Integration of Scholarly Communication Metadata using Knowledge Graphs

Afshin Sadeghi[1], Christoph Lange[1 2], Maria-Esther Vidal[2], and Sören Auer[3 4]

[1] Institute for Applied Computer Science, University of Bonn, Germany
[2] Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS), Germany
{sadeghi,langec,vidal}@cs.uni-bonn.de
[3] Computer Science, Leibniz University of Hannover, Germany
[4] TIB Leibniz Information Center for Science and Technology, Hannover, Germany
soeren.auer@tib.eu

Abstract. Important questions about the scientific community, e.g., what authors are the experts in a certain field, or are actively engaged in international collaborations, can be answered using publicly available datasets. However, data required to answer such questions is often scattered over multiple isolated datasets. Recently, the Knowledge Graph (KG) concept has been identified as a means for interweaving heterogeneous datasets and enhancing answer completeness and soundness. We present a pipeline for creating high quality knowledge graphs that comprise data collected from multiple isolated structured datasets. As proof of concept, we illustrate the different steps in the construction of a knowledge graph in the domain of scholarly communication metadata (SCM-KG). Particularly, we demonstrate the benefits of exploiting semantic web technology to reconcile data about authors, papers, and conferences. We conducted an experimental study on an SCM-KG that merges scientific research metadata from the DBLP bibliographic source and the Microsoft Academic Graph. The observed results provide evidence that queries are processed more effectively on top of the SCM-KG than over the isolated datasets, while execution time is not negatively affected.

1 Introduction

Yearly thousands of research articles are published in journals and conference proceedings around the world. To conduct research and take advantage of the latest knowledge in an area, it is imperative for researchers to follow the work of other scientists. Therefore, metadata describing articles, authors, journals, calls and conferences can enable effective and efficient research communication. A data source can be rich in one aspect and insubstantial in other aspects. For example, the *DBLP* computer science bibliography database gathers ample information about publications in specific conferences but has sparse data about their keywords and no data about citations. Furthermore it lacks metadata on publications in different fields of research. The *Microsoft Academic Graph* fills these gaps but is less complete in every scientific field. We claim that collecting research communication metadata from heterogeneous sources and integrating them in a queryable environment not only leads to a more robust knowledge base but also, thanks to increased completeness, enables more effective data analysis.

From the 2012 blog post in which Google used the term 'Knowledge Graph' for the first time [11], knowledge graphs have been an important subject of research, but still there does not exist a single, widely accepted definition of this term. Many authors refer to 'knowledge graphs' as a structured base of human knowledge in the form of a graph, with an emphasis on comprehensiveness and large scale [9,4]. Examples of famous knowledge graphs include *DBpedia*, *YAGO*, and *Freebase*.

In this work, we created an integrated graph of scientific knowledge from DBLP and the Microsoft Academic Graph and describe the challenges in matching, linking and integrating the datasets and our approach to addressing these challenges as a methodology that can be reused to build similar knowledge graphs. We present the application of semantic structure based similarity measures in instance matching and show that traditional linking frameworks such as *Silk* are capable of linking with high relative precision and recall, when they consider data semantics during the linking process.

The remainder of this paper is as follows: Section 2 describes DBLP and the Microsoft Academic Graph, and motivates the need for knowledge graph integration with concrete examples. Section 3 defines our concept of a knowledge graph for scholarly communication metadata (SCM-KG). Section 4 shows how the integrated knowledge graph is built. Section 5 reviews related work, and Section 6 reports on the evaluation of our approach. Finally, section 7 concludes and provides an outlook to future work.

2 Motivating Example

In this example, we target the problem of data accuracy in DBLP and the Microsoft Academic Graph and show how creating a high-quality integrated knowledge graph from these heterogeneous sources helps to solve ambiguity problems.

DBLP[5] is an up-to-date dataset of publications, authors and conferences in the area of computer science. Information about an article includes the title and the year of publication; information about authors includes their most recent affiliation. DBLP rarely includes keywords of its publications and misses valuable information such as abstracts and information on the citation of articles. DBLP can be browsed online and is available for download as an XML dump; third parties also provide RDF dumps.

The **Microsoft Academic Graph** (henceforth called "MAG")[6] covers publications, authors, and conferences in all scientific areas. It is neither updated as regularly nor as complete as DBLP in the computer science area, but it includes abstracts, keywords, and citation relations. Further, for each publication, it covers the author affiliations at the time of publication. MAG is available as a relational database dump in CSV format.

In the latest DBLP version of April 2017, there are four authors named "Christoph Lange", indexed 0001 to 0004. When one of these four persons publishes a new article, the maintainers of DBLP face the challenge of linking the article to the right person using his affiliation but DBLP keeps only the current affiliation. By matching authors' publications and recent affiliations, we can link DBLP authors to MAG authors. Now, an old, unindexed publication by a researcher named "Christoph Lange" can be matched

[5] `http://dblp.l3s.de/dblp++.php`, accessed on 10 April 2017

[6] `https://academicgraphwe.blob.core.windows.net/graph-2016-02-05` accessed on 10 April 2017

against the author and affiliation information in the unified knowledge graph and linked to the correct person entity – at least when no two different persons published at the same institution at different times. This example shows how combining multiple available data sources can solve an ambiguity problem.

3 SCM Knowledge Graph Concept

In this section, we first define basic principles of knowledge graphs and then our notion of a scholarly communication metadata knowledge graph (SCM-KG).

Identification A key prerequisite for a knowledge graph is to uniquely identify things. All entities of interest should be uniquely identified by Universal / International Resource Identifiers (URI/IRI).

Representation We need to ensure that information about these things can be easily understood by different parties. The *W3C Resource Description Framework* (RDF) has meanwhile evolved into the *lingua franca* of data integration.

Integration For data exchange in a digitized domain to scale, organizations and involved people need to develop a common understanding of the data. Vocabularies define common concepts (classes) and their attributes (properties) and assign unique identifiers to them.

Coherence Scholarly meta-data frameworks use a large number of data models and data exchange and serialization techniques including relational databases, XML, and JSON. Meanwhile transformation techniques for the RDF data model have been standardized by the W3C.

Access Depending on the usage scenario, there are different requirements and possibilities for data access, such as push vs. pull or individual vs. bulk access. To support these scenarios, knowledge graphs should provide various methods to access data.

Coverage Knowledge graphs should cover a sizeable, extensible area of knowledge stretching across several domains. Even though the field of scholarly publications is well defined with high-quality reference datasets, their incompleteness justifies the need for an integrated knowledge graph.

Knowledge Graph Based on the principles introduced previously, a knowledge graph is a fabric of concept, class, property, relationships, and entity descriptions. It uses a knowledge representation formalism, typically RDF, RDF Schema, or OWL. It aims at a holistic representation of knowledge covering multiple sources, multiple domains, and different granularity. It can be *open* (e.g., DBpedia), *private* or *closed*. It includes schema data as well as instance data. Publishing our knowledge graph as LOD allows clients to easily consume it directly or by performing queries over an SPARQL endpoint. Additionally, it can be integrated with other data quite easily. Third parties who want to perform further integration would not have to install our pipeline but could also follow alternative approaches.

Applying these principles to the domain of scholarly communication requires:

- **Identification** is provided by a scholarly schema such as ORCID for authors, DOI for articles and books, or ISBN for books.
- Besides RDF-based representations, the XML schema of DBLP serves as a well-known **representation**.

– Common RDF-based vocabularies for knowledge **integration** include those from the SPAR family of ontologies[7].
– Regarding **coherence**, it is necessary to map data from a variety of sources, e.g., DBLP from XML and MAG from CSV.

There is not currently an integrated knowledge graph that satisfies all criteria of the definition given above, but besides DBLP and MAG and non-free data sources such as those of Google Scholar or ResearchGate, there are other open datasets, and their schemas could serve as sources for a more comprehensive integration. For example, *Scholarly Data* is a well-engineered RDF dataset on papers of Semantic Web conferences[8] and *OpenCitations*[9] is an open repository of scholarly citation data.

4 Building a Knowledge Graph

In this section, we step by step explore our general approach to build high quality knowledge graphs. We use the scientific communication domain as an example, although the methodology is domain-independent. Figure 1 shows the architecture of the overall system, called SCM-KG-PIP (SCM-KG creation Pipeline).

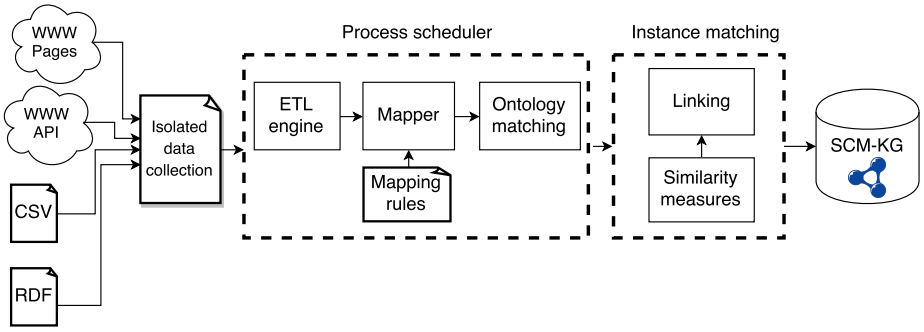

Fig. 1: The pipeline to create a knowledge graph from heterogeneous resources.

As input of SCM-KG-PIP, heterogeneous data arrives in different formats, such as CSV, RDF, web pages, or data returned by calling Web APIs. Our approach results in a high-quality, queryable semantic knowledge graph, using a unified schema.

The following subsections present the components of the SCM-KG-PIP architecture in detail and describe how we applied them to scholarly communication metadata to build a knowledge graph for that domain. The pipeline steps in order of execution are: (1) data acquisition, (2) ontology engineering, (3) mapping data to the ontology, (4) calculating similarity and instance matching, and (5) producing the KG and querying it. Steps (1)–(2) are carried out manually, steps (3)–(5) are executed automatically.

[7] http://www.sparontologies.net/
[8] http://www.scholarlydata.org
[9] http://opencitations.net/

4.1 Data Acquisition

Data available in heterogeneous sources can be obtained in different ways. When they are available as structured dumps, e.g., as CSV, SQL or RDF, their structure may not match the target ontology. For example, the DBLP and OpenAIRE[10] datasets are available as RDF, and MAG is available as CSV. Data from Web APIs, another source of structured data, can be collected by gradual querying. Usually, the number of API calls in a specific time window is limited; therefore, throttling has to be applied to requests.

When structured data is not provided through open interfaces, one may be forced to resort to scraping data from web pages. Currently, Google Scholar and ResearchGate, two highly relevant sources of data about authors' current affiliations and recent publications, do not provide ways to access metadata other than by web scraping. Web scraping requires finding relevant pages, parsing them, and extracting the desired metadata from their content. In the concrete case of ResearchGate, we experimented with such a parser for author and publication metadata, implemented using the *Scrapy* Python framework[11], but found it hard to maintain, as, after just half a year, the content structure of the ResearchGate pages had changed significantly.

4.2 Ontology Engineering

Different structured data sources may use different schemas, e.g., DBLP and MAG model the same concepts (e.g., affiliation) differently. Creating an integrated knowledge graph requires a mapping step to accommodate these differences, e.g., that can model both an author's current affiliation and earlier ones.

In the SCM-KG pipeline, we reused subsets of existing vocabularies including the *SWRC ontology*, *Dublin Core* and *FOAF*[12] to create a core vocabulary. We created and matched classes for resources, i.e., nodes, in the source datasets to the core vocabulary modeled initially and instantiated it with one of the data sources.

When the initial vocabulary is missing a definition for a concept from a joining data source we created a new class for it. Thereupon, we linked this new concept to the existing classes by defining a new relation type in our ontology. As a concrete example, DBLP is missing a notion of fields of study but MAG has a distinct index of fields of study and also article keywords, and relates keywords of articles to fields of study. We related the articles integrated from DBLP in the SCM-KG to fields of study with a new RDF property given that we knew their relation to fields of study by integrating MAG.

Challenges in integrating occur with structured datasets whose schemas model the same concept in a way different from the ontology of the knowledge graph existing so far. Nguyen [7] has classified these challenges. As Nguyen describes, a conflict on the concept level occurs when classes with same name have different structures in two merged ontologies. We encountered this issue when mapping the affiliation property. We addressed it by keeping the more descriptive vocabulary in our ontology model and pruning the other, conflicting vocabulary from the model. The notion of an author's

[10] `http://lod.openaire.eu`

[11] `https://scrapy.org`, accessed on 5 April 2017

[12] SWRC: `http://ontoware.org/swrc`, FOAF: `http://xmlns.com/foaf/spec/`

affiliation has a temporal dimension that *swrc:affiliation* used by DBLP does not cover, as it merely models the *current* affiliation, not the affiliation at the time a certain article was published. We simplified a temporal modeling approach proposed by Nuzzolese et al. [8] by following the reification pattern of MAG's *paperAuthorAffiliations* table, i.e., turning each ternary relation of a publication, its authors and their affiliations at the time of publication into a resource. A conflict on the instance level occurs when descriptions of identical instances in different ontologies are different. To resolve it we either could choose only one instance by fact checking their materialized instances against the real word or if possible extend the class of the instance such that it holds both conflicting descriptions for later check. For example, publication dates of some articles are different in MAG and DBLP and we had to find the correct year manually, e.g., via the homepages of their authors.

4.3 Mapping Data to an Ontology

Data acquired from different sources can follow a variety of data models (e.g., graph, relational, tree) or even be unstructured. Thus, having acquired the data, and having modeled a common integration ontology, the next step of constructing a knowledge graph is to convert all data into a common model. RDF is well suited as a target data model for integration and thanks to the wide availability of mapping languages and tools for it, mapping data from different sources to RDF is practically feasible (cf. Section 3).

In our concrete situation, an RDF version of DBLP is already available and the CSV sources of MAG can be mapped to RDF. We developed a process scheduler with a command line interface to execute this step of the pipeline in general. For CSV sources such as MAG, the Sparqlify-CSV tool [5] maps the source ontology to the integration ontology. To use Sparqlify-CSV we expressed mapping rules in its intuitive Sparqlification Mapping Language [14].

In some cases direct mapping of CSV files is not possible. Therefore we implemented an ETL component to shape the data in the format required for the mapping by applying string manipulations. Using a process scheduler, we stream results of the ETL component into Sparqlify-CSV. To improve the performance of mapping, we run multiple parallel instances of the process scheduler. Each row in a CSV file and each set of triples that the Sparqlify-CSV mapping engine creates from it is semantically independent from the other rows. Based on this understanding, the scheduler executes the conversion in parallel processes. After breaking the big input files, e.g., from a size of 9 GB into 20 KB in-memory-processable chunks it creates queues that convert and map the data chunks in parallel and finally merges the respective mapping results. Section 6.3 presents a performance evaluation of this module.

4.4 Calculating Similarity and Instance Matching

In Section 4.2 we addressed how we mapped semi relational data to a common ontology but did not cover the level of mapping *instances* where multiple instances refer to the same real world thing. We therefore added a data linking step to our pipeline. First of all, we keep data integrated from different sources in separate URI namespaces to avoid clashes in case different sources use same identifiers. We then created "same as"

links between different URIs referring to the same thing by instance matching. *Articles* can be matched by common title, publication year and, if provided, the name of the conference or journal. To increase linking coverage, we considered the incidence of variations of title strings in punctuation and letter cases that occurs in different datasets, and compared them using the Jaccard similarity measure. We implemented these conversions and comparisons and the linking of the articles using the Silk workbench [18]. A high-quality instance-level linking of *persons* is a challenge for the Silk Workbench. A mere triple based matching, as applied in Silk, fails to distinguish different persons with similar or even same names.

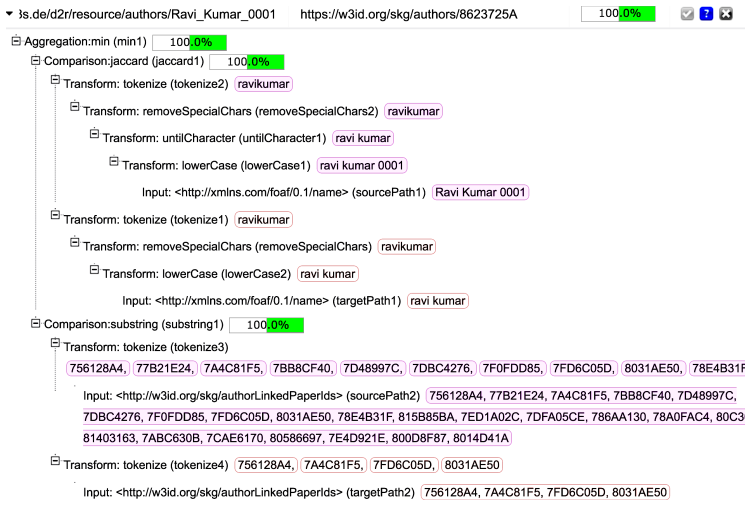

Fig. 2: Matching instances of an author in two datasets

We tackled this problem using the semantic *relations* of the persons with their articles. In our data sources, persons only occur in the role of authors of publications; additionally, we can rely on links between papers as identified in the previous step. We leverage this semantics by embedding it into the author molecules[13]. First we create a hash for each article. Provided that instance matching of articles is performed in the last step and they are stored in the SCM-KG, we find those articles of a person that have been matched to an article in other datasets. We concatenate the IDs of these articles to a comma-separated string. We then associate this string immediately to the author via a new property of type *authorLinkedPaperIds*. We store these new links in the SCM-KG to use them in Silk subsequently.

By applying a substring similarity metric defined by Stoilos et al. [15] on the concatenated list of unique IDs of articles, we can discover if two instances of Person have common publications. The more common publications, the higher the value of this metric. Figure 2 depicts an example of this step of instance matching in action.

[13] Here, a "molecule" refers to a set of one node in the knowledge graph and the immediate links to its neighbors.

4.5 Producing and querying a KG

Our objective in the final pipeline step is to store all the data in a form that is accessible via SPARQL queries. We employed the high-performance Apache Jena TDB as our RDF store. After importing our data into TDB we configured Apache Jena Fuseki 2 to make the data queryable using SPARQL 1.1, both from the command line and, via HTTP, from a SPARQL endpoint. The latter SPARQL endpoint enables the integration of Silk in the linking step and the resulting links are added to the KG in the end. Fuseki also supported the evaluation (cf. Section 6) by enabling us to query the dataset conveniently via a web frontend. To further improve performance, we employed the Cassandra big data database to cache query results of Fuseki. However, we did not consider Cassandra in the evaluation of the query execution time to have a fair comparison.

5 Related Work

Recent approaches toward constructing knowledge graphs, e.g., NOUS [2], Knowledge Vault [3] or NELL [1] focus on materializing a knowledge graph by inferring relations in the existing data. In comparison, our focus was to integrate data from heterogeneous resources and to increase the quality of the integrated knowledge graph. For that we evaluated the steps of the knowledge graph construction pipeline and optimized our pipeline based on that.

In a similar work, Szekely et al. [16] created a knowledge graph of human trafficking data; text and images from the Web were parsed and unstructured data was mapped to a vocabulary. In contrast, we resolved the challenge of structure variations of the data being integrated. As explained in subsection 4.3, we mapped semi-structured metadata into triples using Sparqlify-CSV. This step distinguishes our pipeline from the research of Szekely et al. They integrated data by building up a new ontology model while we modified the existing ontology model of the manually maintained DBLP and aggregated other vocabularies to it. The vocabulary used in DBLP has already a combination of the common vocabularies in describing the scientific metadata. Therefore, we accumulated other terms and vocabularies or modified the current model when the vocabulary of DBLP was not sufficiently describing the integrating data.

Another difference of the two works is the ETL component. From a technical perspective, Szekely et al. used the Karma framework [6] for data mapping. Their approach is limited as they apply ETL the Karma component used for mapping. ETL rules in Karma are in Python, while we implemented an efficient ETL component in C++. Furthermore, Szekely et al. enhanced their linking with image similarly measures, whereas we used semantics of the incoming data to increase the quality of instance matching.

Traverso et al. [17] suggested applying semantics in relation discovery in existing knowledge graphs. Similarly, we apply the concept of semantic molecular similarity, but we use the semantic relations in the network toward the linking of instances during the creation of a knowledge graph.

In a recent research, Danh Le-Phuoc et al. [10] integrated data from variety of resources including sensors, the Twitter social network and RSS resources of famous news websites to create a knowledge graph of things. Their pipeline similarly needs to

process a holistic amount of data in batch and makes them queryable via a SPARQL endpoint. In contrast to our work, they process streaming data coming from resources that are much more loosely coupled in comparison to the resources in our pipeline.

In our experiment one of the data sources is in CSV format, i.e., semi-structured relational data. Many approaches have been investigated to map relational data to RDF, e.g., heuristic and rule-based methods, graph analysis, probabilistic approaches, reasoning, machine learning, etc. We chose a manual rule-based mapping method. This allows for vocabulary reuse but requires users to be familiar with popular Semantic Web vocabularies to choose the most suitable terms [13].

6 Evaluation and results

We conducted an empirical evaluation to study the effectiveness of the proposed pipeline in creating a knowledge graph from different data sources in the domain of scholarly communication metadata (SCM-KG). We assessed the following research questions:

RQ1) Can relative answer completeness be enhanced when queries are executed against an SCM-KG instead of the original sources? Is the query execution time affected when queries are executed against an SCM-KG? **RQ2)** How accurate is the linking of the integrated dataset in terms of precision and relative recall? **RQ3)** How much data can be processed per second in the mapping and linking steps of the pipeline?

Datasets: For the evaluation, we chose a subset of authors and their papers from both DBLP and MAG [12]. This subsection involves all the metadata relevant to the WWW conference series in both datasets[14]. WWW has a long history, and this fraction of data covers all the vocabulary and structure used in the whole dataset. MAG was last updated on 5 February 2016, and we acquired the DBLP dataset on 10 November 2016 from the DBLP++ website[15]. We chose Apache Jena Fuseki as our triple store.

We executed each query 15 times, each time instantiated with a different author. We selected these 15 authors among the most publishing authors in WWW as found by another SPARQL query over the SCM-KG.

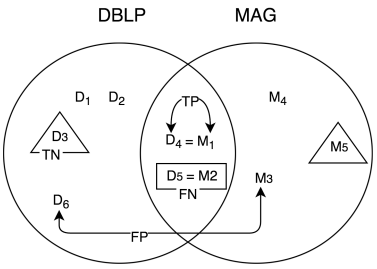

Fig. 3: Articles belonging to an author in DBLP and MAG. Arrows represent the matched instances.

Queries: In the next two experiments, we defined queries and compared their results over the integrated knowledge graph with their evaluation on the isolated source datasets.

Metrics: We evaluated how much the integration enhanced the accuracy and completeness of the query results. Some authors do not have a Google Scholar profile or any other "complete" publication list available, therefore the dataset completeness is calculated in a relative way. In the second experiment, we tested the quality of the linking in terms of relative precision

[14] The integrated WWW dataset has 346,480 triples including the "same as" links between matched instances.

[15] `http://dblp.l3s.de/d2r/sparql`

and recall.[16] The D_4–M_1 connection in Figure 3 is an example of a true positive link. When the equivalence of items is not discovered we consider that a false negative (FN). For example, the lack of a D_5–M_2 connection is a FN. When two articles are linked that are not really equivalent we assume it as a false positive, such as the arrow connecting D_6 and M_3. When the instance matching step correctly does not relate two different articles, we consider this a true negative, depicted as a triangle in the diagram.

We also evaluated the data integration process by comparing the execution times of the queries provided above over the different datasets.

Implementation: Experiments 1 and 2 were run on a test platform with an Intel i7-4710HQ 2.5 GHz CPU and 16 GB 1333 MHz DDR3 RAM; the operating system was Mac OS 10.12. The test queries were executed on Jena Fuseki. In Experiment 3, we used a machine with 32 GB RAM and an Intel(R) Xeon(R) 3.00 GHz CPU with 16 cores; the operating system was openSUSE Linux. We implemented the process scheduler in C++ with a shell script frontend. SPARQL queries were executed to create triples for the semantic based similarity measurement. The process manager, Sparqlify-CSV mapping rules, ETL source code, and the test datasets evaluated are publicly available.[17]

6.1 Experiment One: Relative Completeness

Publications and the number of hits in the different datasets were collected. Queries were executed for each of the 15 selected authors over the three datasets and compared them in terms of relative completeness of the result sets. Comparing the number of WWW publications in MAG, DBLP, and SCM-KG, we observed that although DBLP contains more articles for the selected authors, there exist articles that are only included in MAG. The mapping and linking process allows for identifying common articles in both datasets; thus, the resulting dataset includes more articles for these authors.

Query response time for WWW publications in MAG, DBLP, and SCM-KG indicated that these queries had an average response time of 8.8 ms on DBLP, while equivalent queries on MAG had an average response time of 11.66 ms, and 12.8 ms was the average response time of their equivalent on the integrated graph. These values suggest that the integration did not affect query response time significantly.

6.2 Experiment Two: Linking Accuracy and Relative Coverage

In this survey we ran a SPARQL query over MAG and SCM-KG and evaluated how much the process of linking affected the integrity of the author entities in MAG.

We first defined a query that finds an author entity and his/her articles. It searches instances of authors by name. We observed that for cases like Ravi Kumar the query yields several different author entities instead of one. Likewise, his/her published articles were scattered between different author entities in MAG.

By running the same query over SCM-KG, we observed that the instant matching of author entities in MAG and DBLP had brought these pieces of information together.

[16] In the process of linking articles by an author, true positives (TP) are articles whose metadata exist in both DBLP and MAG and their instances are correctly linked in the matching step.

[17] `https://github.com/EIS-Bonn/SCM-KG`, accessed on 5 April 2017

To survey the indirect merging of authors in MAG, we considered the scattering of an author's articles into each extra instance of an author as a false negative, i.e., author instances in MAG that were equal but not found by the linking process; true positives correspond to merged instances of authors.

This query was executed for 15 selected authors. The comparison of indirect integrated duplicate author entries in MAG, due to instance matching between MAG and DBLP, indicates a correct linking (TP) with a precision of 1 in all cases, and an average recall value of 0.986. Secondly, we tested if, per author, the linked articles belonging to each author are linked to correct equivalent items between datasets. The linking performed in this experiment had a precision of 1 and an average recall of 0.982; these results show the positive effect of using semantic molecular relations in linking.

6.3 Performance evaluation of the mapping process scheduler and linking

In the mapping step, the process scheduler generated 10 parallel processes that occupied approx. 99.5 percent of the available 16 CPU cores and 3.6 GB RAM. By the SCM-KG pipeline, we converted 96.88 GB of MAG and generated approx. 2.9 B triples from MAG and integrated them with 150 M triples from DBLP. The process scheduler could generate approx. 250,000 triples per second, that thanks to parallelization, is significantly faster than the original Sparqlify RDB2RDF transformation engine [5]. The instance matching process could find approximately 500 matches per second when tested on the Mac OS system mentioned in the introduction of section 6.

7 Conclusions and Future Work

In this paper, we presented the concept of Scholarly Communication Metadata Knowledge Graph (SCM-KG), which integrates heterogeneous, distributed schemas, data and metadata from a variety of scholarly communication data sources. As a proof-of-concept, we developed an SCM-KG pipeline to create a knowledge graph by integrating data collected from heterogeneous data sources. We showed the capability of parallelization in rule-based data mappings, and we also presented how semantic similarity measures are applied to determine the relatedness of concepts in two resources in terms of the relatedness of their RDF interlinking structure. Results of the empirical evaluation suggest that the integration approach pursued by the SCM-KG pipeline is able to effectively integrate pieces of information spread across different data sources. The experiments suggest that the rule based mapping together with semantic structure based instance matching technique implemented in the SCM-KG pipeline integrates data in a knowledge graph with high accuracy. Although our initial use case addresses the scientific metadata domain, we generated billions of triples with high accuracy in mapping and linking, and we regard it capable at an industrial scale and in use cases demanding high precision. In the context of the OSCOSS project on Opening Scholarly Communication in the Social Sciences[18], the SCM-KG approach will be used for providing authors with precise and complete lists of references during the article writing process.

[18] `http://eis.iai.uni-bonn.de/Projects/OSCOSS.html`

Acknowledgments: This work has been partially funded by the European Commission under grant agreements 643410 (OpenAIRE2020) and 644564 (BigDataEurope), and the DFG under grant agreement AU 340/9-1 (OSCOSS).

References

1. A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. H. Jr., and T. M. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the 24th AAAI*, 2010.
2. S. Choudhury, K. Agarwal, S. Purohit, B. Zhang, M. Pirrung, W. Smith, and M. Thomas. NOUS: construction and querying of dynamic knowledge graphs. In *ICDE*, 2017.
3. X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *SIGKDD*, 2014.
4. L. Ehrlinger and W. Wöß. Towards a definition of knowledge graphs. In *SEMANTiCS*, 2016.
5. I. Ermilov, S. Auer, and C. Stadler. User-driven semantic mapping of tabular data. In *9th International Conference on Semantic Systems, ISEM*, pages 105–112, 2013.
6. C. A. Knoblock, P. A. Szekely, J. L. Ambite, A. Goel, S. Gupta, K. Lerman, M. Muslea, M. Taheriyan, and P. Mallick. Semi-automatically mapping structured sources into the semantic web. In *ESWC 2012*, pages 375–390, 2012.
7. N. T. Nguyen. A method for ontology conflict resolution and integration on relation level. *Cybernetics and Systems*, 38(8):781–797, 2007.
8. A. G. Nuzzolese, A. L. Gentile, V. Presutti, and A. Gangemi. Semantic web conference ontology - A refactoring solution. In *ESWC 2016 Satellite Events*, pages 84–87, 2016.
9. H. Paulheim. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, 8(3):489–508, 2017.
10. D. L. Phuoc, H. N. M. Quoc, H. N. Quoc, T. T. Nhat, and M. Hauswirth. The graph of things: A step towards the live knowledge graph of connected things. *J. Web Sem.*, 37-38, 2016.
11. A. Singal. Introducing the knowledge graph: Things, not strings, 2012.
12. A. Sinha, Z. Shen, Y. Song, H. Ma, D. Eide, B. P. Hsu, and K. Wang. An overview of microsoft academic service (MAS) and applications. In *WWW Companion*, 2015.
13. D. Spanos, P. Stavrou, and N. Mitrou. Bringing relational databases into the semantic web: A survey. *Semantic Web*, 3(2):169–209, 2012.
14. C. Stadler, J. Unbehauen, P. Westphal, M. A. Sherif, and J. Lehmann. Simplified RDB2RDF mapping. In *Proceedings of the Workshop on Linked Data on the Web, LDOW 2015*, 2015.
15. G. Stoilos, G. B. Stamou, and S. D. Kollias. A string metric for ontology alignment. In *ISWC*, pages 624–637, 2005.
16. P. A. Szekely, C. A. Knoblock, J. Slepicka, C. Yin, A. Philpot, A. Singh, D. Kapoor, P. Natarajan, D. Marcu, K. Knight, D. Stallard, S. S. Karunamoorthy, R. Bojanapalli, S. Minton, B. Amanatullah, T. Hughes, M. Tamayo, D. Flynt, R. Artiss, S. Chang, T. Chen, G. Hiebel, and L. Ferreira. Using a knowledge graph to combat human trafficking. In *ISWC*, 2015.
17. I. Traverso Ribón, G. Palma, A. Flores, and M. Vidal. Considering semantics on the discovery of relations in knowledge graphs. In *EKAW*, pages 666–680, 2016.
18. J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov. Silk – a link discovery framework for the web of data. In *Proceedings of the 2nd Linked Data on the Web Workshop*, pages 1–6, 2009.