 3, 4] or, more recently, data spaces [5, 6] emerge as a strategic response to these challenges, providing structured virtual environments that facilitate the secure, standardized storage, exchange, and processing of data.

In order to enable the interoperability of heterogeneous data sources, semantic modeling, also referred to as information modeling, plays an important role for data spaces [7]. By using standardized vocabularies and ontologies, semantic modeling imbues data with context, meaning, and relationships, enabling machines and systems to comprehend and interpret data beyond its raw form. In this way, data becomes more interpretable, allowing for enhanced interoperability, context-aware decision-making, and more sophisticated analytics.

In the industrial context, however, the creation of semantic models is still a major challenge for enterprises. The many automated approaches for creating semantic models are still very unreliable in industrial contexts [8]. Schema-based approaches, such as [9, 10, 11, 12, 13],

*SemIIM'23: 2nd International Workshop on Semantic Industrial Information Modelling, 7th November 2023, Athens, Greece, co-located with 22nd International Semantic Web Conference (ISWC 2023)*

[*] Corresponding author.

[†] These authors contributed equally.

paulus@uni-wuppertal.de (A. Paulus); burgdorf@uni-wuppertal.de (A. Burgdorf); meisen@uni-wuppertal.de (T. Meisen); pomp@uni-wuppertal.de (A. Pomp)

0000-0002-0774-1528 (A. Paulus); 0000-0001-7776-8497 (A. Burgdorf); 0000-0002-1969-559X (T. Meisen); 0000-0003-0111-1813 (A. Pomp)

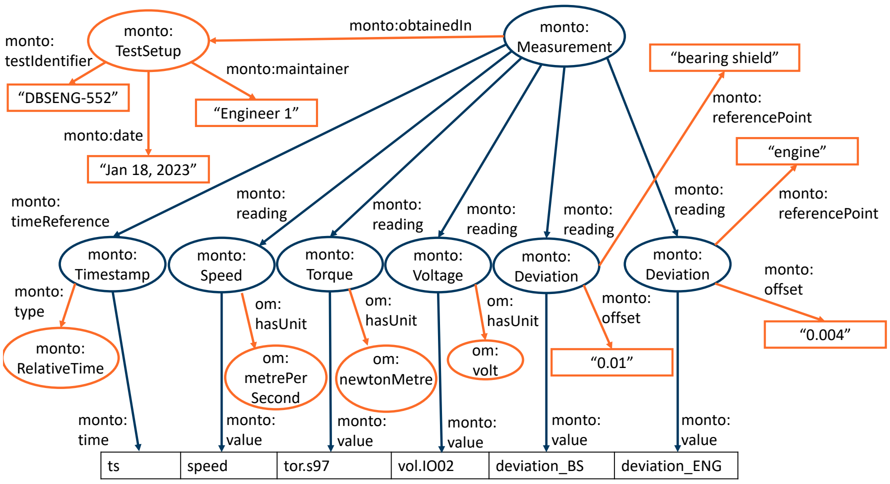


**Figure 1:** Enriched version of a minimal semantic model derived from a collaboration with Siemens AG. Highlighted elements are added as part of the rich semantic model.

usually fail in real-world use cases due to ambiguous, strongly simplified, strongly encoded or obfuscated schemas. Data-driven approaches, on the other hand, require data corpora consisting of data and corresponding historical semantic models to be trained and used reliably. Examples include [14, 15, 16, 17, 18, 19]. However, the required data corpora are usually enterprise- and/or domain-dependent or simply do not exist. In addition, approaches for automatically generating semantic models, such as [20, 21, 22], produce only minimal spanning trees. Figure 1 shows a semantic model for a dataset obtained from a machine test setup. The blue elements represent the minimal spanning tree. Industrial applications, however, often require rich semantic models that include additional information such as units of measurement for values or configuration parameters such as reference points or offsets, indicated by the highlighted elements in Figure 1.

This means that whenever an automated algorithm makes a mistake or information is missing from the semantic model, a human must intervene (*semantic refinement*). At the same time, especially for the data-driven approaches, enough semantic models must first be built by humans to have sufficient training data. Considering industrial environments, the manual creation and refinement of the generated models is typically done by domain experts, i.e., users who know the data very well but often have limited knowledge of semantic technologies.

Thus, different semantic modeling platforms have been developed in recent years, i.e., tools that provide a user interface for checking, validating, or correcting the results of an automated semantic modeling algorithm as well as for manually creating semantic models from scratch. Examples include KARMA [3], the RML Editor [23], MantisTable [24, 25], SAND [26]. However, these tools do not actively support the modelers during model creation but rely on the proficiency of the modeler to fulfill the task. Thus, correcting or manually constructing semantic models still remains a challenge for domain experts. Other tools, like [27, 28] support domain experts in dealing with or creating ontologies but do not include semantic refinement.

In this paper, we detail our *PLatform for Auxiliary Semantic Modeling Approaches* (PLASMA), which is specifically designed for the creation of semantic models in industrial data spaces and

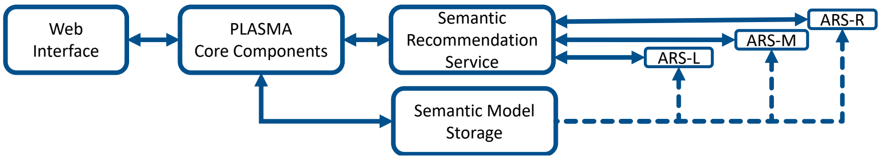


**Figure 2:** Integration of Auxiliary Recommendation Services (ARS) to PLASMA

scenarios. PLASMA[1] has been developed over the last years and has already been presented in various publications [29, 30, 31]. It has been applied to different use cases, for example, with the Siemens AG [30] and in a semantic data platform for smart city data [5].

Compared to our previous publications, this paper focuses especially on the system components that allow PLASMA to be extended with different automation approaches. Based on the underlying architecture, we show how two approaches that support domain experts during the semantic modeling creation are attached to the platform. The first case demonstrates how we automatically create initial semantic mappings based on textual data documentations (Section 3.1). This type of descriptive text for data is the most common information found in industrial contexts, and can help experts and now also automation approaches to build semantic models. The second case is a continuously learning recommendation engine, that explicitly supports modelers during the model creation (Section 3.2).

## 2. The PLASMA System

PLASMA is a semantic modeling system that aims to simplify the annotation of datasets, especially for hierarchical data formats such as JSON or XML. It covers the entire process of analyzing a dataset, semantic model creation using automated approaches as well as refining it within a user interface. During semantic refinement, the modeler can improve the semantic model by adding new, additional information, removing errors or exchanging individual parts. The interface features convenient and well-known drag & drop interaction patterns in a graph-based modeling environment [32] to ensure usability by modelers who are not necessary familiar with semantic modeling tools, but operate other applications such as domain specific modeling and diagram editors [30]. One of the key features of PLASMA is the integration of (external) services that provide the automation of semantic labeling and modeling (semantic relation inference) and may also provide recommendations during the semantic refinement, guiding the user towards more detailed models [30, 31]. Furthermore, PLASMA maintains a local knowledge graph comprised of all semantic models created within the platform.

The core idea of PLASMA, as indicated by the name, is the integration of auxiliary semantic modeling approaches, which in general are existing automation approaches that PLASMA wants to connect in order to assist the modeler. While for other platforms such as KARMA, MantisTable and SAND, automation approaches are added directly to the application code, e.g. via a plugin system [26], PLASMA follows a different strategy by integrating those approaches as independent modules referred to as *Auxiliary Recommendation Services* (ARS).

[1] Available as open source on https://github.com/tmdt-buw/plasma

Within the architecture of PLASMA exists a component that handles requests for recommendations on behalf of a modeler, referred to as the semantic recommendation service (SRS). The SRS manages the communication to all ARS connected to PLASMA, serving as a proxy for recommendation requests, ensuring that data formats match and that proposed recommendations are valid (see Figure 2). The service itself does not contain any recommendation logic, but maintains an index of which ARS are available / online and how to reach them. Communication between the SRS and each ARS is specified by a shared and generic interface, allowing new services to be created and plugged in, even during runtime. This interface specifies how data inside PLASMA is represented, e.g., the *combined model* as the central data structure to contain both the data schema from the input file as well as the semantic model being built (cf. [29]).

ARS provide their generated suggestions as independent *modifications*, a set of changes to a combined model, which can subsequently be presented to the modeler in the web interface. Modifications can contain *(i)* a single triple to add, *(ii)* a semantic mapping, i.e., data types for specific data attributes, as well as *(iii)* a complete semantic model provided by an automated semantic modeling approach. Technically, every change to a semantic model is encoded using modifications. However, modifications obtained through the SRS are optional and may be accepted (applied to the model) or rejected (discarded) by the modeler.

## 3. Integration of Supportive ARS

Each ARS in PLASMA encapsulates an algorithm to perform a specific task in either the labeling, modeling or refinement phase of the semantic model creation process [29]. The different ARS are referred to as *ARS for Labeling (ARS-L)*, *ARS for Modeling (ARS-M)* and *ARS for Refinement (ARS-R)*, respectively. An ARS receives the current state of the combined model and optionally some meta information and will return a set of modifications for the user to chose from. The services adopt the standardized API defined by the SRS to be quickly pluggable or exchangeable. It is even possible to support multiple ARS per type simultaneously by gathering their proposed modifications and sorting them using a provided confidence score.

ARS are modular services, contained in their own environment, e.g., using a Docker container. This poses as little limitations as possible to developers on what technology stack to use when implementing an ARS. As long as an ARS communicates via the defined API provided by the platform and SRS, it can be connected. If the approach is an already implemented and running system with a predefined API, the ARS can also function as a proxy or bridge to it. This setup can make a re-implementation or modification of the existing external service unnecessary, although converting between the SRS data model and the external systems API may be challenging.

In cases where additional data is required by the ARS, access to other services may be granted to an ARS to obtain the information. Registered ARS can access all ontologies added to PLASMA as well as the created semantic models, e.g., for training machine learning models. It is also possible for an ARS to query any outside source, such as knowledge graphs, via SPARQL or REST. In the following, two exemplary approaches that have been realized as ARS in PLASMA are presented.

### 3.1. Semantic Labeling with Textual Data Documentations

DocSemMap [33, 34, 35], an advanced semantic labeling methodology, uses attribute names from input datasets and corresponding user-provided textual documentations to facilitate initial semantic labeling (ARS-L). The core of DocSemMap lies in the strategic use of external knowledge sources to optimize semantic alignment, similar to the cognitive processes employed by human experts. By incorporating a diverse set of background knowledge, the approach aims to improve the accuracy of automated semantic labeling in three phases. In the *Initialization Phase*, DocSemMap preprocesses the textual documentation and the attribute names to extract linguistic features and embeddings. The *Candidate Selection Phase* is dedicated to generating candidate concepts for each attribute. It consists of eight sequential steps, mainly based on embeddings, history and linguistic rules and a sequence to sequence model which predicts target concepts of different attributes in combination to preserve the dataset context.

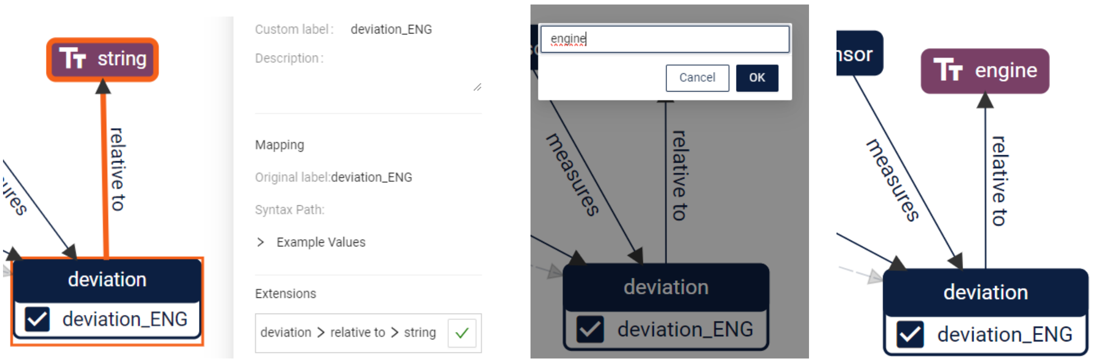
Finally, the *Decision Making Phase* involves the selection of the best concept for each attribute through a decision-making process. This process weighs candidates based on historical knowledge, context matching, and other criteria. Moreover, the integration of additional techniques, such as the Seq2Seq model, further bolsters the accuracy and efficiency of the semantic labeling process.

Being realized as an ARS, DocSemMap seamlessly interfaces with the relevant PLASMA components to procure essential data. Frequently, the ARS retrieves all ontologies indexed in PLASMA and updates the pre-processing linguistic data for classes and properties. For each request posed to this ARS, available metadata, such as a textual description, is obtained for the dataset. Based on the linguistic concept data and the textual description, the candidate selection steps are executed on each attribute of the provided dataset. A modification is build using the class with the highest confidence score estimated for each attribute. DocSemMap is capable of capturing and linking the context of different documentation. This provides added value especially for use in companies that do not necessarily work on complete ontologies. For this reason, PLASMA provides a favorable application scenario for DocSemMap.

### 3.2. Refinement Recommendation Generation

Essential information missing in a (rich) semantic model decreases the interpretability of the data contained in the dataset. While in general domain experts know which information is relevant, they tend to omit this information explicitly while assuming that it is commonly known. Thus, the first challenge of semantic refinement are the identification of missing knowledge in the semantic model, i.e., which information could be added to improve the model. Second, once missing information has been identified, a suitable way to express it in a formalized language such as RDF has to be found. If the same information is modeled differently by different modelers, the discrepancies reduce the consistency of the resulting semantic models and in turn increase the workload of writing queries that retrieve all similar information.

To improve consistency and at the same time alert the modeler about potentially missing information, recommendations can be used. A recommendation in PLASMA is a (visual) hint to the modeler that an added triple could improve the quality of the semantic model. At the same time, through the recommendation of a specific triple, not only is the missing information



Custom label : deviation_ENG

Description :

Mapping

Original label:deviation_ENG

Syntax Path:

> Example Values

Extensions

deviation > relative to > string ✓

engine

(a) Visualization (b) Value (c) Literal

**Figure 3:** Recommendations in the PLASMA modeling interface

indicated, but at the same time, a matching formalization is proposed.

Providing recommendations during semantic refinement has been demonstrated through the generation of Linked Model Extensions [31, 36] as well as single concepts for specific attributes [37]. The presented approaches utilize machine learning models to identify suitable triples the user might want to add. A combination of both approaches has been added to PLASMA as an ARS-R, generating recommendations that are visually presented inside the modeling interface. Recommendations may then be accepted and modified, e.g., values inserted into literals, as shown in Figure 3. The ARS-R runs on a Python environment and its original API has been adjusted to match the PLASMA data model, allowing a direct communication with the SRS. The ARS-R frequently queries all semantic models present in PLASMA and re-trains its internal recommendation engine based on those models. This ensures that newly created semantic models are included to improve the recommendations over time and increase consistency. Recommendations are then generated based on observed patterns inside the training data.

## 4. Conclusion and Future Work

In this paper, we showed two examples of existing approaches for semantic modeling automation that have been integrated into the semantic modeling platform PLASMA to improve the semantic modeling process for domain experts. Two cases of recommendation engines, one for semantic labeling and one for semantic refinement, were described and their integration into PLASMA as an auxiliary service was detailed. Both auxiliary services access other PLASMA components to obtain the data necessary to operate even in a closed environment.

In the future, we aim to connect multiple other services in order to further advance the assistive capabilities of PLASMA during semantic model creation. Alongside the addition of those services, a feedback loop about the modeler decisions (accept/reject) back to the recommendation services may improve recommendations through active learning and filtering in the interactive semantic model creation process. In addition, a service auto-discovery as well as a service self description accessible through the modeling interface are planned.

## References

[1] E. Kharlamov, T. Mailis, G. Mehdi, C. Neuenstadt, Ö. Özçep, M. Roshchin, N. Solomakhina, A. Soylu, C. Svingos, S. Brandt, et al., Semantic access to streaming and static data at siemens, Journal of Web Semantics 44 (2017) 54–74.

[2] E. Kharlamov, D. Hovland, M. G. Skjæveland, D. Bilidas, E. Jiménez-Ruiz, G. Xiao, A. Soylu, D. Lanti, M. Rezk, D. Zheleznyakov, et al., Ontology based data access in statoil, Journal of Web Semantics 44 (2017) 3–36.

[3] C. A. Knoblock, P. Szekely, J. L. Ambite, A. Goel, S. Gupta, K. Lerman, M. Muslea, M. Taheriyan, P. Mallick, Semi-automatically Mapping Structured Sources into the Semantic Web, in: E. Simperl (Ed.), The semantic web, volume 7295, Springer, 2012.

[4] A. Pomp, A. Paulus, S. Jeschke, T. Meisen, Eskape: Information platform for enabling semantic data processing, in: International Conference on Enterprise Information Systems, volume 2, SCITEPRESS, 2017, pp. 644–655.

[5] A. Pomp, A. Paulus, A. Burgdorf, T. Meisen, A semantic data marketplace for easy data sharing within a smart city, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 4774–4778.

[6] A. Donald, A. Galanopoulos, E. Curry, E. Muñoz, I. Ullah, M. Waskow, M. Dabrowski, M. Kalra, Towards a semantic approach for linked dataspace, model and data cards, in: Companion Proceedings of the ACM Web Conference 2023, 2023, pp. 1468–1473.

[7] J. Theissen-Lipp, M. Kocher, C. Lange, S. Decker, A. Paulus, A. Pomp, E. Curry, Semantics in dataspaces: Origin and future directions, in: Companion Proceedings of the ACM Web Conference 2023, 2023, pp. 1504–1507.

[8] A. Paulus, A. Burgdorf, A. Pomp, T. Meisen, Recent advances and future challenges of semantic modeling, in: 2021 IEEE 15th International Conference on Semantic Computing (ICSC), IEEE, 2021, pp. 70–75.

[9] P. Papapanagiotou, P. Katsiouli, et al., RONTO: Relational to Ontology Schema Matching, AIS Sigsemis Bulletin 3 (2006) 32–36.

[10] Z. Syed, T. Finin, et al., Exploiting a Web of Semantic Data for Interpreting Tables, in: Proceedings of the Second Web Science Conference, 2010.

[11] J. Wang, H. Wang, et al., Understanding Tables on the Web, in: Conceptual Modeling, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 141–155.

[12] C. Pinkel, C. Binnig, et al., IncMap: Pay as You Go Matching of Relational Schemata to OWL Ontologies, in: OM, 2013, pp. 37–48.

[13] S. Polfliet, R. Ichise, Automated Mapping Generation for Converting Databases into Linked Data, in: Proceedings of the 2010 International Conference on Posters & Demonstrations Track - Volume 658, ISWC-PD'10, CEUR-WS.org, Aachen, Germany, Germany, 2010, pp. 173–176.

[14] A. Goel, C. Knoblock, K. Lerman, Exploiting Structure within Data for Accurate Labeling Using Conditional Random Fields, in: Proceedings of the 14th International Conference on Artificial Intelligence (ICAI), 2012.

[15] S. K. Ramnandan, A. Mittal, et al., Assigning Semantic Labels to Data Sources, in: The Semantic Web. Latest Advances and New Domains, Springer International Publishing, Cham, 2015, pp. 403–417.

[16] M. Pham, S. Alse, et al., Semantic Labeling: A Domain-Independent Approach, in: The Semantic Web – ISWC 2016, Springer International Publishing, Cham, 2016, pp. 446–462.

[17] N. Rümmele, Y. Tyshetskiy, A. Collins, Evaluating Approaches for Supervised Semantic Labeling, CoRR abs/1801.09788 (2018).

[18] J. Chen, E. Jimenez-Ruiz, et al., Learning Semantic Annotations for Tabular Data, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, California, 8/10/2019 - 8/16/2019, pp. 2088–2094.

[19] M. Hulsebos, K. Hu, et al., Sherlock, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD '19, ACM Press, New York, New York, USA, 2019, pp. 1500–1508.

[20] M. Taheriyan, C. A. Knoblock, et al., Learning the Semantics of Structured Data Sources, Journal of Web Semantics 37-38 (2016) 152–169.

[21] B. Vu, C. Knoblock, J. Pujara, Learning Semantic Models of Data Sources Using Probabilistic Graphical Models, in: The World Wide Web Conference, WWW '19, ACM, New York, NY, USA, 2019, pp. 1944–1953.

[22] G. Futia, A. Vetrò, J. C. De Martin, Semi: A semantic modeling machine to build knowledge graphs with graph neural networks, SoftwareX 12 (2020) 100516. doi:https://doi.org/10.1016/j.softx.2020.100516.

[23] K. Sengupta, P. Haase, M. Schmidt, P. Hitzler, Editing R2RML Mappings Made Easy, in: Proceedings of the 12th International Semantic Web Conference (Posters & Demonstrations Track) - Volume 1035, ISWC-PD '13, CEUR-WS.org, Aachen, DEU, 2013, pp. 101–104.

[24] M. Cremaschi, A. Rula, A. Siano, F. de Paoli, MantisTable: A Tool for Creating Semantic Annotations on Tabular Data, in: The semantic web, volume 11762 of *LNCS sublibrary. SL 3, Information systems and applications, incl. Internet/Web, and HCI*, Springer, Cham, 2019.

[25] R. Avogadro, M. Cremaschi, MantisTable V: A novel and efficient approach to Semantic Table Interpretation, in: SemTab@ISWC, 2021.

[26] B. Vu, C. A. Knoblock, SAND : A Tool for Creating Semantic Descriptions of Tabular Sources, in: P. Groth (Ed.), The semantic web, volume 13384 of *Lecture Notes in Computer Science*, Springer, Cham, 2022, pp. 63–67.

[27] A. Soylu, E. Kharlamov, D. Zheleznyakov, E. Jimenez-Ruiz, M. Giese, M. G. Skjæveland, D. Hovland, R. Schlatte, S. Brandt, H. Lie, et al., Optiquevqs: A visual query system over ontologies for industry, Semantic Web 9 (2018) 627–660.

[28] J. Lipp, L. Gleim, M. Cochez, I. Dimitriadis, H. Ali, D. H. Alvarez, C. Lange, S. Decker, Towards easy vocabulary drafts with neologism 2.0, in: European Semantic Web Conference, Springer, 2021, pp. 21–26.

[29] A. Paulus, A. Burgdorf, L. Puleikis, T. Langer, A. Pomp, T. Meisen, Plasma: Platform for auxiliary semantic modeling approaches., in: ICEIS (2), 2021, pp. 403–412.

[30] A. Paulus, A. Burgdorf, T. Langer, A. Pomp, T. Meisen, S. Pol, Plasma: A semantic modeling tool for domain experts, in: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, 2022, pp. 4946–4950.

[31] A. Paulus, A. Pomp, T. Meisen, The plasma framework: Laying the path to domain-specific semantics in dataspaces, in: Companion Proceedings of the ACM Web Conference 2023, 2023, pp. 1474–1479.

[32] A. Pomp, A. Paulus, D. Klischies, C. Schwier, T. Meisen, A Web-based UI to Enable Semantic Modeling for Everyone, Procedia Computer Science 137 (2018) 249–254.

[33] A. Burgdorf, A. Pomp, T. Meisen, Towards NLP-supported Semantic Data Management, ???? URL: http://arxiv.org/pdf/2005.06916v1.

[34] A. Burgdorf, A. Paulus, A. Pomp, T. Meisen, Docsemmap: Leveraging textual data documentations for mapping structured data sets into knowledge graphs, in: 2022 IEEE 16th International Conference on Semantic Computing (ICSC), IEEE, 2022, pp. 209–216.

[35] A. Burgdorf, A. Paulus, A. Pomp, T. Meisen, Docsemmap 2.0: Semantic labeling based on textual data documentations using seq2seq context learner, in: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, 2022.

[36] A. Paulus, A. Burgdorf, A. Pomp, T. Meisen, Filtering recommender system for semantic model refinement, in: 2023 IEEE 17th International Conference on Semantic Computing (ICSC), IEEE, 2023, pp. 183–190.

[37] A. Pomp, V. Kraus, L. Poth, T. Meisen, Semantic Concept Recommendation for Continuously Evolving Knowledge Graphs, in: J. Filipe, M. Śmiałek, A. Brodsky, S. Hammoudi (Eds.), Enterprise Information Systems, volume 378 of *Lecture Notes in Business Information Processing*, Springer, Cham, 2020, pp. 361–385.