Knowledge and Information Systems (2022) 64:1239–1262
https://doi.org/10.1007/s10115-022-01668-7

REGULAR PAPER

Context mining and graph queries on giant biomedical knowledge graphs

Jens Dörpinghaus[1,4] · Andreas Stefan[2,3] · Bruce Schultz[2] · Marc Jacobs[2]

Received: 9 May 2020 / Revised: 17 February 2022 / Accepted: 18 February 2022 /
Published online: 29 March 2022

Abstract
Contextual information is widely considered for NLP and knowledge discovery in life sciences since it highly influences the exact meaning of natural language. The scientific challenge is not only to extract such context data, but also to store this data for further query and discovery approaches. Classical approaches use RDF triple stores, which have serious limitations. Here, we propose a multiple step knowledge graph approach using labeled property graphs based on polyglot persistence systems to utilize context data for context mining, graph queries, knowledge discovery and extraction. We introduce the graph-theoretic foundation for a general context concept within semantic networks and show a proof of concept based on biomedical literature and text mining. Our test system contains a knowledge graph derived from the entirety of PubMed and SCAIView data and is enriched with text mining data and domain-specific language data using Biological Expression Language. Here, context is a more general concept than annotations. This dense graph has more than 71M nodes and 850M relationships. We discuss the impact of this novel approach with 27 real-world use cases represented by graph queries. Storing and querying a giant knowledge graph as a labeled property graph is still a technological challenge. Here, we demonstrate how our data model is able to support the understanding and interpretation of biomedical data. We present several real-world use cases that utilize our massive, generated knowledge graph derived from PubMed data and enriched with additional contextual data. Finally, we show a working example in context of biologically relevant information using SCAIView.

Keywords Current research information systems · Knowledge graphs · Graph embeddings · Semantic search · Complexity · NLP · Graph theory

✉ Jens Dörpinghaus
jens.doerpinghaus@bibb.de

[1] Federal Institute for Vocational Education and Training (BIBB), Bonn, Germany

[2] Fraunhofer Institute for Algorithms and Scientific Computing SCAI, Sankt Augustin, Germany

[3] Bonn-Rhein-Sieg University of Applied Sciences, Sankt Augustin, Germany

[4] German Center for Neurodegenerative Diseases (DZNE), Bonn, Germany

1 Background

The amount of available and stored data is constantly increasing in many areas in the course of digitalization. The increasing amount of data represents a great challenge for storage and requires the development of new storage technologies. At the same time, with more available data and different storage technologies, new applications based on the data are of great interest. Large data collections are used for *data mining* and *knowledge discovery* to answer new and complex questions more efficiently. For this purpose, data is often stored in non-relational databases, and while there are many types available, one of the more interesting and promising types is *knowledge graphs*. In this database structure, the entities of a domain are stored as nodes in a graph while connections between these entities are represented by edges. This allows for visualization and analysis of networks between the data in order to discover new applications.

Current systems use RDF (Resource Description Framework) Triple Stores, systems that inherently have some serious limitations especially when compared to a labeled property graph. For example, nodes and edges have no internal structure which does not allow complex queries like subgraph matchings or traversals and it is not possible to uniquely identify instances of relationships which have the same type, see [1]. Several approaches have been made to create RDF knowledge graphs, for example Bio2RDF (see [2] and [3], reviewed by [4] or [5]). For our generalized concept of context, we require labeled property graph structures.

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 [6], 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 also to be able to store this data for further natural language processing (NLP), querying and discovery approaches. Here, we propose a multiple step knowledge graph-based approach to utilize context data for biological research and knowledge expression based on our results published in [7]. We present a proof of concept using biomedical literature and present an outlook on additional improvements which can be implemented in the next generation of knowledge extraction, e.g., training approaches from artificial intelligence and machine learning. Figure 1 depicts a real-world example subgraph induced by both automatically detected and manually curated context data which highlights the complexity and density of these graphs.

Knowledge graphs have been shown to play an important role in recent knowledge mining and discovery. A *knowledge graph* (sometimes also called a *semantic network*) is a systematic way to connect information and data to knowledge on a more abstract level compared to language graphs. This type of data structure has many advantages in terms of searching within biomedical data and serves as a vital tool capable of generating novel ideas. Another important attribute when generating knowledge is context and therefore connecting knowledge graphs using contextual information can further enhance data analysis and hypothesis generation.

As a basis for this work, we generated a knowledge graph that initially contains publication metadata from *PubMed* (see https://www.ncbi.nlm.nih.gov/pubmed) which has more than 30 million documents at its disposal, including biomedical publications. In subsequent steps, the knowledge graph was expanded to include *BEL* (Biological Expression Language) relations and named entities obtained from text mining using JProMiner (see [8]) and stored in SCAIView (see https://www.scaiview.com/) as well as ontologies or terminologies like *MeSH*. This results in a large amount of data for the graph with a very high number of nodes

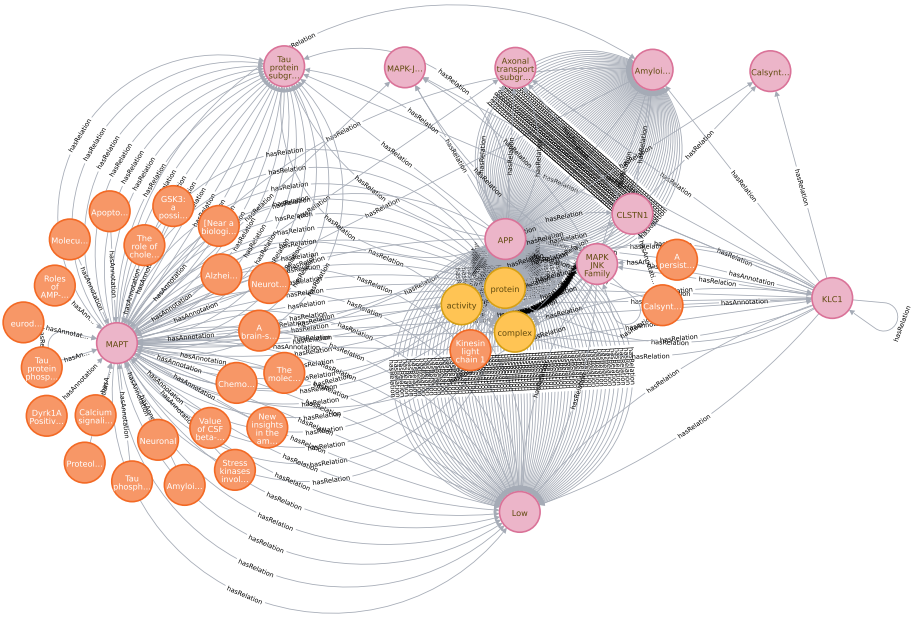

Fig. 1 Here, we present an example subgraph of the knowledge graph that shows several molecular key players (like Amyloid-beta precursor protein (APP), and Calsyntenin-1 (CLSTN!)) and their interactions which are involved in processes that finally lead to acquiring Alzheimer's disease. We focus on the BEL statement `act(p(HGNC:KLC1))->p(HGNC:MAPT)`. This BEL statement describes that the phosphorylation activity of the human (HGNC) "Kinesin light chain 1" (KLC1) protein leads to a phosphorylation of the "Microtubule-associated protein tau" (MAPT) protein. Which is a key event. The Cypher-query found all relations and this graph was extended to the neighborhood of both nodes. Both HGNC terms have evidences in different documents (orange). Some context entities (pink nodes). Several entities can be found in this context, for example APP, CLSTN1 or SFAM MAPK JNK Family. Some other entities are automatically detected using text mining, others are manually curated like confidence value Low (bottom) or subgraph annotations like "Tau protein subgraph" or "Axonal transport subgraph" (top) (colour figure online)

and edges. Saving and managing such a graph poses challenges due to the horizontal scalability of graph databases, therefore, it is to be expected that search queries on the graph have a long runtime. This paper presents a polyglot persistence approach to tackle this challenge using Neo4j, a graph database with a native graph storage.

Here, we use a general definition of context data assuming that each information entity can also be contextual information for other entities, for example a document can also serve as context for other documents (e.g., by citing or referring to the other publication). An author is both metainformation for a document, but also itself context (by other publications, affiliations, co-author networks, ...). Other data is more obviously purely context: named entities, topic maps, keywords, etc. extracted with text mining from documents. However, relations extracted from a text document may stand for themselves, occurring in multiple documents and still valuable without the original textual information.

To start, we begin with a simple document graph and, in the first step, we added context metainformation (see Fig. 2). This leads to an initial knowledge graph which can be used for preliminary context-based text mining approaches. In doing so, additional context data is added to the knowledge graph, such as entities or concepts from ontologies or relations extracted from the analyzed text. The resulting knowledge graph can be used as starting basis

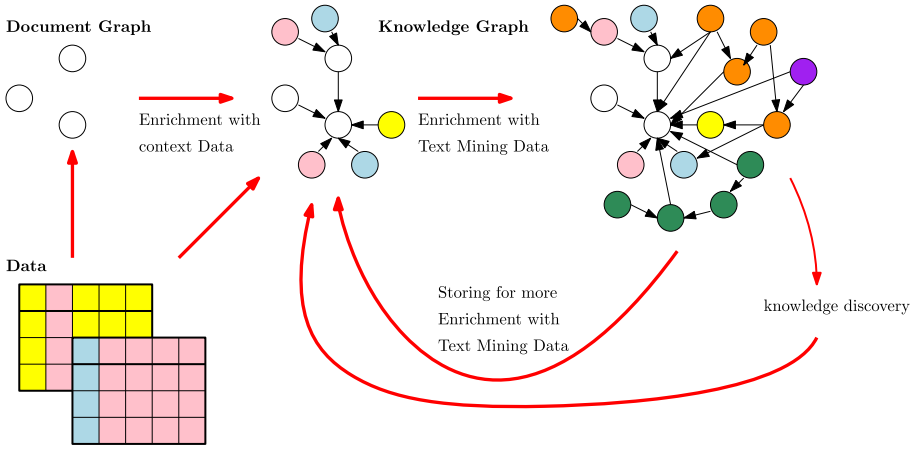

Fig. 2 Proposed workflow to extend a knowledge graph. First starting with a document graph, the basic document metainformation like authors, keywords, etc. are added. This can be used as a basis for text mining which can be used to extend the graph again, for example named entity recognition (NER) may use normalized keywords as context. Topic detection may also benefit from already assigned keywords, journals or author information. The graph can also be extended by knowledge discovery processes, for example finding parameters of a clinical trial, progression within electronic health records, etc. In any case new context information are added to the initial graph and improve the input of further algorithms

for more detailed text mining approaches which utilize the novel context data. These steps can be repeated several times to further enrich the graph.

In fact, using a graph structure to house data has several additional advantages for knowledge extraction: biological and medical researchers, for example, are interested in exploring the mechanisms of living organisms and gaining a better understanding of underlying fundamental biological processes of life. Systems biology approaches, such as integrative knowledge graphs, are important to decipher the mechanism of a disease by considering the system as a whole, which is also known as the holistic approach. To this end, disease modeling and pathway databases both play an important role. Knowledge graphs built using BEL are widely applied in biomedical domain to convert unstructured textual knowledge into a computable form. The BEL statements that form knowledge graphs are semantic triples that consist of concepts, functions and relationships [9]. In addition, several databases and ontologies can implicitly form a knowledge graph. For example Gene Ontology, see [10] or DrugBank, see [11] or [12] cover a large amount of relations and references to which reference other fields.

There are still several crucial issues to consider when converting literature to knowledge such as evaluating the quality and completeness of such networks. Here, we rely on existing data sets and present a novel approach on this data. We do not omit the question of quality control this as a task of the initial data. Furthermore, in order to generate new knowledge, context of concepts in a knowledge graph must be considered.

To start, we first present a preliminary overview about information theory and management. Afterward, we will introduce and discuss the novel approach of managing and mining contextual data of knowledge graphs. Finally, we will give a detailed list of issues that need to be addressed and show the results from evaluating real use cases.

1.1 Preliminaries

Data and knowledge management, sometimes also called information management, is a core topic of data engineering and data mining. It is also an interdisciplinary field encompassing economics (how efficient and expensive is the solution?), psychology (do people use this solution in a way that was intended?) and, of course, informatics. One of the core concepts is DIKW (data, information, knowledge, wisdom, see [13]), an approach used to describe all of the important steps which are necessary to understand the ideas of data and knowledge management.

Knowledge is often seen as either explicit or implicit, while data is always presented as an explicit concept. It is important to note that implicit knowledge is not available for data mining as it is only available as personal knowledge or experience. In information theory, knowledge is obtained from data and information. Data are recorded, context-free facts such as measured values from devices (mass spectroscopy) or basic notes (weight of patients), but can also include images (e.g., computer tomography). If this data is enriched by context, which implies meaning and purpose we get information. This information leads to knowledge and wisdom if—once again—enriched by context.

The concept of DIKW hierarchy is crucial for the understanding of the work presented here. First proposed by [14] in 1987, it was developed by [15] in 1989 who also introduced the perspective of wisdom. At times, this hierarchy is depicted as a *knowledge pyramid* while other times it is a linear chain. We may combine both perspectives: The linear perspective of understanding and context with past and future and the pyramid's perspective describing the amount of data leading to a smaller amount of information, etc. More information about this topic can be found in the work of [13] or [16].

A *knowledge graph* is a systematic way to connect information and data to knowledge. It is thus a crucial concept on the way to generate knowledge and wisdom, to search within data, information and knowledge.

Definition 1.1 (*Knowledge graph*) We define a knowledge graph as graph $G = (E, R)$ with entities $e \in E = \{E_1, \ldots, E_n\}$ coming from a formal structure E_i like ontologies.

The relations $r \in R$ can be ontology relations, thus in general we can say every ontology E_i which is part of the data model is a subgraph of G indicating $E \subseteq G$. In addition, we allow inter-ontology relations between two nodes e_1, e_2 with $e_1 \in E_1$, $e_2 \in E_2$ and $E_1 \neq E_2$. In more general terms, we define $R = \{R_1, ..., R_n\}$ as a list of either inter-ontology or inner-ontology relations. Both E as well as R are finite discrete spaces.

Every entity $e \in E$ may have some additional metainformation which needs to be defined with respect to the application of the knowledge graph. For instance, there may be several node sets (some ontologies, some document spaces (patents, research data, ...), author sets, journal sets, ...) $E_1, ..., E_n$ so that $E_i \subset E$ and $E = \cup_{i=1,...,n} E_i$. The same holds for R when several context relations come together such as "is cited by," "has annotation," "has author," "is published in," etc.

Definition 1.2 (*Context*) We define context C as a set with context subsets $C = \{c_1, ..., c_m\}$. This is a finite, discrete set. Every node $v \in G$ and every edge $r \in R$ may have one or more contexts $c \in C$ denoted by $con(v) \subset G$ or $con(r) \subset G$.

It is also possible to set $con(v) = \emptyset$. Thus we have a mapping $con : E \cup R \to \mathcal{P}(C)$. If we use a quite general approach toward context, we may set $C = E$. Therefore, every inter-ontology relation defines context of two entities, but also the relations within an ontology can

be seen as context. With the neighborhood $N(E_i)$ every node set $E_i \in \{E_1, ..., E_n\}$ induces a subgraph $G[E_i] \subset G$:

Definition 1.3 (*Extended context subgraph, graph embeddings*) With $G^c[E_i] = G[E_i] \cup N(E_i)$ we denote the extended context subgraph which also contains the neighbors of each node in G, which is context of that node.

For a graph drawing perspective, if $G^c[E_i]$ defines a proper surface, we can think about a graph embedding of another subgraph $G^c[E_j]$ on $G^c[E_i]$. This concept was introduced in [17]. Here, semantic knowledge graph embeddings were displayed between different layers. Every layer (for example: molecular layer, document layer, mechanism layer) corresponds to another context defining new contexts on other layers.

Definition 1.4 (*Context metagraph*) We can create the metagraph $M = (C, R')$ of these contexts. Each context is identified by a node in M. If there is a connection in G between two contexts, we add an edge $(c_1, c_2) \in R'$. This means if $\exists (v_1, v_2) \in R : \ c_1 \in con(v_1),\ c_2 \in con(v_2) \Rightarrow (c_1, c_2) \in R'$ or $\exists (v_1, v_2) \in R : \ c_1 \in con((v_1, v_2)),\ c_2 \in con(v_2) \Rightarrow (c_1, c_2) \in R'$ or $\exists (v_1, v_2) \in R : \ c_1 \in con(v_1),\ c_2 \in con((v_1, v_2)) \Rightarrow (c_1, c_2) \in R'$.

Adding edges between the knowledge graph G or a subgraph $G' = (E', R') \subseteq G = (E, R)$ and the metagraph M in $G \cup M$ will lead to a novel graph. This can be either seen as inverse mapping $con^{-1}(G')$ or as the hypergraph $\mathcal{H}(G') = (X, \hat{E})$ given by

$$X = E' \cup G^c[E_i], \ \ \hat{E} = \{\{e_i, e \forall e \in N(e_i)\} \forall e_i \in X\}$$

This graph can be seen as an extension of the original knowledge graph G' where contexts connect not only to the initial nodes, but also every two nodes in G' are connected by a hyperedge if they share the same context.

If $C = E$, this will lead to new edges in G thus enriching the original graph. This step should be performed after every additional extension of graph G.

We denote this hypergraph H on a knowledge graph G and a metagraph M with $H_{G|M}$. We can add multiple metagraphs M_1 and M_2 which is denoted by $H_{G|M_1,M_2}$.

The resulting graph can thus be seen as an enrichment of the original knowledge graph G with contexts. It can be used to answer several research questions and to find graph-theoretic formulations of research questions.

If the mapping *con* is well defined for the domain set, then Graph H can be generated in polynomial time. Since this is generally not the case, this step usually contains data or text mining task to generate other contexts from free texts or knowledge graph entities. With respect to the notation described in [18] this problem p can be formulated as $p = \mathbb{D}|R|\mathbf{f} : \mathbb{D} \rightarrow \mathbb{X}|err|\emptyset$. Here, the domain set $\mathbb{D}$ is explicitly given by $\mathbb{D} = G$ or—if additional full-texts $\hat{D}$ supporting the knowledge graph G exist—$\mathbb{D} = \{G, \hat{D}\}$, which in our case is the domain subset $R = \mathbb{D}$. Therefore, we need to find a description function $f : \mathbb{D} \rightarrow \mathbb{X}$ with a description set $\mathbb{X} = C$ which holds all contexts. To find relevant contexts, we also need to measure the error as defined by $err : \mathbb{D} \rightarrow [0, 1]$.

Several research questions must be considered. First, what metainformation can be used to generate context for a new metagraph? Several promising candidates include authors, citations, affiliation, journal, MeSH terms and other keywords since they are all available in most databases. We also need to discuss text mining results such as NER and relationship mining. Having more general data including study data, genomics, images, etc. we might also consider side effects; disease labels, population labels (male; female; age; social class; etc.). Figure 2 shows a proof of concept for a less complex text mining metadata approach

which describes the process of starting with a simple document graph that can be extended with more context data derived from text mining. We discuss this in more detail in the next section.

The second research question addresses the application of this novel approach for both biomedical research as well as text classification and clustering, NLP and knowledge discovery, with a focus on Artificial Intelligence (AI). How can we use the context metagraph to answer biomedical questions? What can we learn from connections between contexts and how do they look like in the knowledge graph? How can we use efficient graph queries utilizing context? It may also be useful to filter paths in the knowledge graph according to a given context or to generate novel visualizations. A possible question might be to learn about mechanisms linked to comorbidities or mechanisms being contextualized by drug information. The metagraph may also contain information about cause-and-effect relationships in the knowledge graph that are "valid" in a biomedical sense under certain conditions as well as contextualization based on demographic information or polypharmacy information. We will discuss several use cases in the last section of this paper.

1.2 Method

1.2.1 Technical setup

We illustrate the following methods with example runs on PubMed and PMC data. Both sources are already included in the SCAIView NLP-pipeline. PubMed contains 30 million abstracts from biomedical literature, while PMC houses nearly 4 million full-text articles. First and foremost, the knowledge graph must be stored and accessed by the software in an efficient manner. To this end, a software component was written to integrate the knowledge graph into our SCAIView microservice architecture, see [19]. This integration also ensures that the knowledge graph is constantly updated with preprocessed data. The software component also provides an API to execute several queries on the knowledge graph and is capable of returning the result in JSON Graph Format which can be easily displayed by many frontend frameworks.

Our software component was written in Java using Spring Boot and Spring Data to be able to access the database backend in an abstract way and ensure the exchangeability of the database technology. The database backend in our case is the graph database Neo4j. To this end, we designed a software component that exports the data derived from SCAIView as CSV files.

Storing a large knowledge graph from PubMed, such as the one presented here, in a single database is not a simple task, and we expected the execution of our graph queries to be very slow due to the size of the knowledge graph. To speed up the run times of the queries, we decided to implement an approach that divides the graph using polyglot persistence. Polyglot persistence is defined as combining heterogenous data storing technologies into a single application. Instead of storing all of the data in one database, we chose to store different parts of the data in different database technologies. The benefit of polyglot persistence is that each database technology has different strengths and the application can take advantage of them all.

In Neo4j, the graph structure is stored separately from the properties of nodes and edges. This organization structure makes traversing the knowledge graph easier, however, storing and accessing string attributes takes longer than integer attributes because of this property [20]. To take advantage of this characteristic of Neo4j, we designed a storing system that

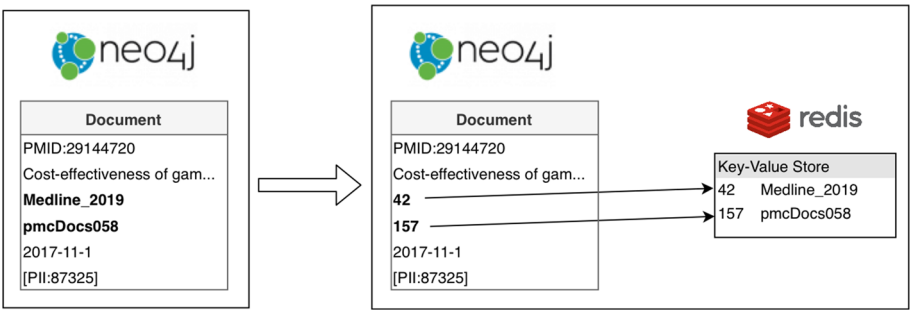

Fig. 3 Example of a stored document node in Neo4j. On the left side, a PubMed document is stored with all of its attributes. Using polyglot persistence we see on the right side the same document containing integer encodings for two original attributes in Neo4j. The encoding of the used attributes is stored in the key-value database Redis. An other attribute for content of the document like "Cost-effectiveness..." is still stored as its original string value

encodes either some or all string (depending on the test scenario) attributes of the graph as integers using polyglot persistence. By encoding and storing these attributes in key-value databases, we reduced the data size of the knowledge graph and were able to speed up the property access of Neo4j. Figure 3 provides an illustration of the designed polyglot persistence system.

In two iterations, we selected suitable attributes of all node types thus leading to three systems: the original one using only Neo4j (called *Full*) and two polyglot persistence systems (called *Poly1* and *Poly2*). *Full* stores all data directly in Neo4j. *Poly1* stores only a few information in another redis database while *Poly2* uses not a single redis database but rather combines multiple redis databases storing different information and the Neo4j graph database.

We implemented another software component to execute the data preprocessing step for *Poly1* and *Poly2*. It uses the created CSV input files of *Full* to run the data encoding in key-value databases and generates CSV input files for the Neo4j graph databases of the polyglot persistence systems. The whole process is illustrated in Fig. 4.

To compare the execution runtime of queries on all three systems *Full*, *Poly1* and *Poly2*, we collected 27 real-word graph queries using the given knowledge graph. The results of the query runtimes are discussed in Sect. 2.

1.2.2 Creating a document and context graph with basic context extraction

The first step in creating a document and context graph with basic context extraction is to define the entity sets E_1, ..., E_n and their relations. The articles and abstracts from PubMed and PMC already contain a lot of contextual data. Thus, the starting point for our data schema is straight forward: We define $E_{Document}$ as the document set containing nodes, with each one representing one document. Furthermore, we may add a set $E_{Source} = \{\text{PubMed, PMC}\}$ as the source of a document. Thus, each document can be interpreted as contextual data of a particular data source.

Since the original data set contains a lot of additional metadata, we need to add them as single data points: all metadata are stored in new node sets. E_{Author} stores the set of authors and $E_{Affiliation}$ stores their affiliation, which is again considered context for the authors. Another relevant piece of contextual information is the publisher, in our case $E_{Journal}$. PubMed has several classifications for $E_{Journal}$ including: Books and Documents, Case

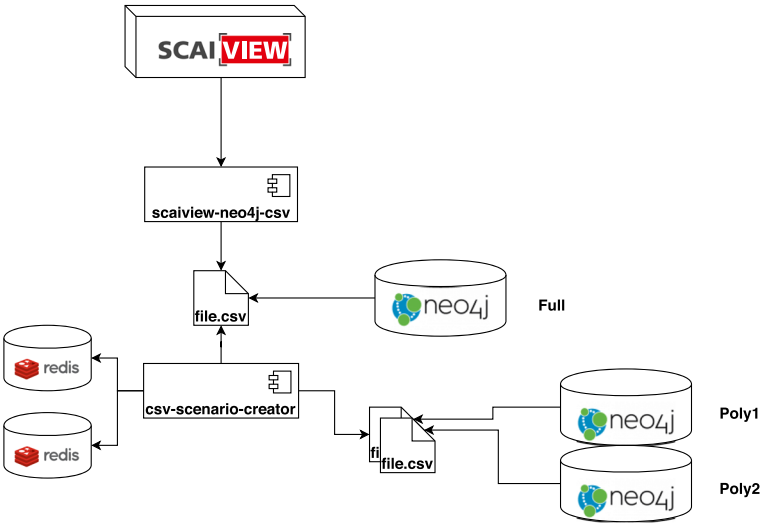

Fig. 4 The software component `scaiview-neo4j-csv` creates CSV files for the bulk import in Neo4j from SCAIView data. The created files are used as input for the system called *Full*. The second software component `cdv-scenario-creator` uses the CSV files, runs the encoding of the selected string attributes and created CSV import files for *Poly1* and *Poly2*

Table 1 All node types and an excerpt of attributes

Node type	Attributes
Author	Forename, surname
Affiliation	Affiliation
Document	DocumentId, title, collection, provenance, etc.
Journal	Journal
PublicationType	Identifier, type
Entity	Source, identifier, preferredLabel, uri
Unstructured	Value, uri
BELFunction	

These nodes are linked with relations (e.g., `hasAffiliation`, `isAuthor`, `hasDocument`, `hasRelation` (Attributes: type, function, provenance, context) and `optionalLabel`)

Reports, Classical Article, Clinical Study, Clinical Trial, Journal Article and Review. We store this classification in $E_{PublicationType}$. Here, the relations are directly induced by the original data schema: `hasAffiliation`, `isAuthor`, `hasDocument`, `hasCitation` (Attribute: provenance), `isOfType`.

Other important context directly obtained from the initial document data is $E_{Annotation}$ which stores multiple types of annotations such as named entities or keywords, all of which come from the MeSH tree, see [21]. Therefore, $E_{MeSH} \subset E_{Annotation}$ inherently contains a hierarchy and edges R_{MeSH}. The value of MeSH terms and their hierarchy for knowledge extraction was shown in several recent studies [22]. Figure 5 depicts the knowledge graph of a single document; Table 1 shows a list of all node types and relations.

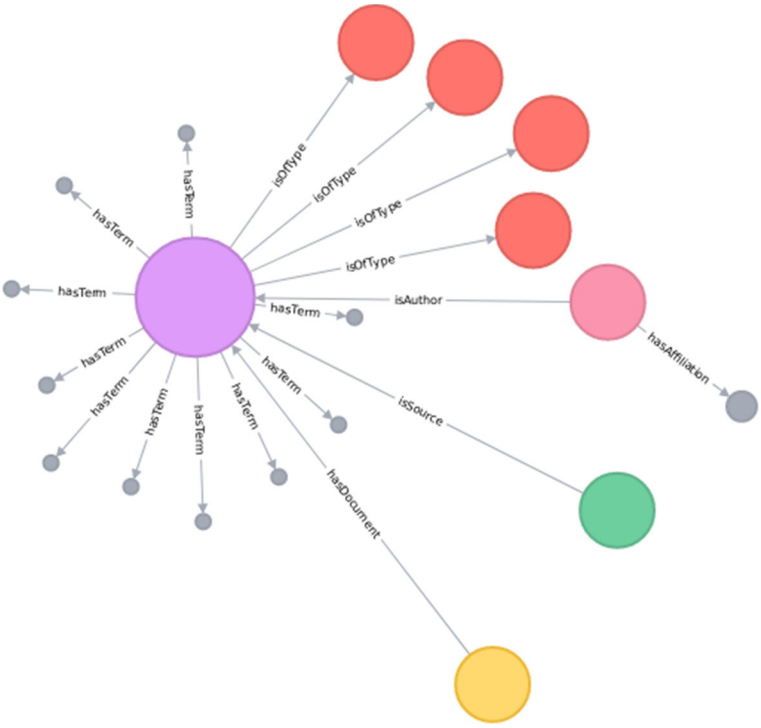

Fig. 5 An illustration of a single document within the context graph. The document node (purple) has several gray annotation nodes, four red publication type nodes, a pink author node with a gray affiliation. The source (PubMed) is annotated in a green node, the journal in a yellow node (colour figure online)

All other relations can be added between the sets E_i, for example $R_{isCoAuthor}$, $R_{hasAffiliation}$, etc. With this information, it is—from an algorithmic point of view—quite easy to combine all context relations such as $R_{hasDocument}$, $R_{isAuthor}$, $R_{hasAnnotation}$ and $R_{hasCitation}$, though these edges should also store additional provenance information as shown in Fig. 6.

1.2.3 Extending the knowledge graph using NLP-technologies

The initial knowledge graph can be extended by NLP-technologies. Terminologies and Ontologies are a widely considered topic in research during the last years. They play an important role in data and text mining as well as knowledge representation in the semantic web. They have become increasingly more important once data providers began publishing their data in a semantic web formats, namely Resource Description Framework (RDF, see [23]) and Web Ontology Language (OWL, see [24]), to increase integratability. The term *terminology* refers to the Simple Knowledge Organization System metamodel (SKOS, see [25] which can be summarized as concepts, unit of thoughts which can be identified, labeled with lexical strings, assigned notations (lexical codes), documented with various types of note, linked to other concepts and organized into informal hierarchies and association networks, aggregated, grouped into labeled and/or ordered collections and mapped to concepts. Several complex models have been proposed in the literature and have been implemented in

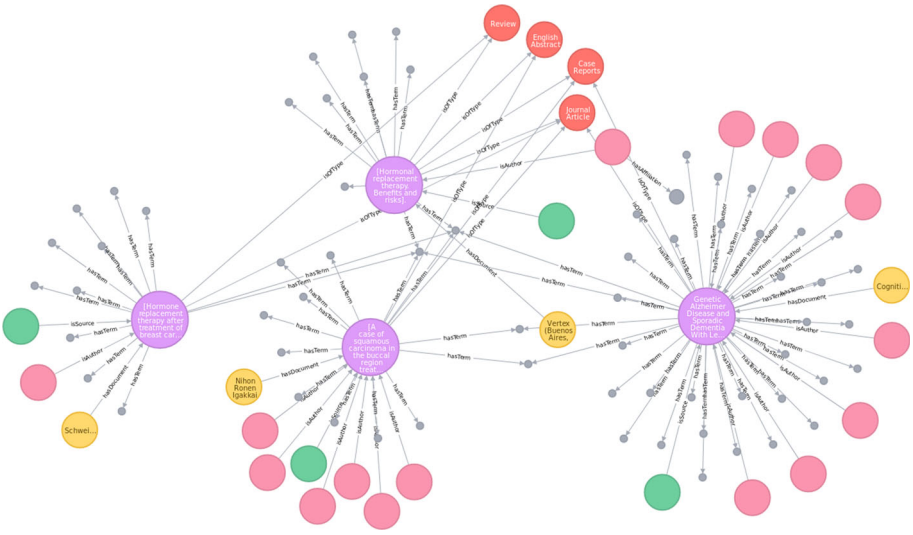

Fig. 6 An illustration of the initial document and context graph. A PubMed node is the source of document nodes (purple). There are several context annotations like article type (red), keywords (gray), authors (pink) and journal (yellow). Authors have additional context (affiliations, gray) (colour figure online)

software, see [26]. *Controlled Vocabularies* contain lists of entities which may be completed to a *Synonym Ring* to control synonyms. *Ontologies* also present properties and can establish associative relationships which can also be done by *Thesauri* or *Terminologies*. See [27] and [28] for a complete list of all models.

Here, we define Terminologies similar to Thesauri as a set of concepts. They form a *DAG* with child and parent concepts. Additionally, we have an associative relation which identifies related concepts. Each concept has at least one label, one of which is used as the preferred identifier while all others are synonyms. To sum up, using ontologies or terminologies for NER has several advantages. In particular, it leads to a hierarchy within these ontologies and orders named entities according to these relations. Though, we must not only consider ontologies and terminologies, but also controlled vocabularies such as MeSH. Here, we have additional annotations with different provenances, one derived as keywords with the data and one obtained from NER. The relations itself can be determined by using the original data structure: Either they are related to a document and thus describe an annotation or they describe a relation between two entities and the relation is described with the original data set which we will describe later.

Another example of a terminology is the Alzheimer's Disease Ontology (ADO, see [29]) E_{ADO} or the Neuro-Image Terminology (NIFT, see [30]) E_{NIFT} coming with their hierarchy R_{ADO}, R_{NIFT}. The process of NER leads to another context relation $R_{hasAnnotation}$. Since not all ontologies or terminologies are described using the RDF or OBO format, we have to add data using multiple external sources via a central tool capable of providing all the necessary ontology data. We use a semantic lookup platform containing Ontology Lookup Service (OLS) and Ontology Xref Service (OxO) (see [31]).

Additional context data useful for knowledge extraction are citations such as the edges $R_{hasCitation}$ between two nodes in $E_{Document}$. Data from PMC already contains citation data with unique identifiers (PubMed IDs). Some data is available with WikiData, see [32] and

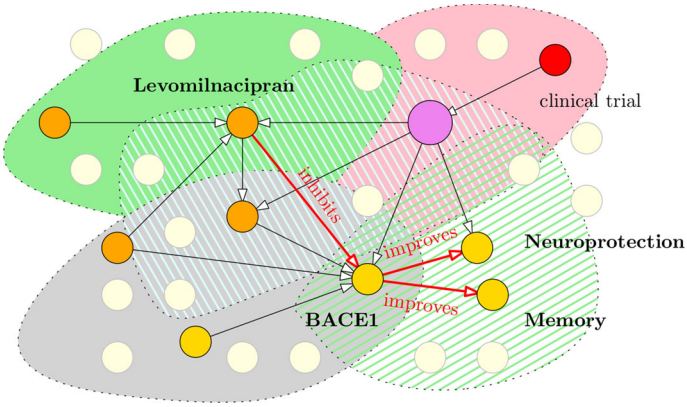

Fig. 7 An illustration of biological knowledge within the context graph. The document node (purple) has several orange annotation nodes which come from different terminologies found with NER. The areas in the background indicate arbitrary context subgroups to highlight that the different nodes belong to different backgrounds. The relation extraction task found the relation "Levomilnacipran" inhibits "BACE1," "BACE1" improves "Neuroprotection" and "BACE1" improves "Memory." These relations are illustrated with red edges. Since the document describes a clinical trial, this is also context for the relations as well. All other context is illustrated by colored sets, defining subgraphs (colour figure online)

[33]. Other sources are rare, but exist, see [34]. Especially for PubMed a lot of research is working on this difficult topic, see for example [35].

Furthermore, we can consider the relational information between entities. For example, BEL statements naturally form knowledge graphs by way of semantic triples that consist of concepts, functions and relationships [9]. To tackle such complex tasks they constantly gather and accumulate new knowledge by performing experiments, and also studying scientific literature that includes results of further experiments performed by researchers. Existing solutions are primarily based on the methods of biomedical text mining which consists of extracting key information from unstructured biomedical text (such as publications, patents and electronic health records). Several information systems have been introduced to support curators in generating these networks such as BELIEF, a workflow that builds BEL-like statements semi-automatically by retrieving publications from a relevant corpus generator system called SCAIView, see [36] and [37].

Figure 7 illustrates a few basic relations such as *"Levomilnacipran" inhibits "BACE1," "BACE1" improves "Neuroprotection"* and *"BACE1" improves "Memory,"* all of which were found using relation extraction methods on named entities in a document. Here, the relations between entities are directly described as BEL relations. It is important to note that context for a document can also be context for the derived relations and vice versa. If an entity that forms part of a relation has synonyms, or is found within another document with a different context, this may lead to a deeper understanding about the statement. Due to the complexity, the resulting graph structures become difficult to manually parse and interpret thus requiring algorithmic approaches to properly analyze.

2 Results

2.1 Real-world use cases for testing

We collected 27 real-world questions and queries in scientific projects. They are of varying complexity (Table 2) and can be used to test the biomedical knowledge graph. Some of them use local structures, for example conjunctive regular path queries (CRPQ, see [38]) which combine subgraph pattern with queries regarding paths (problems 1,3,5,7,9,10,13,15,20) or the extended version ECRPQ (8,18,22). Other local structures include Regular Path Queries (RPQ, see [39]) (problems 2,11,14,16,17,19,21) and finding shortest path (problems 4,12). Additional queries use global structures such as centrality which include Page Rank (6,23), Betweenness Centrality (25) or Degree Centrality (26). Another global problem is community detection, for example Louvain Modularity (24) or Connected Components (27).

Because the general subgraph isomorphism problem is known to be NP-complete, we expect that some of our queries, such as finding the shortest paths in P, to require a wide range runtimes.

2.2 Storing the knowledge graph

Storing all of the data in one graph database without using Redis (Full) uses 58.9 GB of memory, while Poly1 only uses 50.82 GB (Neo4j) and 0.9 GB (Redis) of memory. The third system, Poly2, uses 50.74 + 10.2 GB (Neo4j) and 1.4 GB (Redis) memory.

The import data is about 50 GB and generates nearly 160M nodes with relations. These nodes are merged by Neo4j to unique nodes. In the end, we obtained 71M unique nodes and 860M relationships. Given the input data, we create ∼30M nodes describing documents from PubMed and PMC, about 17M dedicated to authors, 21M affiliations and around 5M entities. The graph contains 554M annotation relationships and in total 850M relationships.

2.3 Polyglot persistence systems

Figure 8 shows the runtime results of the 27 real-world queries described in Table 2.

We see that execution of some queries required a large amount of time with the longest query taking more than one hour. Interestingly, the execution time for most of the queries improved when ran using either the Poly1 or Poly2 implementation. We experienced that seven out of the 27 queries did not terminate. This was mainly due to main memory issues, other reasons for endless runtime could not be examined but we assume time complexity and implementation issues for that.

For most queries, the polyglot persistence systems achieve better results, in the best case up to 43%. However, there are differences between the systems for a few of the queries tested in that *Poly1* can sometimes have better results than *Poly2* and vice versa. Contrary to expectations, *Full* was found to have the best query time in most cases. The advantage of *Poly1* over *Poly2* can be explained by the fact that the memory consumption of *Poly2* increased significantly due to the process of converting from string to integer and therefore the execution of the queries is slowed down. For the queries in which *Poly2* performed better, this can be explained by the fact that the queries take advantage of the optimized polyglot data schema despite the higher memory consumption of the database. This is significant for example in queries 8 and 17.

Table 2 Biomedical example queries on knowledge graphs with context data

#	Query	Input example	Output
1	Which author was the first to state that {Entity1} has an enhancing effect on {Entity2}?	APP, gamma Secretase Complex	Author and document title
2	Which genes {Entity1} play a role in two diseases {Entity2}?	Entity.source = HGNC, MESH	Subgraph of genes with 2 diseases
3	In which journal was it published that {Entity1} has an enhancing effect on {Entity2}?	APP, gamma Secretase Complex	Document and Journal
4	What is the shortest way between {Entity1} and {Entity2} and what is on that way?	Axonal transport, LRP3	Path between nodes
5	Where was it published that {Entity1} has an enhancing effect on {Entity2} and what documents cite this?	APP, gamma Secretase Complex	List of publishing and citing documents
6	What are the most important entities in context of {Entity1} disease?	Alzheimer's	Page Rank of neighboring entities
7	Which authors publish in the same journal on the topic {Entity1} and have not yet published together?	Alzheimer's disease	List of author couples
8	Find a path of biological entities that connects {Entity1} with {Entity2}	Alzheimer's disease, ACHE	Path of entities
9	Are there authors within the same affiliation who make contradictory statements regarding protein {Entity1} and protein {Entity2}?	Apoptotic process, SLC25A21	Number of statements for both variants
10	Do the data in the literature correlate with the concomitant diseases for illness {Entity1}? So are the genes mentioned in {Entity1} documents also mentioned in {Entity2} documents of the concomitant disease?	Alzheimer's, Down syndrome	Genes involved in both diseases in the literature
11	Does the function of a gene {Entity} differ in different contexts?	IL1B	List of all functions in contexts
12	How far apart are {document1} and {document2}?	PMID:16160056, PMID:16160050	Shortest path between documents
13	Does the biological process on gene {Entity1} also exist in context of {Entity2}? And what author describes it?	APOE, brain	Outcome graph in context of the brain
14	Are there BEL statements that have no source, so should be checked?	–	List of relations
15	How many sources are there for the statements of a contradictory BEL statement?	hasRelation. function = increases, decreases	Number of sources for each of the cases

Table 2 continued

#	Query	Input example	Output
16	Is there also a relation between the documents describing the entities {Entity1} and {Entity2} that matches the relation in a BEL statement with the entities {Entity1} and {Entity2}?	APP, Alzheimer	Document pairs
17	Find the oldest document describing an entity {entity}	APP	Oldest Document
18	Is a reviewer {Author1} suitable for a proposal with the author {Author} or is there a conflict of interest? Does the reviewer have relationships with the author in the form of joint work or equal affiliation?	Ulrich Rothe, A. Castillo	Potential Graph between the authors
19	On which topics does the author {Author} write most?	Ulrich Rothe	List of the most frequent annotations
20	In which other journals could the author {Author} write with his main topics? Which journal in which he has not yet published would suit him from his main topics?	Ulrich Rothe	List of journals that could fit him
21	Which Affiliation has the most publications on the topic {Entity} in the Journal {Journal}?	D008358, Biotechnology letters	Affiliation with the highest number of publications
22	From when is the document cited in documents dealing with the subject {Entity}?	D017629	Publication date of cited document
23	Which document is the most cited paper in connection with {Entity}, of papers that also annotate {Entity}? Determined by PageRank.	D017629	Most cited paper-type document
24	Which entities have many relations with {Entity}? Determined by Community Detection.	APP	Surrounding community graph
25	Which author connects the two subject areas {Entity1} and {Entity2} most strongly?	Alzheimer Disease, Parkinson	Author with highest betweenness centrality
26	Which gene {Entity} is the most important?	Entity.source = HGNC	Entity with highest degree centrality
27	Are there strongly connected components between the entities?		Assignment of the entities to cliques

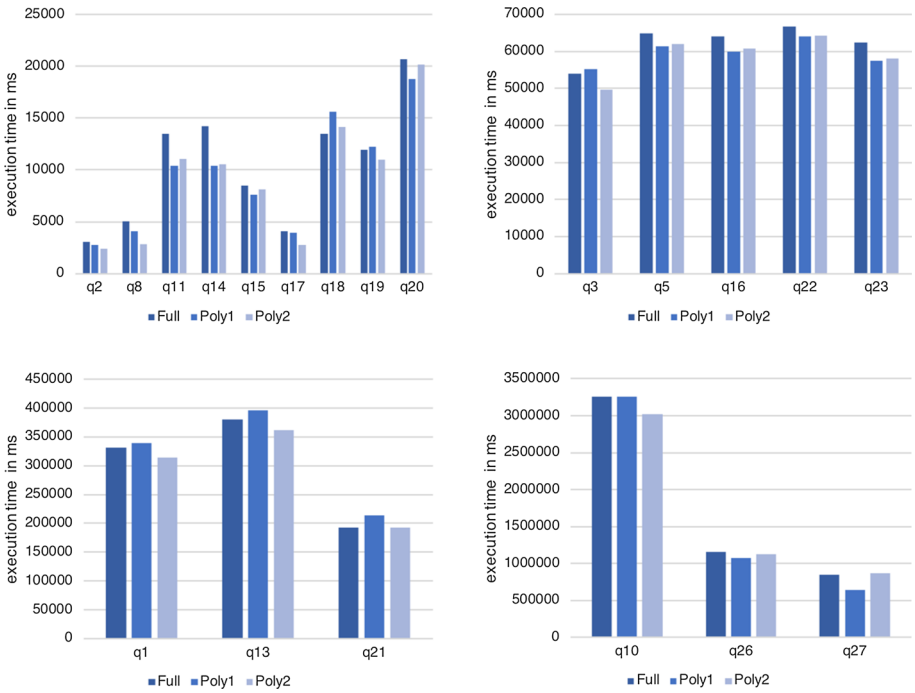

Fig. 8 Runtime results of 27 real-world queries. The queries are grouped in four diagrams with similar runtimes for a better overview. We see that the execution time of most queries is improved with *Poly1* and *Poly2*. In the best case, the improvement is 43%

The differences in the results become clearer when looking at the differences in runtimes in percent comparing them with each other. The differences in the observed running times become clearer when analyzing the percent change in the runtime when compared to *Full* as shown in Table 3. For both systems, the average percent decrease in runtimes is calculated for all queries, in order to compare both polyglot systems each other and with *Full*. It is important to notice that the speedup factor is significant especially for those queries depending on a lot of attribute data—which is the data stored in the redis database, see in particular queries 14, 11 and 2.

There is no information for queries 4, 6, 7, 9, 12, 24 and 25, for which no runtime could be determined on the systems as they did not go to completion. These queries are primarily graph algorithms categorized as *local and global structures* in the schema discussed earlier.

The results do not show a clear trend for any of the categories discussed. The *RPQ* class improves on average by 15.8% while the *ECRPQ* class by 10.5%. The classes *CRPQ*, *Page Rank*, *Degree Centrality* and *Connected Components* are in the single-digit percentage range. Since the speedup factor heavily depends on how many attributes of nodes and edges are considered, it is not easy to measure this impact. This explains why for other time-consuming queries, the improvement of efficiency is not significant. In general, the subcategories of *local structures* seem to benefit more from the polyglot persistence designs. In addition, there is a tendency for queries that only need to consider a few node and edge types (often `entity` and `hasRelation`) to experience a greater decrease in runtimes than queries with many node and edge types.

Table 3 Decrease in the runtime of t_{poly1} and t_{poly2} compared to t_{full} in %, sorted by Poly1 decreasing

Query	Poly1 (%)	Poly2 (%)	Problem
14	26.8	25.8	RPQ
27	23.8	−2.6	Connected components
11	22.5	17.7	RPQ
8	18.2	43.3	ECRPQ
2	11.5	22.9	RPQ
15	10.3	4.5	CRPQ
20	9.2	2.5	CRPQ
23	7.7	6.8	Page rank
26	6.8	2.4	Degree centrality
16	6.6	5.1	RPQ
5	5.4	4.6	CRPQ
22	3.8	3.5	ECRPQ
17	3.1	31.9	RPQ
10	−0.2	7.0	CRPQ
3	−2.3	7.9	CRPQ
19	−2.3	8.0	RPQ
1	−2.5	4.9	CRPQ
13	−4.1	4.8	CRPQ
21	−11.0	−0.3	RPQ
18	−15.7	−15.1	ECRPQ
Average	5.8	9.8	

The evaluation was done using the same queries

2.4 Graph queries

Here, we present results of some of those 27 queries introduced. Query 1 returns a subgraph: Which author was the first to state that {Entity1} has an enhancing effect on {Entity2}? We may execute this query using `match (n:Entity preferredLabel: "APP")-[r:hasRelation function: "increases"]->(m:Entity preferred Label: "gamma Secretase Complex"), (doc:Document documentID: r.context)<-[r2:isAuthor]-(author:Author) return doc, author order by doc.publicationDate limit`.

A result graph can be found in Fig. 9. On the left, the isAuthor relation with the most recent author can be found. On the left the limit parameter was changed to 10 and thus the result graph shows the most recent 10 publications and authors.

Query 2 returns a subgraph: Which genes {Entity1} play a role in two diseases {Entity2}? One example output graph can be found in Fig. 10 (left). Due to the limitation of our model to Alzheimer's disease, it is not surprising to find only one gene—APP. If we remove the limitation to two distinct diseases, the database returns a larger graph, see Fig. 10 (right). Here, we see, that we may need to utilize inherent ontology information to filter those nodes, that cover diseases. But we also see a second gene—TNF—with other diseases like Diabetes.

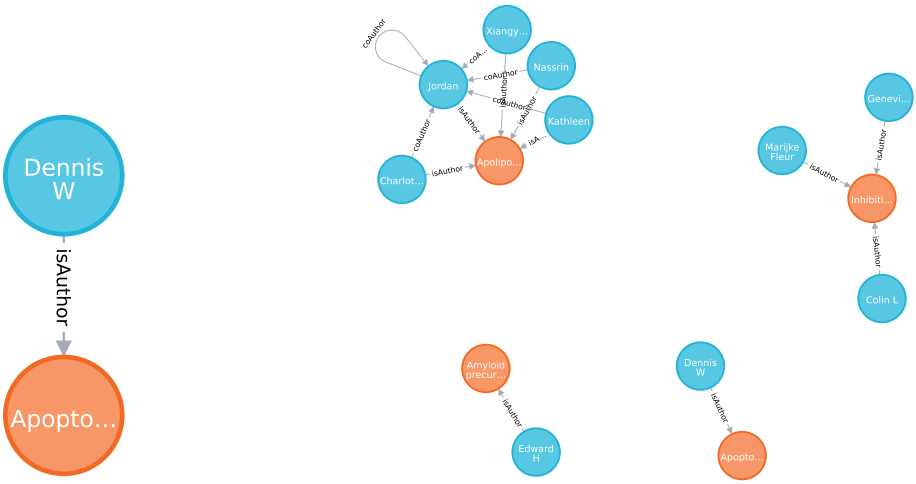

Fig. 9 Example: The resulting subgraph for query 1: Which author was the first to state that {Entity1} has an enhancing effect on {Entity2}? On the left the first author (blue node) and the publication (orange), on the left the result shows the most recent 10 authors (blue) with their publications on this topic (orange). Here, it is obvious that the result graph is often hard to visualize: As the number of nodes and edges increases it is not easy to see all details (colour figure online)

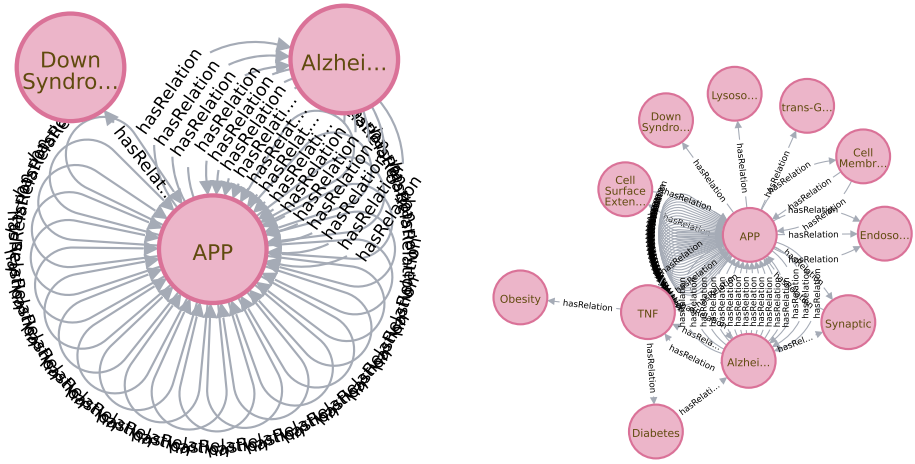

Fig. 10 (left) A result subgraph example for query 2: Which genes {Entity1} play a role in two diseases {Entity2}? Here, we see Alzheimer's disease and Down Syndrome and the gene APP. The relations (and especially the self relations APP→APP) can't be visualized in a readable way but highlight the complexity of the knowledge graph structure. (right) The resulting subgraph for query 2 without limitation to two distinct diseases: Which genes {Entity1} play a role in two diseases {Entity2}? In contrast to figure, the results are even more complex: APP plays a role in even more diseases. There are also some relations related to TNF (Obesity, Diabetes and Alzheimer's disease)

3 Discussion

Here, we introduce the graph-theoretic foundation for a general context concept within semantic networks and show a proof of concept based on biomedical literature and text mining. Our test system contains a knowledge graph derived from PubMed data which is then enriched with text mining data and domain-specific language data coming from BEL. This dense graph has more than 71M nodes and 850M relationships. We discuss the impact of this novel approach using 27 real-world use cases and graph queries.

This proof of concept of a biomedical knowledge graph combines several sources of data by relating their contextual data to one another. We processed data from PubMed and PMC which generated more than 30M document and metadata nodes. This initial knowledge graph was extended using results from text mining and NLR-tools already included in our software as well as with named entities from ontologies also stored in SCAIView. In addition, we added data generated by domain-specific languages such as BEL. Thus, we were able to assess both small data sets as well as large collections of data.

First we discuss the missing data and data integration problems, as well as the technical issues which need to be solved. Afterward, we give an outlook on NLP based on context information and the impact on answering semantic questions which is highly related to the FAIRification of research data. Finally, we discuss the integration of these methods with personalized medicine.

3.1 Missing data and quality control

There were several issues with data integration and missing data. Initially, we tried to integrate publication data from several external sources, but some publishers used OCR technologies to convert PDF documents in XML structures. These proved problematic to process as some fields were either missing or incorrectly filled out.

We have not yet solved the issue of author and affiliation disambiguation which remains a widely discussed topic, see [40]. An interesting novel approach—also based on Neo4j database technology—was introduced in [41]. Franzoni used topological and semantic structures within the graph for author disambiguation. Taking this into consideration, we plan to integrate such state-of-the-art technologies into our software in the future.

In addition, we did not consider the problem of quality control since the focus of our work was different. Our approach merged existing data sets and thus we rely on the quality control of these data sets. But merging data might lead to more quality problems as the issues with missing data have shown. Thus further research has to be carried out here. In addition, we presented some subgraphs received as output of the queries. However, we could not present and discuss a quantitative evaluation of these solutions. Since the output heavily depends on the data stored in the knowledge graph, this is another issue that needs to be considered helping to understand the quality of the results.

3.2 Performance

Furthermore, performance for some semantic queries remains a major problem due to the massive latency for request. Although the software is integrating in our microservice architecture, see [19], some queries did not run to completion. Here, we attempt to improve our initial setup by establishing a polyglot persistence architecture in the database backend [7]. The detailed analysis in Table 3 raises new questions: Is it possible to determine queries which

are optimal for one particular architecture? The results generated through this modification are very encouraging and we will discuss additional topics for further research.

3.3 Context-based NLP

This novel system was designed to extend our knowledge base by utilizing contextual data. Context serves as a very important foundation for text mining [6]. Context-based NER was discussed by [42] and there is still ongoing research such as the content-aware attributed entity embedding (CAAEE), see [43]. The key strength of our approach is that in every step of text mining and NLP, all contextual data is readily available and new data is continuously added. Therefore, this system can be used for both building and validating Machine Learning (ML) and AI approaches.

Of course, novel context data is not only suitable for NER, but also for relation extraction. Prajapati proposed a novel approach to context-based relation extraction [44]. Although our example is based on a small data set, the findings suggest that a lot of existing data can be utilized as context data such as entities annotated by NER or manually curated BEL statements.

Importantly, this research has several practical applications. First, it can be used to validate data sets for ML and AI approaches in context of text mining, however, further investigation is required as to how this data can be used systematically. And second, this approach generalizes the idea of context so that it can be used for semantic questions.

3.4 Answering semantic questions and FAIRification of data

Semantic questions can be formulated as subgraph structures of the initial knowledge graphs. For example we may ask: "Which articles have been authored by Pacheco?". This leads to a subgraph with two nodes v_1, v_2 where v_1 =Pacheco and an edge (v_1, v_2) =isAuthor, though this is a relatively simple query, much more complex examples can also be used.

In general, these semantic subgraph queries (or: graph queries) have an input $Q = (V, E) \subset G$ and output all subgraphs $H \subset G$ with $H \simeq Q$. Therefore, the problem of answering semantic questions is a generalization of the subgraph isomorphism problem. Here, we presented a more detailed classification of queries, of which many can be solved in polynomial time and as shown by their performance (Fig. 8).

We know that the most general case, subgraph isomorphism, is NP-hard, see [45]. It would be interesting to find a formulation of the generalization or restrictions that can be applied to these problems. Because Cypher already provides us with the possibility to query graph substructure, further research should be directed toward exploring the runtime, finding a better categorization of queries and discovering novel heuristics to solve this deficiency.

While this work did not consider the impact of novel ontologies and terminologies, it did substantiate the impact of them on context data. This is an interesting and important step toward the FAIRification of data. Wilkinson introduced his FAIR guiding principles in [46] referring to the findability, accessibility, interoperability and reusability of data, especially in regards to research data. A consequent application of context idea leads to metadata as context on data which can afterward be used to make metadata searchable even if the data itself is protected due to data protection rules. Thus, the inclusion of context in an information system such as SCAIView will allow the data to be both findable and accessible. Furthermore, if interoperable ontologies are available then this data will also be interoperable hence showing that our proposed system already satisfies the three out of four issues addressed by FAIR data.

However, the generalizability of these ideas is subject to certain limitations. For instance, the question of interoperable ontologies or ontologies covering the issues of interoperability of data is still not addressed and there is still no FAIR-data information system yet available.

3.5 Perspectives for personalized medicine

Hypothesis generation and knowledge discovery in biomedical data are widely sought after in medical research and digital health. Researchers often desire and utilize these tools when diagnosing patients, searching for genomic or molecular patterns, or build longitudinal models. In addition, the massive amount of data available can be harnessed to construct a multitude of predictive and personalized medicine using ML and AI approaches. One reasonable approach to tackle reproducible research in predictive medicine would be to use a standardized and FAIR context graph for biomedical research data. However, it would be necessary to annotate not only biomedical literature, but also research data such as molecular data, imaging data, genomics and electronical health records (EHR) with contextual information in order to ensure the most accurate results.

Once implemented, this type of information system can be used to retrieve information by way contextual data (cohort size, settings, demographics, ..) as well as by content (imaging data, genomic or molecular measurements, ...) and would be able to answer questions such as "Give me a clinical trial to reproduce my results or to apply my model" or "Give me literature for phenotype A, disease B age between C and D and a CT-scan with characteristic E."

Here, we presented a novel approach capable of annotating research data with contextual information. The resulting structure is a knowledge graph representation of data, the context graph, which contains computable statement representation (e.g., RDF or BEL). This graph allows one to compare research data records from different sources as well as the selection of relevant data sets using graph-theoretical algorithms.

4 Conclusion

Storing and querying a giant knowledge graph as a labeled property graph is still a technological challenge. Here, we demonstrate how our data model is able to support the understanding and interpretation of biomedical data. We present several real-world use cases that utilize our massive, generated knowledge graph derived from PubMed data and enriched with additional contextual data. Finally, we show a working example in context of biologically relevant information using SCAIView.

Acknowledgements Valuable suggestions during the development of this method were provided by Jürgen Klein and Vanessa Lage-Rupprecht. We thank Tim Steinbach for providing some illustrations to this work. In addition, we thank Alexander Esser for his input on the initial research paper. We thank Martin Hofmann-Apitius for supporting this research activity and his valuable input.

Author Contributions This new approach goes back to an initial idea of JD and was developed by JD, AS and BS. The data sets for evaluation were produced by MJ. The manuscript was written by JD, AS and BS. All authors read and approved the final manuscript.

Ethics approval and consent to participate Not applicable.

Consent for publication Not applicable.

Availability of data and materials Not applicable. The knowledge graph is available on request; the fundamental data is available using SCAIView.

Competing interests The authors declare that they have no competing interests.

Funding This study was funded by Fraunhofer Society under the MAVO Project; Human Brain Pharmacome; EU/EFPIA Innovative Medicines Initiative Joint Undertaking under AETIONOMY (115568 to D.D.F.); European Union's Seventh Framework Programme (FP7/2007-2013); and EFPIA.

References

1. Desai M, Mehta RG, Rana DP (2018) Issues and challenges in big graph modelling for smart city: an extensive survey. Int J Comput Intell IoT 1(1)
2. Dumontier M, Callahan A, Cruz-Toledo J, Ansell P, Emonet V, Belleau F, Droit A (2014) Bio2rdf release 3: a larger connected network of linked data for the life sciences. In: Proceedings of the 2014 international conference on posters and demonstrations track, vol 1272, pp 401–404
3. Callahan A, Cruz-Toledo J, Ansell P, Dumontier M (2013) Bio2rdf release 2: improved coverage, interoperability and provenance of life science linked data. In: Extended semantic web conference, pp 200–212
4. Li S, Xin L (2014) Research on integration and sharing of scientific data based on linked data-a case study of bio2rdf. Res Library Sci 21
5. Natsiavas P, Koutkias V, Maglaveras N (2015) Exploring the capacity of open, linked data sources to assess adverse drug reaction signals. In: SWAT4LS, pp 224–226
6. Aggarwal CC, Zhai C (2012) An introduction to text mining. In: Mining text data. Springer, Berlin, pp 1–10
7. Dörpinghaus J, Stefan A (2019) Knowledge extraction and applications utilizing context data in knowledge graphs. In: 2019 Federated conference on computer science and information systems (FedCSIS). IEEE, pp 265–272
8. Hanisch D, Fundel K, Mevissen H-T, Zimmer R, Fluck J (2005) ProMiner: rule-based protein and gene entity recognition. BMC Bioinform 6 Suppl 1:14
9. Fluck J, Klenner A, Madan S, Ansari S, Bobic T, Hoeng J, Hofmann-Apitius M, Peitsch M (2013) Bel networks derived from qualitative translations of bionlp shared task annotations. In: Proceedings of the 2013 workshop on biomedical natural language processing, pp 80–88
10. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT et al (2000) Gene ontology: tool for the unification of biology. Nat Genet 25(1):25
11. Wishart DS, Feunang YD, Guo AC, Lo EJ, Marcu A, Grant JR, Sajed T, Johnson D, Li C, Sayeeda Z, et al (2017) Drugbank 5.0: a major update to the drugbank database for 2018. Nucleic Acids Res 46(D1):1074–1082
12. Khan K, Benfenati E, Roy K (2019) Consensus qsar modeling of toxicity of pharmaceuticals to different aquatic organisms: ranking and prioritization of the drugbank database compounds. Ecotoxicol Environ Saf 168:287–297
13. Hey J (2004) The data, information, knowledge, wisdom chain: the metaphorical link. Intergovernmental Oceanographic Commiss 26:1–18
14. Zeleny M (1987) Management support systems: towards integrated knowledge management. Hum Syst Manag 7(1):59–70
15. Ackoff RL (1989) From data to wisdom. J Appl Syst Anal 16(1):3–9
16. Rowley J (2007) The wisdom hierarchy: representations of the DIKW hierarchy. J Inf Sci 33(2):163–180
17. Dörpinghaus J, Jacobs M (2019) Semantic knowledge graph embeddings for biomedical research: Data integration using linked open data. In: Posters and demo track of the 15th international conference on semantic systems. (Poster and Demo Track at SEMANTiCS 2019) (2451), 46–50

18. Dörpinghaus J, Darms J, Jacobs M (2018) What was the question? A systematization of information retrieval and nlp problems. In: 2018 Federated conference on computer science and information systems (FedCSIS). IEEE
19. Dörpinghaus J, Klein J, Darms J, Madan S, Jacobs M (2018) Scaiview: a semantic search engine for biomedical research utilizing a microservice architecture. In: Proceedings of the posters and demos track of the 14th international conference on semantic systems - SEMANTiCS2018
20. Webber J, Eifrem E (2015) Graph databases
21. Rogers FB (1963) Medical subject headings. Bull Med Libr Assoc 51:114–116
22. Yang H, Lee H (2018) Research trend visualization by mesh terms from pubmed. Int J Environ Res Public Health 15(6):1113
23. Cyganiak R, Wood D, Lanthaler M (2014) RDF 1.1 concepts and abstract syntax. W3C recommendation, W3C (February 2014). http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/
24. Patel-Schneider P, Rudolph S, Krötzsch M, Hitzler P, Parsia B (2012) OWL 2 web ontology language primer (second edition). Technical report, W3C (December 2012). http://www.w3.org/TR/2012/REC-owl2-primer-20121211/
25. Summers E, Isaac A (2009) SKOS simple knowledge organization system primer. W3C note, W3C (August 2009). http://www.w3.org/TR/2009/NOTE-skos-primer-20090818/
26. Zeng M, Hlava M, Qin J, Hodge G, Bedford D (2007) Knowledge organization systems (kos) standards. Proc Assoc Inf Sci Technol 44(1):1–3
27. Guidelines for the construction (2005) format, and management of monolingual controlled vocabularies. Standard, National Information Standards Organization, Baltimore, Maryland, USA
28. Zeng M (2008) Knowledge organization systems (kos) 35:160–182
29. Malhotra A, Younesi E, Gündel M, Müller B, Heneka MT, Hofmann-Apitius M (2014) Ado: a disease ontology representing the domain knowledge specific to Alzheimer's disease. Alzheimer's Dementia 10(2):238–246
30. Iyappan A, Younesi E, Redolfi A, Vrooman H, Khanna S, Frisoni GB, Hofmann-Apitius M (2017) Neuroimaging feature terminology: a controlled terminology for the annotation of brain imaging features. J. Alzheimers Dis. 59(4):1153–1169
31. Madan S, Fiosins M, Bonn S, Fluck J (2018). A semantic data integration methodology for translational neurodegenerative disease research. https://doi.org/10.6084/m9.figshare.7339244.v1
32. Voß J (2016) Classification of knowledge organization systems with wikidata. In: NKOS@ TPDL, pp 15–22
33. Vrandečić D (2018) Toward an abstract Wikipedia. In: Ortiz M, Schneider T (eds) 31st International workshop on description logics (DL). CEUR workshop proceedings, Aachen
34. Oßwald A, Schöpfel J, Jacquemin B (2015) Continuing professional education in open access. a French-German survey. LIBER Quarterly. J Assoc Eur Res Libraries 26(2):43–66
35. Volanakis A, Krawczyk K (2018) Sciride finder: a citation-based paradigm in biomedical literature search. Sci Rep 8(1):6193
36. Madan S, Hodapp S, Senger P, Ansari S, Szostak J, Hoeng J, Peitsch M, Fluck J (2016) The BEL information extraction workflow (BELIEF): evaluation in the BioCreative V BEL and IAT track. Database 2016
37. Madan S, Szostak J, Dörpinghaus J, Hoeng J, Fluck J (2017) Overview of BEL track: extraction of complex relationships and their conversion to BEL. In: Proceedings of the BioCreative VI workshop (2017)
38. Wood PT (2012) Query languages for graph databases. SIGMOD Rec 41(1):50–60. https://doi.org/10.1145/2206869.2206879
39. Angles R, Arenas M, Barceló P, Hogan A, Reutter J, Vrgoč D (2017) Foundations of modern query languages for graph databases. ACM Comput Surv 50(5):68–16840. https://doi.org/10.1145/3104031
40. Kim J (2019) Correction to: Evaluating author name disambiguation for digital libraries: a case of dblp. Scientometrics 118(1):383–383
41. Franzoni V, Lepri M, Milani A (2019) Topological and semantic graph-based author disambiguation on dblp data in neo4j. arXiv preprint arXiv:1901.08977
42. Nadeau D, Sekine S (2007) A survey of named entity recognition and classification. Lingvisticae Investigationes 30(1):3–26
43. Cai D, Wu G (2019) Content-aware attributed entity embedding for synonymous named entity discovery. Neurocomputing 329:237–247
44. Prajapati P, Sivakumar P (2019) Context dependency relation extraction using modified evolutionary algorithm based on web mining. In: Emerging technologies in data mining and information security. Springer, Göttingen, pp 259–267

45. Cook SA (1971) The complexity of theorem-proving procedures. In: Proceedings of the third annual ACM symposium on theory of computing, pp 151–158 (1971). ACM
46. Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, Blomberg N, Boiten J-W, da Silva Santos LB, Bourne PE, et al (2016) The fair guiding principles for scientific data management and stewardship. Sci Data 3

Dr. Jens Dörpinghaus is senior researcher at the Federal Institute for Vocational Education and Training (BIBB) and the German Center for Neurodegenerative Diseases (DZNE) and lecturer at the University Koblenz-Landau. His main research interests are in the field of data science, knowledge graphs, graph theory and algorithm design.

Andreas Stefan, M.Sc., studied at the Department of Computer Sciences at the Bonn-Rhein-Sieg University of Applied Sciences and was working with Fraunhofer.

Dr. Marc Jacobs is Group leader Software and Scientific Computing, Deputy Head of Business Area Bioinformatics at the Fraunhofer institute SCAI. He is working for Fraunhofer for more than 15 years as a Computer Scientist in the field of algorithm design and professional software development. His main research interests are information extraction and retrieval in the field of chemistry and pharmacology.

Bruce Schultz is a research scientist at Fraunhofer SCAI, focusing on the development and enrichment of knowledge graphs. His career began in immunology and vaccine research but has since shifted his focus towards the creation of software packages for biologists. Bruce is also a lecturer at the University of Bonn, where he teaches scientific programming to students in the Life Science Informatics program.